

Projektna dokumentacija web aplikacije fakulteta

Informacioni sistem za upravljanje predmetima, ispitima i studentima

Programski alati za razvoj softvera

Jovan Jovanovic ITS 20/24

SADRŽAJ

1. Uvod i cilj dokumenta.....	4
2. Motivacija za razvoj sajta za visokoškolsku ustanovu.....	6
3. Kratak istorijat i pregled poznatih sajtova fakulteta i viših škola.....	7
4. Opis osnovne strukture sajta i mapa sajta.....	8
4.1. Ilustrovana mapa sajta.....	9
5. Dijagrami sa slikama i opisima.....	12
5.1. ER dijagram baze podataka.....	12
5.1.1. Opis entiteta.....	12
5.2. Use Case dijagram.....	20
5.3. Sekvencijalni dijagrami.....	22
5.4. Process Flow (tok procesa) dijagram.....	27
6. Test plan i test case-ovi.....	33
6.1. Plan testiranja.....	33
6.2. Struktura test case-a.....	34
6.3. Test case-ovi - uloga: Student.....	35
6.4. Test case-ovi - uloga: Profesor.....	39
6.5. Test case-ovi - uloga: Administrator.....	43
6.6. Zaključak sekcije o testiranju.....	46
7. Tehnologije i resursi.....	48
7.1. Backend tehnologije.....	48
7.2. Frontend tehnologije.....	49
7.3. Baza podataka.....	49
7.4. Alati za izradu dijagrama.....	50
7.5. Razvojno okruženje i deploy.....	50
7.6. Tutorijali, dokumentacija i kursevi.....	51
7.7. Sajtovi za ikonice, slike, fontove i boje.....	52
7.8. Korišćeni AI alati.....	53
8. User Manual (uputstvo za korišćenje).....	54
8.1. Pristup aplikaciji.....	54
8.2. Logovanje i odjava korisnika.....	54
8.2.1. Logovanje.....	54
8.2.2. Odjava (Logout).....	55
8.3. Navigacija i zajednički elementi interfejsa.....	56
8.4. Korišćenje aplikacije - uloga: Student.....	57
8.4.1. Pregled predmeta ("Moji predmeti").....	57
8.4.2. Prijava ispita.....	57
8.4.3. Pregled prijava i ocena.....	58
8.4.4. Preuzimanje nastavnih materijala.....	59
8.4.5. Pregled obaveštenja.....	59
8.5. Korišćenje aplikacije - uloga: Profesor.....	59
8.5.1. Pregled i izbor predmeta.....	59
8.5.2. Upravljanje nastavnim materijalima.....	60
8.5.3. Unos i izmena ocena.....	60
8.5.4. Slanje obaveštenja studentima.....	61

8.6. Korišćenje aplikacije - uloga: Administrator.....	61
8.6.1. Upravljanje korisnicima.....	62
8.6.2. Upravljanje predmetima.....	62
8.6.3. Upravljanje ispitnim rokovima.....	63
8.6.4. Objavljivanje opštih obaveštenja.....	63
8.6.5. Pregled statistike.....	63
8.7. Tipične poruke o greškama i preporuke za korisnike.....	64
9. Design Patterns i organizacija koda.....	66
9.1. MVC (Model-View-Controller) arhitektura.....	66
9.2. Singleton pattern - konekcija ka bazi podataka.....	67
9.3. Šabloni (template) i ponovna upotreba view-ova.....	69
9.4. Factory pattern (mogućnost primene).....	70
9.5. Logička struktura projekta i organizacija direktorijuma.....	71
10. Bezbednost web aplikacije.....	72
10.1. Autentifikacija i zaštita lozinki.....	72
10.2. Kontrola pristupa i uloge (Role-Based Access Control).....	72
10.3. Validacija i sanitizacija ulaznih podataka.....	73
10.4. Zaštita od SQL Injection napada.....	74
10.5. Zaštita od XSS (Cross-Site Scripting) napada.....	74
10.6. Sesije, kolačići i timeout.....	75
10.7. Ostali bezbednosni aspekti i preporuke.....	75
11. Budući razvoj i unapređenja.....	77
11.1. Integracija sa mobilnom aplikacijom i REST API.....	77
11.2. Sistemi obaveštavanja (email / SMS / push notifikacije).....	78
11.3. Napredna analitika i izveštavanje.....	78
11.4. AI chatbot i pametni asistenti.....	79
11.5. Povezivanje sa LMS sistemima i dodatne funkcionalnosti.....	79
12. Zaključak.....	81
13. Literatura.....	83
13.1. Alati i resursi za dizajn.....	83
13.2. AI alati.....	83

1. Uvod i cilj dokumenta

U okviru predmeta *Serverske tehnologije* razvijena je web aplikacija fakulteta čiji je cilj da objedini osnovne procese rada visokoškolske ustanove - upravljanje korisnicima, predmetima, ispitnim rokovima, prijavama ispita, ocenama i nastavnim materijalima.

Ovaj projekat iz predmeta *Programski alati za razvoj softvera* predstavlja nastavak tog rada i fokusira se na **izradu kompletne dokumentacije** za postojeću web aplikaciju, u skladu sa standardima inženjerske prakse. Umesto da se razvija novi softver od nule, akcenat je na planiranju, analizi, modelovanju, testiranju i opisivanju sistema koji je već implementiran u okviru ranijeg projekta.

Cilj dokumenta je da:

- pruži jasan opis svrhe i funkcionalnosti aplikacije,
- prikaže strukturu baze podataka kroz ER dijagram i objašnjenje veza između entiteta,
- definiše korisničke uloge i njihove funkcionalnosti kroz Use Case dijagram,
- prikaže tokove komunikacije između korisnika, web interfejsa, servera i baze podataka kroz sekvencijalne i process flow dijagrame,
- dokumentuje test plan i konkretne test case-ove,
- sistematizuje korišćene tehnologije i alate,
- objasni način korišćenja aplikacije iz perspektive krajnjeg korisnika (user manual),
- predstavi primenjene dizajn paterne i organizaciju koda,
- opiše postojeće bezbednosne mehanizme i predloži unapređenja,

- predloži ideje za budući razvoj sistema (API, chatbot, AI analitika i sl.).

Na ovaj način, dokument služi kao centralno mesto na kome se nalaze sve ključne informacije o softverskom sistemu - od logičkog modela podataka i funkcionalnih zahteva, preko testiranja, pa do bezbednosti i planiranog razvoja. Takva dokumentacija je neophodna u realnim projektnim timovima kako bi se olakšalo održavanje sistema, uvođenje novih članova tima i dalja nadogradnja aplikacije.

2. Motivacija za razvoj sajta za visokoškolsku ustanovu

Savremene visokoškolske ustanove se sve više oslanjaju na web sisteme kako bi informatizovale ključne procese: upis studenata, prijavu ispita, evidenciju ocena, objavu obaveštenja i deljenje nastavnih materijala. Tradicionalan pristup, koji se oslanja na papirnu dokumentaciju ili nepovezane aplikacije, dovodi do čestih grešaka, dupliranja posla i gubitka vremena kako za studente, tako i za administraciju i nastavnike.

Motivacija za razvoj ovakvog sajta je da se **centralizuju podaci i procesi** vezani za studente i nastavu. Jedinstvena web aplikacija omogućava:

- studentima da na jednom mestu vide predmete na koje su upisani, prijave ispite i prate svoje ocene,
- profesorima da jednostavno unose ocene, postavljaju materijale i obaveštenja,
- administraciji da upravlja korisnicima, ispitnim rokovima, predmetima i statistikom.

Pored funkcionalne koristi, ovakav sistem doprinosi i **transparentnosti** rada institucije: studenti imaju jasan uvid u svoje obaveze i rezultate, a profesori i služba imaju tačne i ažurne podatke koji su odmah dostupni.

Dodatna motivacija je edukativna - projekat omogućava studentima informatike da kroz realan primer fakultetskog informacionog sistema primene znanja iz web programiranja, serverskih tehnologija, baza podataka, dizajn paterna, testiranja i bezbednosti, kao i da se upoznaju sa načinom na koji se dokumentuje profesionalni softver.

3. Kratak istorijat i pregled poznatih sajtova fakulteta i viših škola

Prve verzije sajtova fakulteta uglavnom su bile statične stranice sa osnovnim informacijama o ustanovi, studijskim programima i kontakt podacima. Vremenom, kako je rasla potreba za digitalizacijom procesa, sajtovi su počeli da uključuju rasporede nastave, obaveštenja, preuzimanje dokumenata i formulara. Ipak, u mnogim slučajevima ti sistemi su bili odvojeni od internih studentskih servisa i evidencija.

Danas većina fakulteta poseduje naprednije web portale i informacioni sistem za studente ("e-student", "student service", elektronski indeks i sl.). Kroz takve portale studenti mogu da:

- prijave ispite online,
- proveriti rezultate i ocene,
- proveriti uplatu školarine,
- preuzmu nastavne materijale i obaveštenja.

S druge strane, nastavnici putem istih sistema imaju mogućnost da unose ocene, formiraju spiskove studenata po predmetima, objavljuju materijale i rezultate, dok studentska služba vodi evidenciju o upisu, promenama godina studija i drugim administrativnim procesima.

Iako su funkcionalnosti ovakvih sistema u mnogim ustanovama slične, način na koji su implementirani i nivo integracije sa ostalim servisima (finansije, LMS sistemi, elektronsko učenje) značajno se razlikuju. U ovom projektu fokus je na **osnovnom informacionom sistemu fakulteta**, koji objedinjeno rešava upis na predmete, prijavu ispita, vođenje ocena, rad sa materijalima i obaveštenjima, uz jasnu podelu uloga na studente, profesore i administratore.

4. Opis osnovne strukture sajta i mapa sajta

Web aplikacija fakulteta organizovana je tako da jasno razdvoji javni deo sajta (dostupan svim posetiocima) od zatvorenog dela kome pristupaju samo registrovani korisnici (studenti, profesori i administratori).

Na najvišem nivou, struktura sajta obuhvata sledeće celine:

- **Početna stranica (Home)** - osnovne informacije o fakultetu, kratki opis sistema i linkovi ka login stranici.
- **Login / autentifikacija** - formular za unos email adrese i lozinke. Nakon uspešne prijave, korisnik se preusmerava na odgovarajući dashboard u zavisnosti od svoje uloge.
- **Admin panel** - deo sajta namenjen administratorskoj ulozi.
- **Profesorski panel** - deo sajta namenjen nastavnicima.
- **Studentski panel** - deo sajta namenjen studentima.

U okviru **admin panela** nalaze se sekcije:

- *Korisnici* - pregled, dodavanje, izmena i brisanje korisnika sistema (studenti, profesori, admini).
- *Predmeti* - upravljanje predmetima i dodeljivanje profesora predmetima.
- *Ispitni rokovi* - kreiranje i izmena ispitnih rokova (jan/feb/jun...).
- *Obaveštenja* - objava važnih informacija koje vide svi ili određena grupa korisnika.
- *Statistika* - osnovne statistike o broju studenata, prijavama ispita, položenim ispitima i sl.

U **profesorskom panelu** izdvajaju se:

- *Moji predmeti* - lista predmeta za koje je profesor zadužen.
- *Materijali* - upload, izmena i brisanje nastavnih materijala (PDF, prezentacije, zadaci).
- *Ocene* - unos i izmena ocena za prijavljene studente u aktivnim ispitnim rokovima.
- *Obaveštenja* - slanje obaveštenja studentima koji pohađaju određeni predmet.

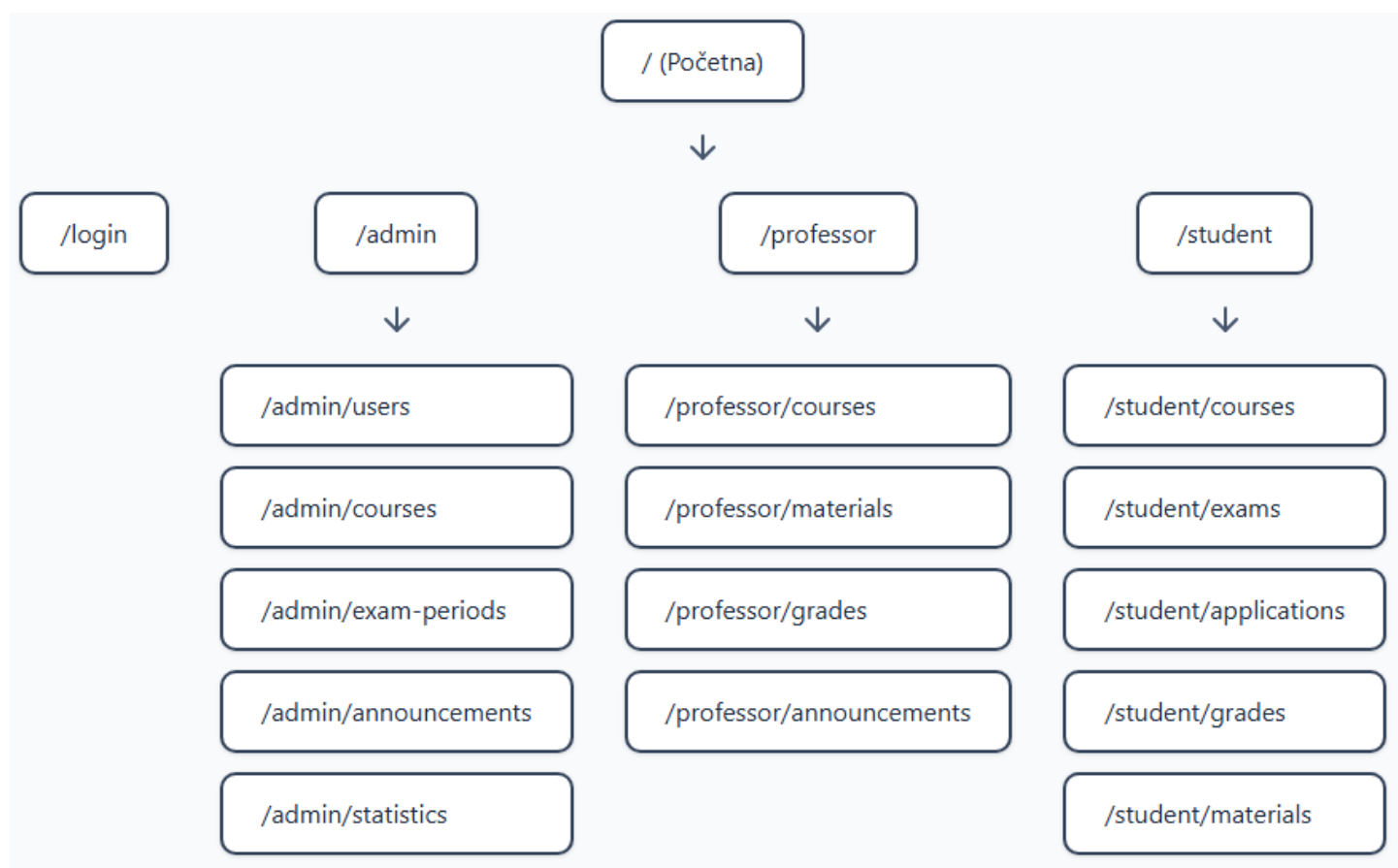
U **studentskom panelu** nalaze se:

- *Moji predmeti* - pregled predmeta na koje je student upisan.
- *Prijava ispita* - lista aktivnih ispitnih rokova i mogućnost prijave ispita za predmete koje student sluša.
- *Moje prijave* - pregled prijavljenih ispita i statusa (prijavljen, položen, nije položen).
- *Ocene* - uvid u sve položene ispite i prosečnu ocenu.
- *Materijali* - preuzimanje nastavnih materijala po predmetima.

Navigacija je realizovana kroz glavni meni koji se razlikuje u zavisnosti od uloge korisnika. Na vrhu svake stranice nalazi se traka sa linkovima ka najvažnijim sekcijama i dugme *Logout* za odjavu korisnika.

4.1. Ilustrovana mapa sajta

Mapa sajta (site map) može se prikazati kao hijerarhijski dijagram stranica, gde strelice pokazuju navigaciju između glavnih celina. Dijagram je moguće izraditi u alatima poput Draw.io ili Figme, a zatim ga eksportovati kao sliku i ubaciti u dokument.



Slika 1 - Mapa sajta web aplikacije fakulteta

Primer strukture mape sajta:

- / **(Početna)**
 - /login
 - /register (opciono, ako postoji)
- /**admin**
 - /admin/users
 - /admin/courses
 - /admin/exam-periods
 - /admin/announcements

- /admin/statistics
- **/professor**
 - /professor/courses
 - /professor/materials
 - /professor/grades
 - /professor/announcements
- **/student**
 - /student/courses
 - /student/exams
 - /student/applications
 - /student/grades
 - /student/materials

5. Dijagrami sa slikama i opisima

U ovom poglavlju prikazani su osnovni dijagrami koji opisuju strukturu i ponašanje informacionog sistema web aplikacije fakulteta. Dijagrami su izrađeni u alatu po izboru (npr. Draw.io, Figma ili Lucidchart), eksportovani kao slike i ubačeni u dokument, uz prateće tekstualno objašnjenje.

5.1. ER dijagram baze podataka

ER (Entity-Relationship) dijagram prikazuje strukturu baze podataka koja stoji iza web aplikacije fakulteta. Dijagram sadrži entitete *users*, *courses*, *enrollments*, *exam_periods*, *exam_applications*, *grades*, *materials*, *announcements* i veze između njih.

Za svaki entitet su jasno naznačeni:

- naziv tabele,
- primarni ključ (PK),
- strani ključevi (FK),
- kardinalnosti veza (1-n, n-n),
- specifični atributi koji se popunjavaju samo za studente (indeks, smer, godina studija).

ER dijagram je u potpunosti usklađen sa strukturom baze implementiranom u projektu.

5.1.1. Opis entiteta

1) USERS

Tabela `users` predstavlja sve korisnike sistema: studente, profesore i administratore.

- **users**
 - *user_id* (PK, INT, AUTO_INCREMENT) - jedinstveni identifikator korisnika
 - *name* (VARCHAR) - puno ime i prezime korisnika
 - *email* (VARCHAR, UNIQUE) - korisnička email adresa za logovanje
 - *password_hash* (VARCHAR) - heširana lozinka
 - *role* (ENUM: 'student', 'professor', 'admin') - uloga u sistemu
 - *index_number* (VARCHAR, NULLABLE) - broj indeksa (popunjen samo za studente)
 - *study_program* (VARCHAR, NULLABLE) - smer / studijski program (samo studenti)
 - *year_of_study* (INT, NULLABLE) - godina studija (samo studenti)
 - *created_at* (DATETIME) - datum kreiranja naloga

2) COURSES

Tabela `courses` opisuje predmete na fakultetu.

- **courses**
 - *course_id* (PK) - jedinstveni identifikator predmeta
 - *name* (VARCHAR) - naziv predmeta
 - *code* (VARCHAR) - šifra predmeta
 - *ects* (INT) - broj ESPB bodova
 - *year* (INT) - godina studija na kojoj se predmet sluša
 - *semester* (INT) - semestar (1-8)

- *professor_id* (FK - users.user_id) - nastavnik nosilac predmeta

Veza:

- Jedan **profesor** (*users* sa ulogom 'professor') može imati **više predmeta** - *users (1) - (n) courses*.

3) ENROLLMENTS

Tabela *enrollments* povezuje studente sa predmetima - predstavlja upis na predmet u određenoj školskoj godini.

- **enrollments**
 - *enrollment_id* (PK)
 - *student_id* (FK - users.user_id) - student koji je upisan
 - *course_id* (FK - courses.course_id) - predmet na koji je student upisan
 - *academic_year* (VARCHAR) - npr. "2024/2025"

Veze:

- Jedan **student** može biti upisan na više predmeta - *users (1) - (n) enrollments*
- Jedan **predmet** može pohađati više studenata - *courses (1) - (n) enrollments*

Ovo u praksi predstavlja n-n relaciju između studenata i predmeta, razloženu kroz tabelu *enrollments*.

4) EXAM_PERIODS

Tabela *exam_periods* čuva informacije o ispitnim rokovima (januarski, februarski, jun, septembar, itd.).

- **exam_periods**
 - *exam_period_id* (PK)
 - *name* (VARCHAR) - naziv roka (npr. "Januarski 2025")
 - *start_date* (DATE) - početak roka
 - *end_date* (DATE) - kraj roka

Jedan ispitni rok može obuhvatiti više prijava ispita za različite predmete i studente.

5) EXAM_APPLICATIONS

Tabela `exam_applications` predstavlja prijavu ispita od strane studenta u određenom roku.

- **exam_applications**
 - *exam_application_id* (PK)
 - *student_id* (FK - `users.user_id`) - student koji prijavljuje ispit
 - *course_id* (FK - `courses.course_id`) - predmet koji se polaže
 - *exam_period_id* (FK - `exam_periods.exam_period_id`) - rok u kome je ispit prijavljen
 - *applied_at* (DATETIME) - datum i vreme prijave
 - *status* (ENUM: 'prijavljen', 'položen', 'nije položen', 'otkazan')

Veze:

- Jedan **student** može imati više prijava - *users (1) - (n) exam_applications*
- Jedan **predmet** može imati više prijava u okviru više rokova - *courses (1) - (n) exam_applications*
- Jedan **ispitni rok** je povezan sa više prijava - *exam_periods (1) - (n) exam_applications*

6) GRADES

Tabela `grades` beleži konačnu ocenu koju je student dobio na konkretnoj prijavi ispita.

- **grades**
 - *grade_id* (PK)
 - *exam_application_id* (FK - `exam_applications.exam_application_id`) - na koju prijavu se odnosi ocena
 - *professor_id* (FK - `users.user_id`) - nastavnik koji je uneo ocenu

- *grade_value* (INT) - ocena (5-10)
- *grade_date* (DATE) - datum unosa ocene

Veza:

- Jedna **prijava ispita** može imati najviše jednu ocenu - *exam_applications* (1) - (0..1) *grades*
- Jedan **profesor** može uneti više ocena - *users* (1) - (n) *grades*

7) MATERIALS

Tabela `materials` čuva nastavne materijale po predmetima.

- **materials**
 - *material_id* (PK)
 - *course_id* (FK - `courses.course_id`) - predmet kome materijal pripada
 - *uploaded_by* (FK - `users.user_id`) - profesor koji je postavio materijal
 - *title* (VARCHAR) - naziv materijala
 - *file_path* (VARCHAR) - putanja do fajla (lokalna ili URL)
 - *uploaded_at* (DATETIME) - datum i vreme upload-a

Veze:

- Jedan **predmet** ima više materijala - *courses* (1) - (n) *materials*
- Jedan **profesor** može postaviti više materijala - *users* (1) - (n) *materials*

8) ANNOUNCEMENTS

Tabela `announcements` sadrži obaveštenja za studente, profesore ili sve korisnike.

- **announcements**
 - *announcement_id* (PK)
 - *title* (VARCHAR) - naslov obaveštenja
 - *content* (TEXT) - sadržaj obaveštenja

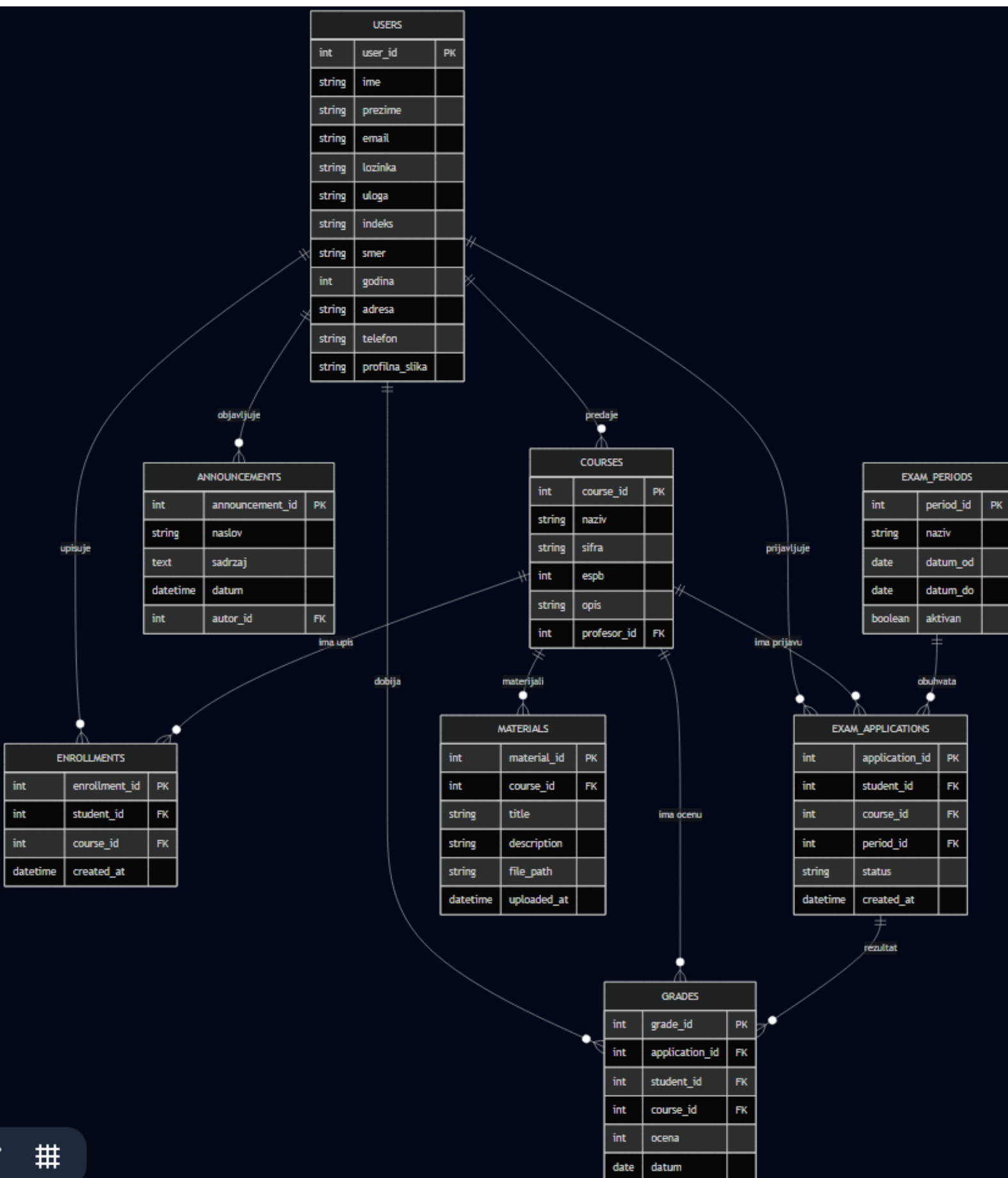
- *created_by* (FK - users.user_id) - korisnik koji je kreirao obavještenje
- *course_id* (FK - courses.course_id, NULLABLE) - ako je obavještenje vezano za konkretan predmet
- *target_role* (ENUM: 'student', 'professor', 'all') - kome je poruka namenjena
- *created_at* (DATETIME) - datum i vreme objave

Veze:

- Jedan **korisnik** može objaviti više obavještenja - *users (1) - (n) announcements*
- Jedan **predmet** može imati više obavještenja vezanih za sebe - *courses (1) - (n) announcements* (opciono)

5.1.2. Prikaz ER dijagrama

ER dijagram je izrađen u alatu *Draw.io* (ili drugom odobrenom alatu), gde su svi entiteti prikazani kao pravougaonici sa listom atributa, primarni ključevi su podvučeni, a strani ključevi posebno označeni. Kardinalnosti (1-n, 0..1) su naznačene na vezama između entiteta. U dijagramu su jasno uočljive veze između korisnika, predmeta, prijava ispita, ocena i materijala, što omogućava laku nadogradnju baze i održavanje aplikacije.



Slika 2 - ER dijagram baze podataka web aplikacije fakulteta

5.2. Use Case dijagram

Use Case dijagram prikazuje funkcionalnosti sistema iz perspektive korisnika. U ovom projektu posmatraju se tri osnovne uloge: **Admin**, **Profesor** i **Student**.

- **Admin:**
 - upravljanje korisnicima (dodavanje, izmena, brisanje),
 - upravljanje predmetima,
 - upravljanje ispitnim rokovima,
 - upravljanje obaveštenjima,
 - pregled osnovne statistike sistema.
- **Profesor:**
 - pregled svojih predmeta,
 - upload, izmena i brisanje nastavnih materijala,
 - unos i izmena ocena,
 - pregled studenata po predmetu,
 - objava obaveštenja studentima.
- **Student:**
 - upis na predmete (ukoliko je implementirano),
 - prijava ispita u aktivnom ispitnom roku,
 - pregled prijava i ocena,
 - preuzimanje nastavnih materijala.

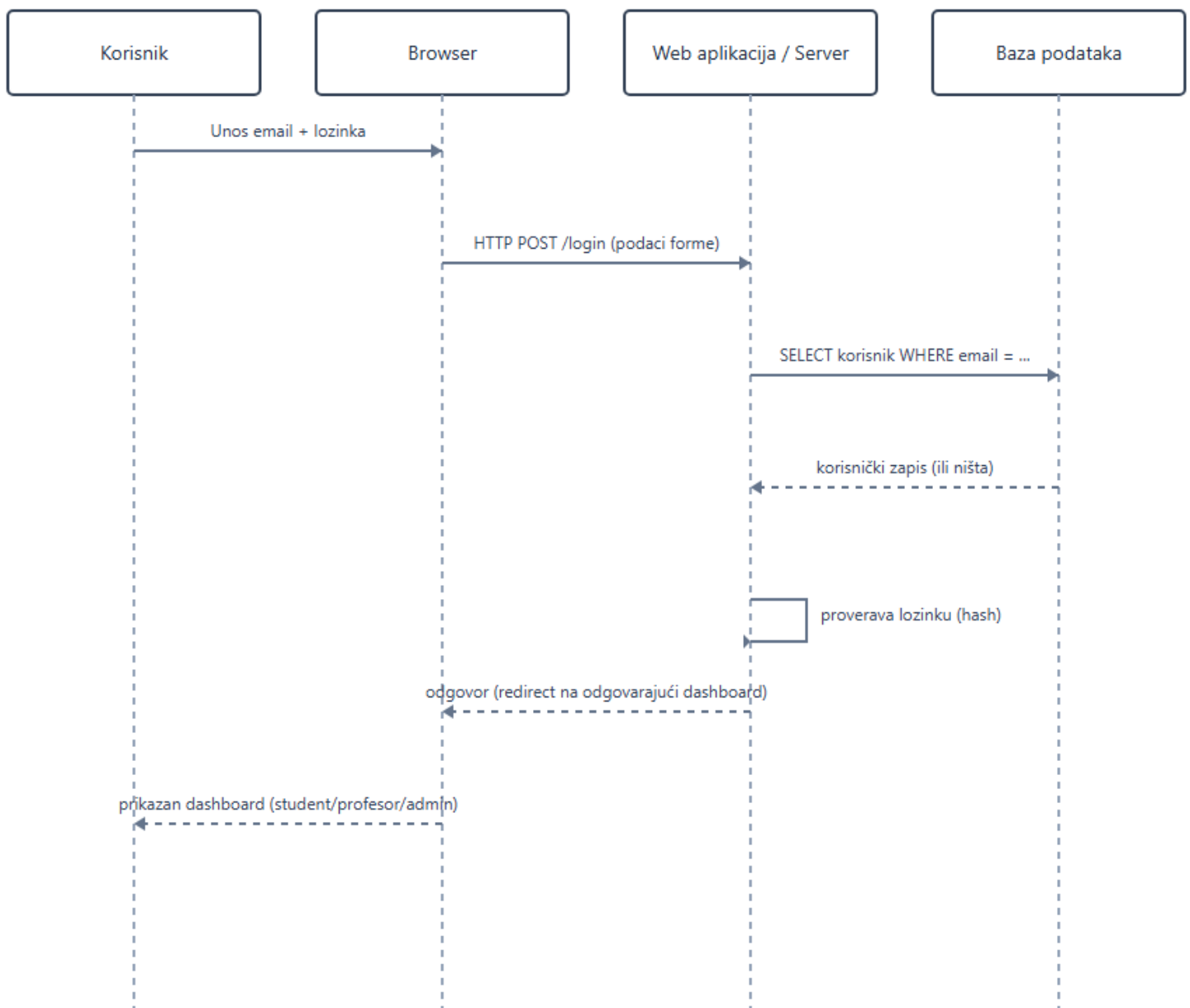


Slika 3 - Use Case dijagram web aplikacije fakulteta

5.3. Sekvencijalni dijagrami

Sekvencijalni dijagrami prikazuju protok komunikacije između korisnika, web interfejsa (browser), servera i baze podataka. Za ovaj projekat izrađena su tri sekvencijalna dijagrama za procese koji su ključni u radu sistema: **logovanje korisnika, prijava ispita i unos ocene**.

5.3.1. Sekvencijalni dijagram - logovanje korisnika

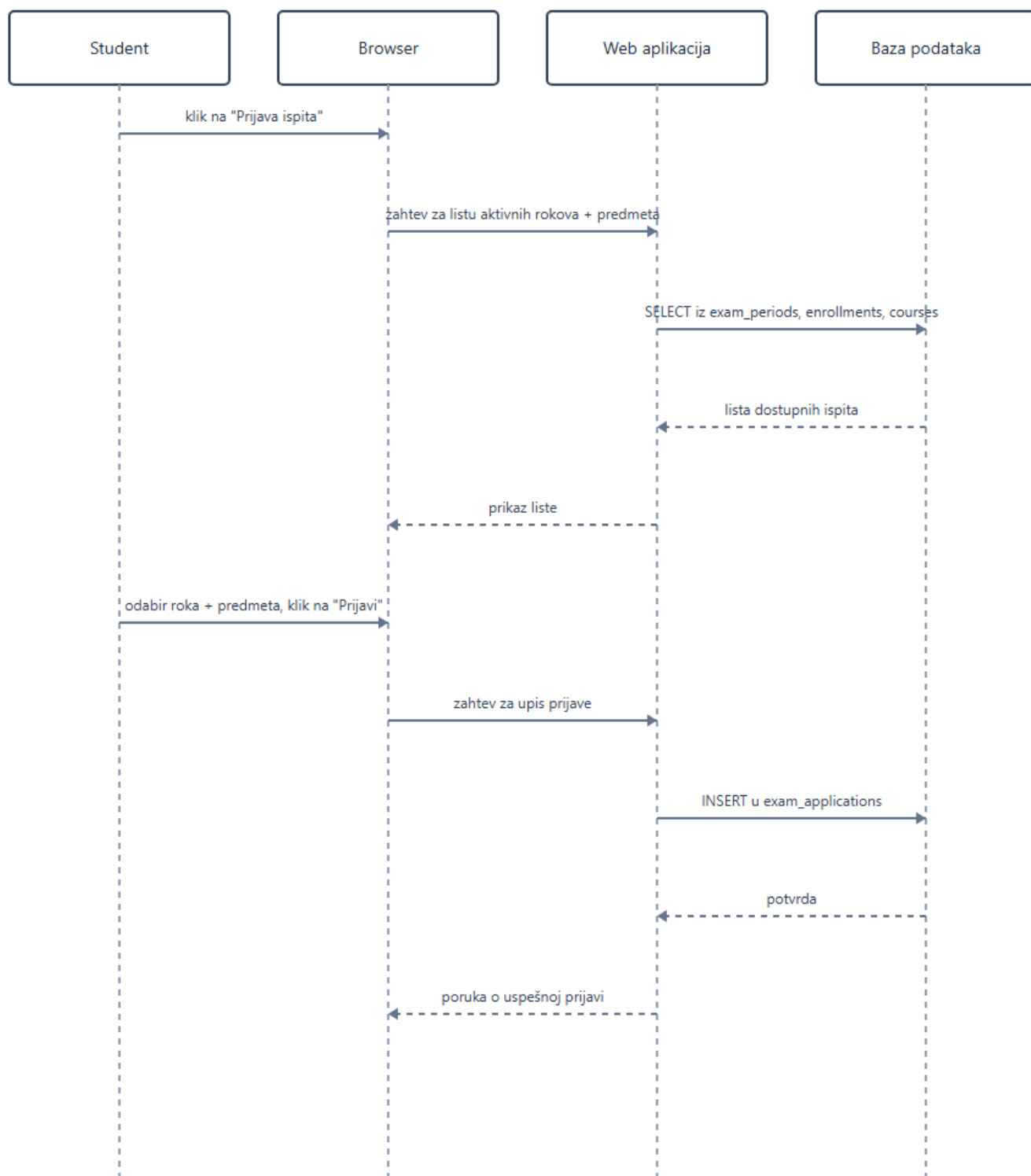


Slika 4 - Sekvencijalni dijagram procesa logovanja korisnika

Dijagram "Logovanje korisnika" prikazuje sledeće korake:

1. Korisnik u browseru otvara login stranicu i unosi email i lozinku.
2. Browser šalje HTTP zahtev ka serveru sa prosleđenim kredencijalima.
3. Server poziva sloj za pristup bazi i proverava da li postoji korisnik sa zadatim email-om.
4. Baza vraća korisnički zapis (ukoliko postoji).
5. Server upoređuje prosleđenu lozinku sa heširanom lozinkom iz baze.
6. Ako je verifikacija uspešna, kreira se sesija i određuje uloga korisnika (student/profesor/admin).
7. Server vraća odgovor browseru i redirektuje korisnika na odgovarajući dashboard.

5.3.2. Sekvencijalni dijagram - prijava ispita

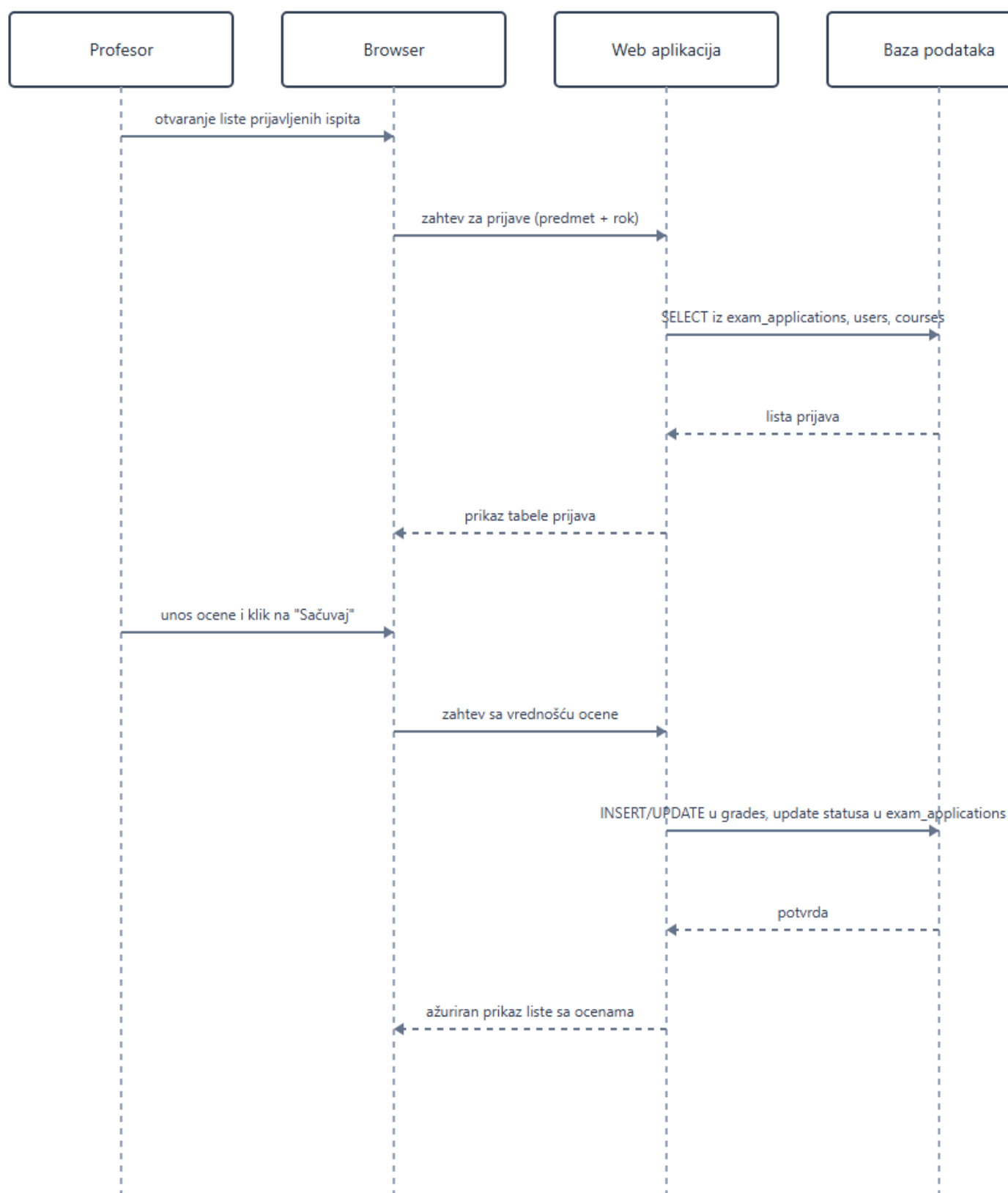


Slika 5 - Sekvencijalni dijagram procesa prijave ispita

Dijagram "Prijava ispita" obuhvata:

1. Student na svom panelu bira opciju "Prijava ispita".
2. Browser šalje zahtev serveru za listu aktivnih ispitnih rokova i predmeta na koje je student upisan.
3. Server čita podatke iz tabela *exam_periods*, *enrollments* i *courses*.
4. Baza vraća listu dostupnih ispita za prijavu.
5. Student bira predmet i rok i potvrđuje prijavu.
6. Browser šalje novi HTTP zahtev serveru sa podacima o izabranom predmetu i roku.
7. Server upisuje novu prijavu u tabelu *exam_applications*.
8. Baza potvrđuje uspešan upis.
9. Server vraća poruku o uspešnoj prijavi i osvežava listu prijavljenih ispita.

5.3.3. Sekvencijalni dijagram - unos ocene



Slika 6 - Sekvencijalni dijagram procesa unosa ocene

Dijagram "Unos ocene" opisuje sledeći tok:

1. Profesor u svom panelu otvara listu prijavljenih ispita za određeni predmet i rok.
2. Browser šalje zahtev serveru za spisak prijava iz *exam_applications* i povezanih studenata.
3. Server preuzima podatke iz baza *exam_applications*, *users* (studenti) i *courses*.
4. Baza vraća tražene podatke.
5. Profesor unosi ocenu za izabranog studenta i potvrđuje.
6. Browser šalje zahtev serveru sa vrednošću ocene.
7. Server kreira ili ažurira zapis u tabeli *grades* i po potrebi menja status u *exam_applications* na "položen" ili "nije položen".
8. Baza potvrđuje upis ocene.
9. Server vraća odgovor browseru i ažurira prikaz liste prijava sa unetim ocenama.

5.4. Process Flow (tok procesa) dijagram

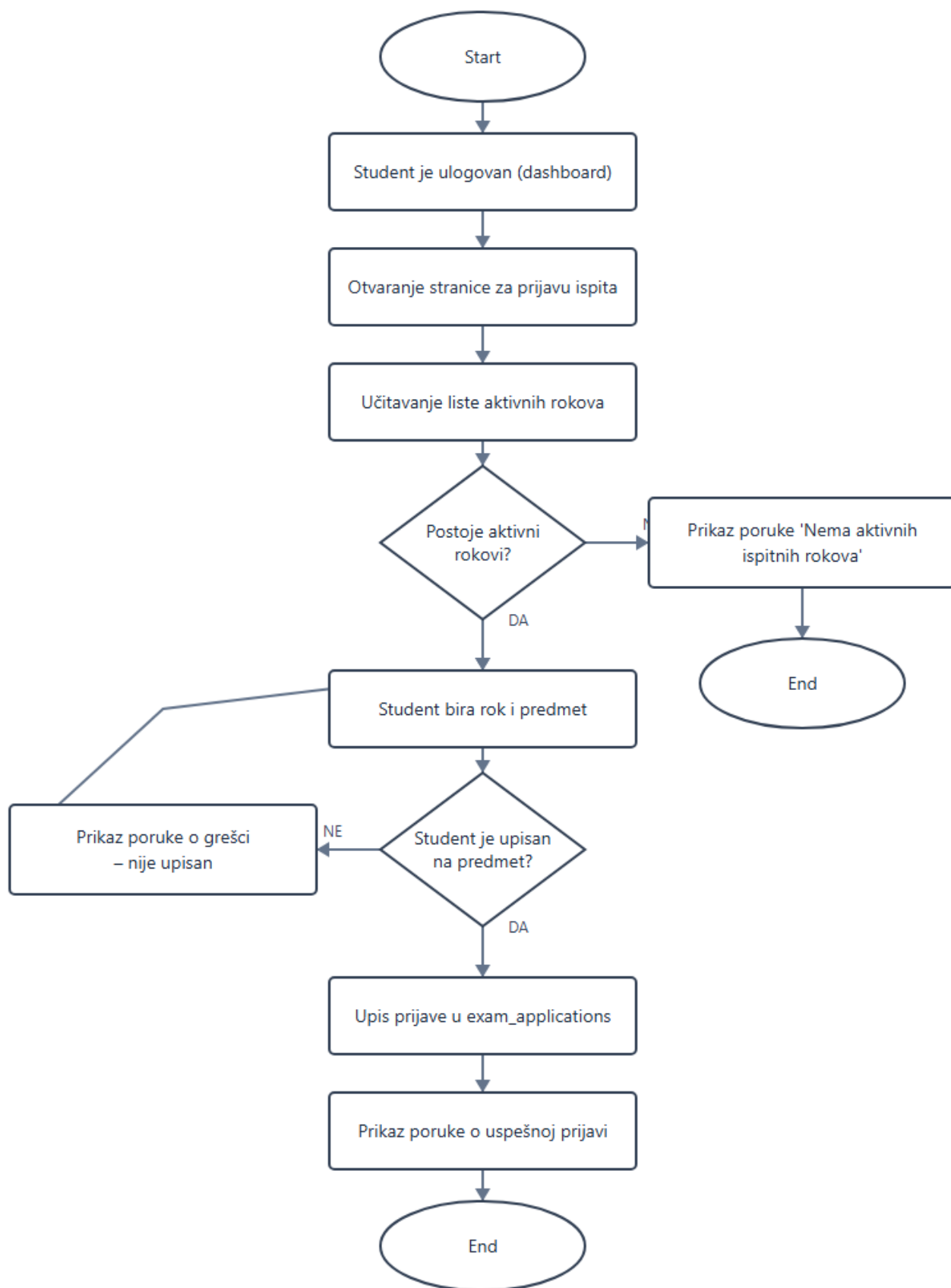
Process Flow dijagram vizuelno prikazuje tok akcija u okviru ključnog procesa u sistemu. U ovom projektu odabran je proces "**Prijava ispita**", jer obuhvata interakciju studenta sa aplikacijom, validaciju na serverskoj strani i upis podataka u bazu.

5.4.1. Tok procesa "Prijava ispita"

Dijagram toka procesa sastoji se od sledećih koraka:

1. **Start** - student je prijavljen u sistem (ulogovan).

2. **Otvaranje stranice za prijavu ispita** - student bira opciju "Prijava ispita" u meniju.
3. **Prikaz liste aktivnih rokova** - sistem proverava tabelu *exam_periods* i prikazuje samo rokove čiji je datum u rasponu *start_date-end_date*.
4. **Provera da li postoje aktivni rokovi** (romboid - odluka):
 - Ako **nema aktivnih rokova** - prikazuje se poruka "Trenutno nema aktivnih ispitnih rokova" i proces se završava.
 - Ako **postoje aktivni rokovi** - student bira rok i predmet.
5. **Provera da li je student upisan na predmet:**
 - Ako nije upisan - prikazuje se poruka o grešci i vraća se na izbor predmeta.
 - Ako jeste upisan - nastavlja se proces.
6. **Upis prijave u tabelu exam_applications** - sistem kreira novi zapis sa referencama na studenta, predmet i rok.
7. **Prikaz poruke o uspešnoj prijavi** - student dobija potvrdu da je prijava zabeležena.
8. **End** - proces je završen, student može da pregleda svoje prijave.



Slika 7 - Dijagram toka procesa „Prijava ispita“

6. Test plan i test case-ovi

U ovom poglavlju definisan je plan testiranja web aplikacije fakulteta i opisani su konkretni test case-ovi za tri korisničke uloge: **student**, **profesor** i **administrator**. Cilj testiranja je da se proveri da li ključne funkcionalnosti sistema rade ispravno, da li su validacije korektno implementirane i da li korisnici dobijaju očekivane poruke i rezultate za tipične i granične scenarije korišćenja.

6.1. Plan testiranja

Testiranje se sprovodi uglavnom kao **funkcionalno (black-box) testiranje**, pri čemu se fokus stavlja na posmatranje sistema iz ugla krajnjeg korisnika (student, profesor, admin), bez ulaska u detaljnu implementaciju koda.

Testovi se izvršavaju u sledećem okruženju:

- lokalni razvojni server ili hosting okruženje na kome je aplikacija deploy-ovana,
- aktuelna verzija web browsera (npr. Google Chrome, Mozilla Firefox),
- testni nalozi za svaku ulogu (barem po jedan student, jedan profesor, jedan administrator).

Test plan obuhvata:

- **pozitivne testove** - provera da li sistem radi ispravno kada korisnik unosi očekivane, validne podatke (npr. uspešno logovanje, pravilna prijava ispita u aktivnom roku),
- **negativne testove** - provera ponašanja sistema kada korisnik unese pogrešne ili nedozvoljene podatke (npr. prijava ispita van roka, unos ocene kada rok nije aktivan),

- **edge case scenarije** - situacije koje se ne dešavaju često, ali mogu da dovedu do greške ukoliko nisu predviđene (npr. brisanje predmeta na koji su studenti već upisani, dupli unos korisnika).

Za svaku ulogu definisano je najmanje **3 testa** (u ovom dokumentu ukupno 12), pri čemu svaki test case sadrži jasno definisane **korake, očekivani rezultat, stvarni rezultat i status**.

6.2. Struktura test case-a

- **Test Case ID:** TC-01
- **Opis testa:** Provera uspešnog logovanja
- **Uloga:** Student
- **Koraci:**
 1. Otvoriti login stranicu.
 2. Uneti ispravan email i lozinku studenta.
 3. Kliknuti na dugme "Login".
- **Očekivani rezultat:** Otvara se Student Dashboard.
- **Stvarni rezultat:** (popunjava se nakon izvršenog testa)
- **Status:** Passed / Failed

6.3. Test case-ovi - uloga: Student

Test Case ID	Opis testa	Uloga	Koraci	Očekivani rezultat	Stvarni rezultat	Status
TC-01	Uspešno logovanje studenta sa ispravnim kredencijalima.	Student	1. Otvoriti login stranicu. 2. Uneti ispravan email i lozinku registrovanog studenta. 3. Kliknuti na dugme "Login/Prijava".	Sistem uspešno autentifikuje studenta i otvara Studentski Dashboard.	Korisnik je uspešno ulogovan i prikazan je Studentski Dashboard, u skladu sa očekivanim rezultatom.	Passed
TC-02	Prijava ispita u aktivnom ispitnom roku.	Student	1. Ulogovati se kao student. 2. Otvoriti sekciju "Prijava ispita". 3. Izabrati aktivan ispitni rok. 4. Izabrati predmet na koji je student upisan. 5. Kliknuti na "Prijavi ispit".	Kreira se zapis u tabeli prijava ispita i ispit je vidljiv u sekciji "Moje prijave" sa statusom "prijavljen".	Prijava ispita je uspešno evidentirana i pojavljuje se na listi "Moje prijave" sa statusom "prijavljen", u skladu sa očekivanim rezultatom.	Passed
TC-03	Preuzimanje nastavnog materijala.	Student	1. Ulogovati se kao student. 2. Otvoriti "Materijali" ili karticu predmeta sa materijalima. 3. Kliknuti na odabrani materijal (npr. PDF) da bi se preuzeo ili otvorio.	Materijal se uspešno otvara ili preuzima bez poruke o grešci (npr. "fajl ne postoji").	Nastavni materijal je uspešno otvoren/preuzet bez greške, što je u skladu sa očekivanim ponašanjem.	Passed
TC-04	Pokušaj prijave ispita van aktivnog roka (negativni test).	Student	1. Ulogovati se kao student. 2. Otvoriti sekciju "Prijava ispita". 3. Pokušati da se prijavi ispit za rok koji nije aktivan (istekao je ili još nije počeo).	Sistem ne dozvoljava prijavu, prikazuje jasnu poruku (npr. "Rok nije aktivan") i ne kreira novi zapis prijave u bazi.	Sistem je odbio prijavu ispita za neaktivan rok i prikazao poruku o grešci, bez kreiranja nove prijave, u skladu sa očekivanim rezultatom.	Passed

Tabela 1 - Test case-ovi za ulogu „Student“

TC-01 - Uspešno logovanje studenta

- **Test Case ID:** TC-01
- **Opis testa:** Provera uspešnog logovanja studenta sa ispravnim kredencijalima.
- **Uloga:** Student
- **Koraci:**
 1. Otvoriti početnu stranicu aplikacije.

2. Kliknuti na link "Login" (ako postoji) ili direktno otvoriti login stranicu.
3. U polje "Email" uneti validnu email adresu registrovanog studenta.
4. U polje "Lozinka" uneti tačnu lozinku.
5. Kliknuti na dugme "Login".

- **Očekivani rezultat:**

1. Sistem uspešno autentifikuje studenta,
2. Otvara se Studentski Dashboard sa prikazom osnovnih informacija (predmeti, prijave, ocene).

- **Stvarni rezultat:** (popunjava se nakon testiranja)

- **Status:** Passed / Failed

TC-02 - Prijava ispita u aktivnom roku

- **Test Case ID:** TC-02

- **Opis testa:** Prijava ispita od strane studenta u ispitnom roku čiji je period aktivan.

- **Uloga:** Student

- **Koraci:**

1. Ulogovati se kao student (koristeći TC-01).
2. U glavnom meniju izabrati opciju "Prijava ispita" ili "Ispiti".
3. Proveriti prikaz liste aktivnih ispitnih rokova.
4. Odabrati jedan aktivan rok (npr. "Januarski 2025").
5. Iz liste predmeta odabrati predmet na koji je student upisan.
6. Kliknuti na dugme "Prijavi ispit" / "Potvrđi prijavu".

- **Očekivani rezultat:**

1. Sistem upisuje novu prijavu u tabelu *exam_applications* za datog studenta, predmet i rok,
2. Student dobija poruku o uspešnoj prijavi,

3. U sekciji "Moje prijave" novi ispit je vidljiv sa statusom "prijavljen".

- **Stvarni rezultat:** (popunjava se nakon testiranja)
- **Status:** Passed / Failed

TC-03 - Preuzimanje nastavnog materijala

- **Test Case ID:** TC-03
- **Opis testa:** Provera mogućnosti preuzimanja nastavnog materijala od strane studenta.
- **Uloga:** Student
- **Koraci:**
 1. Ulogovati se kao student.
 2. U meniju izabrati "Moji predmeti" / "Materijali".
 3. Odabrati predmet za koji postoji postavljen materijal.
 4. Naći stavku materijala (npr. "Predavanje 1 - Uvod.pdf").
 5. Kliknuti na link ili ikonicu za preuzimanje materijala.
- **Očekivani rezultat:**
 1. Materijal se uspešno preuzima (browser započinje download fajla ili otvara dokument),
 2. Ne prikazuje se poruka o grešci "fajl ne postoji" ili slična.
- **Stvarni rezultat:** (popunjava se nakon testiranja)
- **Status:** Passed / Failed

TC-04 - Pokušaj prijave ispita van aktivnog roka (negativni test)

- **Test Case ID:** TC-04
- **Opis testa:** Provera da student ne može da prijavi ispit u ispitnom roku koji više nije aktivan (istekao je) ili još nije počeo.
- **Uloga:** Student
- **Koraci:**

1. Ulogovati se kao student.
2. Otvoriti stranicu za "Prijavu ispita".
3. Pokušati da pristupiš ispitnom roku čiji datum više nije u opsegu *start_date-end_date* ili namerno podesiti datum sistema/baze tako da rok bude neaktivan.
4. Pokušati da klikneš na "Prijavi ispit" za taj rok/predmet (ako je UI uopšte prikazuje).

- **Očekivani rezultat:**

1. Sistem ne dozvoljava prijavu ispita,
2. Dugme za prijavu je onemogućeno ili se prikazuje jasna poruka o grešci (npr. "Rok nije aktivan"),
3. Ne nastaje novi zapis u tabeli *exam_applications* za neaktivan rok.

- **Stvarni rezultat**

- **Status:** Passed / Failed

6.4. Test case-ovi - uloga: Profesor

Test Case ID	Opis testa	Uloga	Koraci	Očekivani rezultat	Stvarni rezultat	Status
TC-05	Dodavanje novog nastavnog materijala za predmet.	Profesor	1. Ulogovati se kao profesor. 2. Otvoriti "Moji predmeti" i izabrati predmet. 3. Otvoriti sekciju "Materijali". 4. Kliknuti na "Dodaj materijal/Upload" i odabrati fajl. 5. Sačuvati materijal.	Materijal je upisan u bazu i pojavljuje se na listi materijala za izabrani predmet (vidljiv studentima).	Novi materijal se pojavljuje na listi materijala za predmet i studenti imaju pristup fajlu.	Passed
TC-06	Unos ocene za studenta u aktivnom ispitnom roku.	Profesor	1. Ulogovati se kao profesor. 2. Otvoriti listu prijavljenih ispita za predmet i aktivan rok. 3. Odabrati studenta i uneti ocenu. 4. Kliknuti na "Sačuvaj".	Kreira se/menja zapis u tabeli ocena, status prijave se postavlja na "položen" ili "nije položen", a lista se osvežava sa novom ocenom.	Ocena je uspešno snimljena, status prijave je ažuriran i lista prijava prikazuje novu ocenu.	Passed
TC-07	Slanje obaveštenja studentima određenog predmeta.	Profesor	1. Ulogovati se kao profesor. 2. Otvoriti sekciju "Obaveštenja". 3. Kliknuti na "Novo obaveštenje". 4. Uneti naslov, tekst i izabrati predmet. 5. Sačuvati obaveštenje.	Obaveštenje je upisano u bazu i vidljivo studentima upisanim na taj predmet u njihovom panelu.	Novo obaveštenje je vidljivo studentima upisanim na predmet i prikazuje se na listi obaveštenja.	Passed
TC-08	Pokušaj unosa ocene u neaktivan rok (negativni test).	Profesor	1. Ulogovati se kao profesor. 2. Otvoriti listu prijava za predmet i rok koji nije aktivan. 3. Pokušati da se unese ili promeni ocena.	Sistem ne dozvoljava unos ocene (polje je zaključano ili se prikazuje poruka o grešci); zapis u bazi se ne menja.	Sistem je blokirao unos ocene za neaktivan rok i nije menjao postojeće podatke, uz prikaz odgovarajuće poruke.	Passed

Tabela 2 – Test case-ovi za ulogu „Profesor“

TC-05 - Dodavanje novog materijala za predmet

- **Test Case ID:** TC-05
- **Opis testa:** Provera da profesor može da doda novi nastavni materijal za predmet na kome je nosilac.
- **Uloga:** Profesor
- **Koraci:**
 1. Ulogovati se kao profesor.

2. U meniju izabrati "Moji predmeti" ili "Materijali".
3. Odabrati predmet sa liste.
4. Kliknuti na opciju "Dodaj materijal" / "Upload".
5. Uneti naslov materijala (npr. "Vežbe 1 - Zadaci").
6. Odabrati fajl sa lokalnog računara (npr. PDF prezentaciju).
7. Kliknuti na "Sačuvaj" / "Upload".

- **Očekivani rezultat:**

1. Novi zapis se upisuje u tabelu *materials* sa ispravnim *course_id* i *uploaded_by* (ID profesora),
2. Materijal se pojavljuje na listi materijala za taj predmet, vidljiv i studentima.

- **Stvarni rezultat:** (popunjava se nakon testiranja)

- **Status:** Passed / Failed

TC-06 - Unos ocene u aktivnom roku

- **Test Case ID:** TC-06

- **Opis testa:** Provera da profesor može da unese ocenu za studenta koji je prijavio ispit u aktivnom ispitnom roku.

- **Uloga:** Profesor

- **Koraci:**

1. Ulogovati se kao profesor.
2. Otvoriti modul "Ocene" ili "Prijavljeni ispiti".
3. Izabrati predmet i aktivan ispitni rok.
4. Na listi prijavljenih studenata odabrati studenta kome se unosi ocena.
5. Uneti validnu vrednost ocene (npr. 8).
6. Kliknuti na "Sačuvaj" / "Unesi ocenu".

- **Očekivani rezultat:**

1. Kreira se ili ažurira zapis u tabeli *grades* sa unetom ocenom i datumom,
 2. Status prijave u *exam_applications* se menja u "položen" (za ocenu ≥ 6) ili "nije položen" (za ocenu 5),
 3. Profesor vidi ažurirane podatke na listi prijava.
- **Stvarni rezultat:** (popunjava se nakon testiranja)
 - **Status:** Passed / Failed

TC-07 - Slanje obaveštenja studentima

- **Test Case ID:** TC-07
- **Opis testa:** Provera da profesor može da pošalje obaveštenje studentima nekog predmeta.
- **Uloga:** Profesor
- **Koraci:**
 1. Ulogovati se kao profesor.
 2. Otvoriti sekciju "Obaveštenja".
 3. Kliknuti na "Novo obaveštenje".
 4. Uneti naslov (npr. "Odlaganje vežbi") i tekst poruke.
 5. Izabrati predmet na koji se obaveštenje odnosi (ako postoji takva opcija).
 6. Potvrditi slanje obaveštenja.
- **Očekivani rezultat:**
 1. Novi zapis se upisuje u tabelu *announcements* sa popunjenim *course_id*, *created_by* i *target_role* (najčešće "student"),
 2. Obaveštenje je vidljivo studentima upisanim na taj predmet na njihovom panelu.
- **Stvarni rezultat:** (popunjava se nakon testiranja)
- **Status:** Passed / Failed

TC-08 - Pokušaj unosa ocene van roka (negativni test)

- **Test Case ID:** TC-08
- **Opis testa:** Provera da profesor ne može da unese ocenu za prijavu u ispitnom roku koji nije aktivan.
- **Uloga:** Profesor
- **Koraci:**
 1. Ulogovati se kao profesor.
 2. Otvoriti listu prijava ispita za predmet i odabrati rok koji je istekao ili još nije aktivan.
 3. Pokušati da uneseš ili izmeniš ocenu za studenta prijavljenog u tom roku.
- **Očekivani rezultat:**
 1. Sistem onemogućava unos ocene (polje je zaključano ili dugme "Sačuvaj" nije dostupno),
 2. Ili se prikazuje jasna poruka o grešci (npr. "Nije dozvoljen unos ocena van aktivnog roka"),
 3. Ne menja se zapis u tabeli *grades* za tu prijavu.
- **Stvarni rezultat:** (popunjava se nakon testiranja)
- **Status:** Passed / Failed

6.5. Test case-ovi - uloga: Administrator

Test Case ID	Opis testa	Uloga	Koraci	Očekivani rezultat	Stvarni rezultat	Status
TC-09	Dodavanje novog korisnika u sistem.	Admin	1. Ulogovati se kao administrator. 2. Otvoriti 'Korisnici'. 3. Kliknuti na 'Dodaj korisnika'. 4. Uneti ime, email, lozinku i ulogu. 5. Sačuvati korisnika.	Novi korisnik se pojavljuje na listi korisnika i može da se uloguje zadatim kredencijalima.	Novi korisnik je uspešno dodat i vidljiv je na listi korisnika.	Passed
TC-10	Kreiranje novog ispitnog roka.	Admin	1. Ulogovati se kao admin. 2. Otvoriti 'Ispitni rokovi'. 3. Kliknuti na 'Novi ispitni rok'. 4. Uneti naziv roka i datume početka i kraja. 5. Sačuvati rok.	Rok je dodat u listu ispitnih rokova i vidljiv studentima kao aktivan u odabranom periodu.	Novi ispitni rok se pojavljuje na listi rokova i studenti ga vide u delu za prijavu ispita tokom definisanog perioda.	Passed
TC-11	Brisanje predmeta koji ima povezane podatke (upisi, prijave, materijale).	Admin	1. Ulogovati se kao admin. 2. Otvoriti 'Predmeti'. 3. Izabrati predmet koji ima povezane upise/prijave/materijale. 4. Kliknuti na 'Obriši predmet' i potvrditi akciju.	Sistem se ponaša prema definisanim ON DELETE pravilima (brisanje povezanih zapisa ili blokiranje uz poruku); baza ostaje u konzistentnom stanju.	Brisanje predmeta je izvršeno u skladu sa podešenim pravilima (bez 'visećih' podataka ili uz jasnu poruku o blokiranju), i baza je ostala konzistentna.	Passed
TC-12	Izmena uloge postojećeg korisnika.	Admin	1. Ulogovati se kao admin. 2. Otvoriti 'Korisnici'. 3. Naći korisnika i kliknuti na 'Izmeni'. 4. Promeniti polje 'Uloga' (npr. sa student na profesor). 5. Sačuvati izmene. 6. Ulogovati se kao taj korisnik i proveriti da li vidi odgovarajući panel.	Uloga korisnika u bazi je izmenjena i nakon ponovnog logovanja korisnik dobija novi dashboard koji odgovara dodeljenoj ulozi.	Nakon izmene uloge i ponovnog logovanja, korisniku se prikazuje odgovarajući dashboard.	Passed

Tabela 3 - Test case-ovi za ulogu „Administrator“

TC-09 - Dodavanje novog korisnika u sistem

- **Test Case ID:** TC-09
- **Opis testa:** Provera da administrator može da doda novog korisnika (studenta ili profesora) u sistem.
- **Uloga:** Admin
- **Koraci:**

1. Ulogovati se kao administrator.
2. Otvoriti sekciju "Korisnici".
3. Kliknuti na "Dodaj korisnika".
4. Uneti ime, prezime, email, početnu lozinku i ulogu (npr. "student").
5. Kliknuti na "Sačuvaj" / "Kreiraj".

- **Očekivani rezultat:**

1. Novi zapis se pojavljuje u tabeli *users* sa unetim podacima i dodeljenom ulogom,
2. Na listi korisnika vidi se novi nalog,
3. Novi korisnik može da se uloguje koristeći zadate kredencijale.

- **Stvarni rezultat:** (popunjava se nakon testiranja)

- **Status:** Passed / Failed

TC-10 - Kreiranje ispitnog roka

- **Test Case ID:** TC-10

- **Opis testa:** Provera da administrator može da kreira novi ispitni rok.

- **Uloga:** Admin

- **Koraci:**

1. Ulogovati se kao administrator.
2. Otvoriti sekciju "Ispitni rokovi".
3. Kliknuti na "Novi ispitni rok".
4. Uneti naziv roka (npr. "Jun 2025"), datum početka i kraja.
5. Kliknuti na "Sačuvaj".

- **Očekivani rezultat:**

1. Novi zapis se upisuje u tabelu *exam_periods* sa unetim parametrima,
2. Rok se pojavljuje na listi ispitnih rokova,

3. Studenti vide novi rok kao aktivan u periodu *start_date-end_date*.

- **Stvarni rezultat:** (popunjava se nakon testiranja)
- **Status:** Passed / Failed

TC-11 - Brisanje predmeta i provera ON DELETE CASCADE funkcionalnosti

- **Test Case ID:** TC-11
- **Opis testa:** Provera ponašanja sistema pri brisanju predmeta koji je povezan sa drugim tabelama (*enrollments*, *exam_applications*, *materials*, *announcements*).
- **Uloga:** Admin
- **Koraci:**
 1. Ulogovati se kao administrator.
 2. Uveriti se da postoji predmet koji ima:
 - upisane studente (*enrollments*),
 - prijave ispita (*exam_applications*),
 - barem jedan materijal (*materials*).
 3. U sekciji "Predmeti" odabrati taj predmet.
 4. Kliknuti na "Obriši predmet" i potvrditi brisanje.
- **Očekivani rezultat:**
 1. U skladu sa podešenim *ON DELETE* pravilima, sistem se ponaša predvidivo (npr. brišu se povezani zapisi ili se brisanje blokira uz poruku),
 2. Ne dolazi do nekonzistentnog stanja baze (npr. "visećih" prijava bez predmeta),
 3. Ako je implementiran *ON DELETE CASCADE*, povezani zapisi se brišu automatski; ako nije, aplikacija baca jasnu grešku i ne dozvoljava brisanje.
- **Stvarni rezultat:** (popunjava se nakon testiranja)
- **Status:** Passed / Failed

TC-12 - Izmena uloge postojećeg korisnika

- **Test Case ID:** TC-12
- **Opis testa:** Provera da administrator može da izmeni ulogu postojećeg korisnika (npr. sa "student" na "admin" ili sa "profesor" na "student" u test okruženju).
- **Uloga:** Admin
- **Koraci:**
 1. Ulogovati se kao administrator.
 2. Otvoriti sekciju "Korisnici".
 3. Naći postojećeg korisnika na listi (prema email adresi).
 4. Kliknuti na "Izmeni" / "Edit".
 5. Promeniti polje "Uloga" na novu vrednost.
 6. Sačuvati izmene.
 7. (Opcionalno) Odjaviti se i pokušati logovanje kao taj korisnik da bi se proverio novi dashboard.
- **Očekivani rezultat:**
 1. U tabeli *users* promenjena je vrednost polja *role*,
 2. Na sledećem logovanju, korisniku se prikazuje dashboard koji odgovara novoj ulozi (student/profesor/admin),
 3. Ne dolazi do greške u navigaciji (nema pristupa sekcijama kojima nova uloga ne bi smela da pristupi).
- **Stvarni rezultat:** (popunjava se nakon testiranja)
- **Status:** Passed / Failed

6.6. Zaključak sekcije o testiranju

Definisani test case-ovi pokrivaju ključne funkcionalnosti aplikacije za sve tri korisničke uloge: logovanje, rad sa ispitnim rokovima, materijalima, ocenama, obaveštenjima i administrativnim operacijama nad korisnicima i predmetima. Kroz kombinaciju pozitivnih i negativnih scenarija, omogućen je osnovni nivo

sigurnosti da sistem radi u skladu sa očekivanjima, a ujedno su identifikovane i potencijalne tačke za dalje unapređenje validacija i poruka o greškama.

7. Tehnologije i resursi

U ovom poglavlju navode se sve **tehnologije, alati i izvori** koji su korišćeni prilikom izrade web aplikacije fakulteta i njene dokumentacije. Poseban akcenat je na backend i frontend tehnologijama, bazi podataka, alatima za crtanje dijagrama, kao i tutorijalima, sajtovima sa ikonama, slikama, fontovima i bojama koji su korišćeni tokom razvoja.

Pored toga, jasno se navode i **korišćeni AI alati**, uz napomenu da oni nisu bili glavni izvor znanja, već pomoćni alat uz zvaničnu dokumentaciju, kurseve i udžbenike.

7.1. Backend tehnologije

Za serversku stranu aplikacije korišćen je **PHP** kao glavni programski jezik, u kombinaciji sa **Apache** web serverom (lokalno kroz XAMPP) i MySQL bazom podataka.

Glavne karakteristike backend dela:

- **PHP 8.x** - korišćen za implementaciju poslovne logike aplikacije (logovanje korisnika, rad sa prijavama ispita, unos ocena, rad sa materijalima, obaveštenjima itd.).
- **Struktura nalik MVC-u** - odvajanje PHP skripti po logici (kontroleri), prikazima (view fajlovi) i sloju za pristup bazi (posebne funkcije/klase za SQL upite), što olakšava održavanje i nadogradnju.
- **Rad sa sesijama** - PHP sesije se koriste za čuvanje informacija o ulogovanom korisniku (ID, uloga), kako bi sistem razlikovao studente, profesore i administratore.
- **Validacija i sanitizacija podataka** - pre upisa u bazu, podaci iz formi prolaze osnovnu validaciju (obavezna polja, formati email adrese itd.), a

SQL upiti koriste pripremljene iskaze (*prepared statements*) ili pravilno escapovane vrednosti da bi se smanjio rizik od SQL injection-a.

7.2. Frontend tehnologije

Za korisnički interfejs (frontend) korišćene su sledeće tehnologije:

- **HTML5** - za semantičku strukturu stranica (naslovi, forme, tabele, liste).
- **CSS3** - za osnovno stilizovanje izgleda stranica, raspored elemenata, boje, margine, fontove.
- **Bootstrap 5** - CSS framework koji obezbeđuje responzivan dizajn i gotove komponente (navbar, kartice, dugmad, tabele). U projektu je Bootstrap iskorišćen za:
 - layout admin, profesorskog i studentskog panela,
 - tabele sa podacima (korisnici, ispiti, predmeti, prijave),
 - navigacioni meni i kartice na dashboardu,
 - osnovnu tipografiju i raspored grid sistema.
- **JavaScript (vanilla)** - korišćen za jednostavne interaktivnosti na klijentskoj strani, poput potvrda prilikom brisanja, validacije formi na frontendu i eventualnog osvežavanja sadržaja bez kompletnog reloada stranice.

7.3. Baza podataka

Za skladištenje podataka aplikacija koristi **relacionu bazu podataka MySQL**.

Korišćeni elementi:

- **MySQL server** - čuva tabele: users, courses, enrollments, exam_periods, exam_applications, grades, materials, announcements i eventualne pomoćne tabele.
- **phpMyAdmin** - grafički alat za administraciju baze (kreiranje tabela, indeksiranje, pokretanje SQL upita, backup/export).
- **Relacije i ograničenja** - primarni i strani ključevi postavljeni su u skladu sa ER dijagramom, uz ograničenja poput **ON DELETE** pravila (npr. zabrana brisanja predmeta ako postoje povezane prijave ispita, ili kaskadno brisanje ako je tako definisano).

7.4. Alati za izradu dijagrama

Za modelovanje i vizuelizaciju sistema korišćeni su online alati za dijagrame:

- **Draw.io (diagrams.net)** - glavni alat za crtanje:
 - ER dijagrama baze podataka,
 - Use Case dijagrama,
 - sekvencijalnih dijagrama,
 - process flow (flowchart) dijagrama za tok "Prijava ispita".

Svi dijagrami su eksportovani kao slike (PNG/JPEG) i ubačeni u dokument, uz naziv slike (npr. "*Slika 3 - Use Case dijagram*") i kratak opis ispod, u skladu sa uputstvom profesora.

7.5. Razvojno okruženje i deploy

Razvoj aplikacije i njeno testiranje rađeni su u sledećem okruženju:

- **Visual Studio Code** - glavni editor za pisanje koda (PHP, HTML, CSS, JS), uz ekstenzije za PHP, IntelliSense, i podršku za Git.
- **XAMPP (Apache + MySQL + PHP)** - lokalno razvojno okruženje za pokretanje aplikacije na *localhost*-u.
- **Git i GitHub** - verzionisanje koda, čuvanje repozitorijuma i mogućnost timskog rada.
- **Hostinger hosting nalog** - produkciono/deploy okruženje gde je aplikacija postavljena kako bi profesor mogao da joj pristupi putem URL-a
- **phpMyAdmin na Hostingeru ili lokalno** - za administraciju baze u produkcijom ili razvojnom okruženju.

7.6. Tutorijali, dokumentacija i kursevi

Tokom izrade aplikacije i dokumentacije korišćeni su sledeći tipovi izvora:

- **Zvanična dokumentacija:**
 - PHP Manual - za rad sa sesijama, cookie-jima, rad sa bazom (PDO/MySQLi), hashovanje lozinki.
 - MySQL dokumentacija - za definisanje tabela, indeksa, ograničenja i relacija.
 - Bootstrap dokumentacija - primeri komponenti (navbar, cards, tabele, forme).
 - Dokumentacija za korišćene JS biblioteke (ako postoji DataTables ili sličan plugin).

- **Online tutorijali i kursevi:**

- Serije tutorijala za *PHP* + *MySQL* (npr. "PHP login system with MySQL", "PHP CRUD tutorial") - korišćeni kao referenca za strukturu login sistema i rad sa bazom.
- Tutorijali za Bootstrap - za izradu responzivnog admin panela, grid sistem i stilizovanje tabela.
- Video materijali iz oblasti ER dijagrama i Use Case dijagrama - kao pomoć pri pravilnom modelovanju.

- **Reference za web tehnologije:**

- MDN Web Docs - za detaljne informacije o HTML5, CSS3 i JavaScript-u.
- W3Schools - kao brzi referentni priručnik za sintaksu i jednostavne primere.

7.7. Sajтови za ikonice, slike, fontove i boje

Za vizuelni deo aplikacije korišćeni su sledeći resursi:

- **Ikonice:**

- *Flaticon* - izvor vektorskih ikonica korišćenih u UI-u (npr. ikonice za korisnike, predmete, ispite).

- **Slike:**

- *Pexels* - besplatne stock fotografije za ilustraciju početne stranice ili promotivnih sekcija.
- *Pixabay* - dodatni izvor slika bez autorskih ograničenja.

- **Boje i paleta:**

- *Coolors.co* - korišćen za generisanje i odabir kolor palete (primarna, sekundarna boja, boje pozadine i naglašavanja).
- Kombinacija boja prilagođena je tako da bude čitljiva i dosledna u svim panelima (admin, profesor, student).
- **Fontovi:**
 - *Google Fonts* - izabran je jedan ili dva web fonta (npr. "Roboto" za tekst i "Poppins" za naslove), koji su uvezeni u CSS i korišćeni kroz celu aplikaciju.

U dokumentu je važno navesti **naziv sajta i svrhu** (ikonice, slike, boje, fontovi), čak i ako je korišćen mali broj resursa, zbog kriterijuma "Tehnologije i alati".

7.8. Korišćeni AI alati

Tokom rada na projektu korišćeni su i **AI alati** kao pomoć, pre svega za:

- generisanje ideja za organizaciju koda i strukturu baze,
- dobijanje predloga za test case-ove i edge scenarije,
- formulisanje delova tekstualne dokumentacije (uz kasniju proveru i prilagođavanje),
- brzu proveru sintakse i pronalaženje bagova u PHP/SQL kodu.

Primer korišćenog AI alata:

- **ChatGPT (OpenAI)** - korišćen kao asistent pri razmišljanju o arhitekturi, dijagramima, testiranju i pisanju dokumentacije, ali ne kao jedini izvor znanja.

U skladu sa uputstvom profesora, AI alati **nisu korišćeni kao glavni izvor gradiva**, već kao podrška uz stručne izvore (dokumentaciju, tutorijale, knjige, kurseve i nastavne materijale sa predavanja i vežbi).

8. User Manual (uputstvo za korišćenje)

U ovom poglavlju opisano je kako krajnji korisnici (studenti, profesori i administratori) koriste web aplikaciju fakulteta. Uputstvo pokriva pristup aplikaciji, proces logovanja i odjave, osnovnu navigaciju kroz interfejs, kao i tipične scenarije rada za svaku korisničku ulogu.

8.1. Pristup aplikaciji

Web aplikacija fakulteta je dostupna putem web browsera (Google Chrome, Mozilla Firefox, Microsoft Edge itd.).

Da bi korisnik pristupio aplikaciji, potrebno je da:

1. Otvori web browser.
2. U adresnu liniju unese URL aplikacije (npr. `http://localhost/fakultet` u lokalnom okruženju ili odgovarajući domen na Hostinger hostingu).
3. Nakon učitavanja početne stranice, korisnik može da ode na login stranicu klikom na dugme/link "Prijava" / "Login" ili direktnim odlaskom na rutu za logovanje (npr. `/login`).

Aplikacija ne zahteva posebnu instalaciju na računaru korisnika - sve funkcionalnosti dostupne su direktno kroz browser.

8.2. Logovanje i odjava korisnika

8.2.1. Logovanje

Logovanje je obavezno za sve korisnike koji žele da pristupe zatvorenom delu sistema (studenti, profesori, administratori).

Koraci za logovanje:

1. Otvoriti login stranicu aplikacije.
2. U polje **Email** uneti registrovanu email adresu.
3. U polje **Lozinka** uneti lozinku dodeljenu pri kreiranju naloga.
4. Kliknuti na dugme "**Prijava**" / "**Login**".

Ukoliko su podaci tačni:

- student biva preusmeren na **Studentski Dashboard**,
- profesor na **Profesorski Dashboard**,
- administrator na **Admin panel**.

Ukoliko su podaci netačni:

- aplikacija prikazuje poruku o grešci (npr. "Pogrešan email ili lozinka"),
- korisnik ostaje na login stranici i može ponovo da pokuša da se uloguje.

8.2.2. Odjava (Logout)

Za odjavu iz sistema:

1. U gornjem delu stranice (najčešće na desnoj strani navbar-a) kliknuti na dugme "**Odjava**" / "**Logout**".
2. Aplikacija briše aktivnu sesiju korisnika i vraća ga na početnu stranicu ili login formu.

Preporučuje se da se korisnici uvek odjave sa sistema nakon završetka rada, posebno kada koriste javne računare (biblioteka, internet klub itd.).

8.3. Navigacija i zajednički elementi interfejsa

Nakon logovanja, interfejs aplikacije prilagođava se ulozi korisnika. Ipak, određeni elementi su zajednički za sve:

- **Gornja navigaciona traka (navbar)**
 - prikazuje naziv aplikacije ili logo fakulteta,
 - sadrži linkove ka glavnim sekcijama (npr. "Početna", "Moji predmeti", "Ispiti", "Obaveštenja"),
 - na desnoj strani obično sadrži ime ulogovanog korisnika i dugme "Logout".
- **Levi meni (sidebar) (ako postoji)**
 - dodatna navigacija kroz specifične module (npr. "Korisnici", "Predmeti", "Ispitni rokovi" za admina).
- **Dashboard**
 - početna stranica nakon logovanja,
 - prikazuje najvažnije informacije za konkretnu ulogu: broj predmeta, aktuelne ispite, poslednja obaveštenja, osnovnu statistiku itd.
- **Tabele i forme**
 - podaci se uglavnom prikazuju u tabelama (korisnici, predmeti, prijave, ocene),
 - iznad ili ispod tabela nalaze se dugmad za dodavanje, izmenu ili brisanje zapisa,

- forme (login, dodavanje korisnika, unos ocena) sastoje se od polja za unos teksta, padajućih lista i dugmadi "Sačuvaj", "Otkazi" i sl.

8.4. Korišćenje aplikacije - uloga: Student

Studenti koriste aplikaciju da bi:

- videli predmete na koje su upisani,
- prijavili ispite,
- pratili svoje prijave i ocene,
- preuzimali nastavne materijale,
- čitali obaveštenja relevantna za svoje predmete i studije.

8.4.1. Pregled predmeta ("Moji predmeti")

1. Nakon logovanja kao student, otvoriti sekciju **"Moji predmeti"**.
2. Prikazuje se tabela sa spiskom predmeta na koje je student upisan (naziv, šifra, godina, profesor).
3. Klikom na određeni predmet (ili dugme "Detalji", ako postoji) student može da vidi dodatne informacije, obaveštenja ili materijale vezane za taj predmet.

8.4.2. Prijava ispita

1. U studentskom panelu izabrati opciju **"Prijava ispita"** / "Ispiti".
2. Sistem prikazuje listu **aktivnih ispitnih rokova** i predmeta koje student sluša.
3. Student bira:

- željeni ispitni rok (npr. "Januarski 2025"),
- predmet koji želi da prijavi.

4. Klikom na dugme "**Prijavi ispit**" potvrđuje prijavu.

5. Nakon uspješne prijave, sistem prikazuje poruku o uspehu, a prijavljeni ispit se pojavljuje u sekciji "**Moje prijave**" sa statusom "prijavljen".

Ako je rok neaktivan ili student nije upisan na predmet, aplikacija vraća poruku o grešci i ne kreira prijavu.

8.4.3. Pregled prijava i ocena

- **Moje prijave**

1. Otvoriti sekciju "**Moje prijave**".
2. Prikazuje se lista svih prijavljenih ispita (rok, predmet, datum prijave, status).
3. Student ovde može da proveri da li je ispit i dalje prijavljen ili je eventualno otkazan / obrađen.

- **Ocene**

1. Otvoriti sekciju "**Ocene**".
2. Prikazuju se svi položeni ispiti sa informacijama o predmetu, datumu polaganja i oceni.
3. Po želji, može da se prikaže i prosečna ocena kao dodatna informacija.

8.4.4. Preuzimanje nastavnih materijala

1. U studentskom panelu izabrati "**Materijali**" ili u okviru "Mojih predmeta" kliknuti na opciju "Materijali" za konkretan predmet.
2. Prikazuje se lista dostupnih fajlova (naziv materijala, datum postavljanja).
3. Klikom na naziv materijala ili dugme "Preuzmi", fajl se otvara ili preuzima na računar korisnika.

8.4.5. Pregled obaveštenja

1. Izabrati sekciju "**Obaveštenja**" u studentskom panelu.
2. Na listi su prikazana:
 - opšta obaveštenja (za sve studente),
 - obaveštenja vezana za predmete na koje je student upisan.
3. Svako obaveštenje sadrži naslov, datum i sadržaj poruke.

8.5. Korišćenje aplikacije - uloga: Profesor

Profesori koriste aplikaciju da bi:

- upravljali predmetima za koje su zaduženi,
- postavljali nastavne materijale,
- unosili i ažurirali ocene,
- slali obaveštenja studentima.

8.5.1. Pregled i izbor predmeta

1. Nakon logovanja kao profesor, otvoriti sekciju "**Moji predmeti**".

2. Prikazuje se lista predmeta za koje je profesor nosilac.
3. Klikom na predmet otvaraju se dodatne opcije: prikaz studenata, materijali, prijavljeni ispiti itd.

8.5.2. Upravljanje nastavnim materijalima

1. U okviru izabranog predmeta ili kroz sekciju "**Materijali**", profesor vidi listu postojećih materijala.
2. Za dodavanje novog materijala:
 - kliknuti na dugme "**Dodaj materijal**" / "**Upload**",
 - uneti naslov materijala (npr. "Predavanje 3 - Ponavljanje"),
 - odabrati fajl sa računara,
 - potvrditi upload klikom na "Sačuvaj" / "Upload".
3. Profesor može:
 - izmeniti naslov materijala,
 - obrisati materijal koji više nije aktuelan.

8.5.3. Unos i izmena ocena

1. Otvoriti sekciju "**Ocene**" ili "Prijavljeni ispiti".
2. Izabrati predmet i ispitni rok.
3. Sistem prikazuje listu studenata koji su prijavili ispit u tom roku.
4. Za svakog studenta profesor može:
 - uneti novu ocenu,

- izmeniti postojeću ocenu (ako je sistem to dozvoljava),
- sačuvati izmene klikom na dugme "Sačuvaj".

5. Nakon čuvanja, ocena postaje vidljiva studentu u njegovom panelu ("Ocene"), a status prijave se ažurira (položen / nije položen).

8.5.4. Slanje obaveštenja studentima

1. Otvoriti sekciju "**Obaveštenja**" u profesorskom panelu.
2. Kliknuti na "**Novo obaveštenje**".
3. Uneti:
 - naslov obaveštenja,
 - sadržaj poruke,
 - predmet kome je obaveštenje namenjeno (ako je obaveštenje vezano za određeni predmet).
4. Kliknuti na "Pošalji" / "Sačuvaj".
5. Obaveštenje postaje vidljivo studentima upisanim na taj predmet (ili svim studentima, u zavisnosti od podešavanja).

8.6. Korišćenje aplikacije - uloga: Administrator

Administrator ima najviši nivo privilegija u sistemu. Njegove glavne obaveze su:

- upravljanje korisnicima,
- upravljanje predmetima,
- upravljanje ispitnim rokovima,
- objava opštih obaveštenja,

- pregled statistike.

8.6.1. Upravljanje korisnicima

1. Nakon logovanja kao admin, otvoriti sekciju "**Korisnici**".
2. Prikazuje se tabela sa svim korisnicima sistema (ime, email, uloga, indeks za studente itd.).
3. Admin može:
 - **dodati novog korisnika** - klikom na "Dodaj korisnika", popunjavanjem forme (ime, email, lozinka, uloga) i čuvanjem,
 - **izmeniti korisnika** - klikom na "Izmeni" pored konkretnog korisnika i izmenom podataka,
 - **obrisati korisnika** - klikom na "Obriši" (po potrebi uz dodatnu potvrdu da bi se sprečilo slučajno brisanje).

8.6.2. Upravljanje predmetima

1. Otvoriti sekciju "**Predmeti**".
2. Tabela prikazuje postojeće predmete (naziv, šifra, godina, profesor nosilac).
3. Admin može:
 - **dodati predmet** - unosom naziva, šifre, godine, ECTS bodova i dodeljivanjem profesora,
 - **izmeniti predmet** - prilagoditi naziv, nosioca ili druge podatke,
 - **obrisati predmet** - u skladu sa pravilima baze (paziti na postojeće upise, prijave i materijale).

8.6.3. Upravljanje ispitnim rokovima

1. Otvoriti sekciju "**Ispitni rokovi**".
2. Tabela prikazuje listu rokova (naziv, datum početka, datum završetka).
3. Admin može:
 - **kreirati novi rok** - unosom naziva (npr. "Septembar 2025") i opsega datuma,
 - **izmeniti postojeći rok** - npr. prilagoditi datume,
 - **obrisati rok** - ako još uvek nije korišćen ili u skladu sa pravilima sistema.

Aktivni rokovi su potom dostupni studentima u delu za prijavu ispita.

8.6.4. Objavljivanje opštih obaveštenja

1. Otvoriti sekciju "**Obaveštenja**" u admin panelu.
2. Kliknuti na "Novo obaveštenje".
3. Uneti:
 - naslov,
 - sadržaj poruke,
 - ciljnu grupu (svi, samo studenti, samo profesori).
4. Sačuvati obaveštenje.
5. Obaveštenje će biti prikazano u panelima korisnika kojima je namenjeno.

8.6.5. Pregled statistike

1. Otvoriti sekciju "**Statistika**" (ako je implementirana).

2. Administrator može da vidi:

- broj registrovanih korisnika po ulozi,
- broj predmeta i aktivnih rokova,
- broj prijavljenih ispita u određenom roku,
- druge indikatore uspešnosti i aktivnosti sistema.

8.7. Tipične poruke o greškama i preporuke za korisnike

Tokom korišćenja aplikacije korisnici se mogu susresti sa porukama o greškama. Najčešće situacije su:

- **Pogrešan email ili lozinka** - javlja se pri neuspešnom logovanju; potrebno je proveriti da li je email tačan i da li je uključen Caps Lock.
- **Nemate dozvolu za pristup ovoj stranici** - javlja se kada korisnik pokuša da pristupi delu sistema koji nije namenjen njegovoj ulozi (npr. student pokušava da otvori admin stranicu).
- **Rok nije aktivan** - prikazuje se ako student pokuša da prijavi ispit van datuma trajanja ispitnog roka.
- **Obavezna polja nisu popunjena** - prilikom čuvanja formi (kreiranje korisnika, predmeta, roka itd.) ukoliko nisu unesene sve neophodne informacije.

Preporuke za korisnike:

- uvek proveriti da li je ulogovana prava uloga (student/profesor/admin),
- nakon rada na javnim računarima obavezno se odjaviti,
- koristiti moderan browser i redovno osvežavati stranicu ako dođe do vizuelnih problema,
- u slučaju neočekivanih grešaka kontaktirati administratora sistema.

9. Design Patterns i organizacija koda

U ovom poglavlju opisani su osnovni dizajn paterni i principi organizacije koda koji su primenjeni u web aplikaciji fakulteta. Cilj je da se pokaže da sistem nije samo skup nasumičnih PHP skripti, već da iza implementacije postoji jasna arhitektura koja olakšava održavanje, testiranje i dalji razvoj.

Poseban naglasak je na **MVC (Model-View-Controller)** i **Singleton** paternu, kao i na organizaciji direktorijuma i ponovnoj upotrebi delova interfejsa (šabloni za header, footer i navigaciju).

9.1. MVC (Model-View-Controller) arhitektura

MVC (Model-View-Controller) je jedan od najčešće korišćenih paterna u modernim web framework-ovima (Laravel, ASP.NET, Express.js itd.) i predstavlja način da se razdvoje podaci (Model), logika aplikacije (Controller) i prikaz (View).

Iako aplikacija nije rađena u okviru kompletnog framework-a, njena struktura je organizovana tako da **prati osnovne principe MVC-a**:

- **Model (podaci i baza)**
 - Podaci se čuvaju u MySQL bazi u tabelama `users`, `courses`, `enrollments`, `exam_periods`, `exam_applications`, `grades`, `materials`, `announcements`.
 - U posebnom PHP fajlu (npr. `db.php` ili `database.php`) nalazi se kod koji obezbeđuje konekciju ka bazi, kao i funkcije/metode za čitanje i upis podataka (npr. `getAllCourses()`, `getStudentExams($studentId)`, `insertExamApplication(...)`).
- **Controller (poslovna logika)**
 - Za svaku ulogu postoje PHP skripte koje primaju HTTP zahteve, obrađuju unos, pozivaju funkcije za rad sa bazom i odlučuju koji view da se prikaže.

- Primer: skripta koja obrađuje prijavu ispita:
 - proverava da li je student ulogovan,
 - proverava da li je rok aktivan,
 - poziva Model funkciju koja upisuje prijavu u `exam_applications`,
 - nakon uspeha redirektuje na stranicu sa listom prijava.
- **View (prikaz korisniku)**
 - HTML i Bootstrap kod se nalazi u posebnim datotekama (npr. `student_dashboard.php`, `admin_users.php`), uz include zajedničkih delova interfejsa (header, footer, navigacija).
 - View ne sadrži direktno SQL upite, već samo prikazuje podatke koje dobija iz Controller-a (npr. niz predmeta, niz prijava ispita).

Ovakva podela olakšava:

- menjanje dizajna (View) bez diranja logike,
- promenu načina čuvanja podataka (Model) bez promene UI-a,
- čitljivost i održavanje koda, jer se tačno zna gde se šta nalazi.

9.2. Singleton pattern - konekcija ka bazi podataka

Singleton je dizajn pattern koji obezbeđuje da u aplikaciji postoji **tačno jedna instanca** određenog objekta i da joj se pristupa centralizovano. U web aplikacijama se često koristi za konekciju ka bazi podataka, kako bi se izbeglo otvaranje više nepotrebnih konekcija.

U ovoj aplikaciji, logika za konekciju ka MySQL bazi je izdvojena u poseban fajl (npr. `db.php`). Konceptualno, taj fajl se ponaša kao Singleton, jer se:

- konekcija ka bazi uspostavlja **na jednom mestu**,

- isti objekat konekcije se ponovo koristi u svim skriptama koje uključe taj fajl (include / require),
- izbegava se dupliranje koda za `mysqli_connect` / `new PDO(...)` po različitim fajlovima.

```
class Database {
    private static $instance = null;
    private $connection;

    private function __construct() {
        $this->connection = new PDO(
            "mysql:host=localhost;dbname=fakultet;charset=utf8",
            "root",
            ""
        );
        $this->connection->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
    }

    public static function getInstance() {
        if (self::$instance === null) {
            self::$instance = new Database();
        }
        return self::$instance;
    }

    public function getConnection() {
        return $this->connection;
    }
}
```

Korišćenje:

```
$db = Database::getInstance()->getConnection();
// dalje se koristi $db za SELECT/INSERT/UPDATE upite
```

Čak i ako je u projekatu korišćen jednostavniji pristup (npr. globalna `$conn` promenljiva iz `db.php` fajla), ideja je ista: **jedna centralna konekcija ka bazi**

koja se potom deli među svim delovima sistema, umesto da se na svakoj stranici pravi nova veza.

9.3. Šabloni (template) i ponovna upotreba view-ova

Još jedan važan aspekt organizacije koda je korišćenje **šablona (template)** za zajedničke delove interfejsa:

- **header** (gornji deo stranice, logo, naslov, glavna navigacija),
- **sidebar** (levi meni u admin/professor panelu, ako postoji),
- **footer** (donji deo stranice).

Ovi delovi se nalaze u posebnim fajlovima (npr. `includes/header.php`, `includes/sidebar.php`, `includes/footer.php`) i **ubacuju se** na svaku stranicu pomoću `include / require`.

Primer strukture:

```
<?php include 'includes/header.php'; ?>
```

```
<h1>Pregled korisnika</h1>  
<!-- tabela korisnika -->
```

```
<?php include 'includes/footer.php'; ?>
```

Na ovaj način se postiže:

- **konzistentan izgled** kroz ceo sajt (isti meni, ista boja pozadine, isti footer),
- **lakše održavanje** - izmena u `header.php` se automatski vidi na svim stranicama,
- izbegava se dupliranje HTML koda.

Ovaj princip je blizak **Template** paternu, gde se zajednički elementi nalaze u osnovnom šablonu, a specifičan sadržaj stranica se ubacuje unutar tog okvira.

9.4. Factory pattern (mogućnost primene)

Factory pattern se koristi kada želimo da centralizujemo logiku kreiranja objekata, umesto da svuda po kodu pišemo `new`. U ovom projektu se koncept Factory paterna može primeniti na kreiranje objekata korisnika (*User*) i predmeta (*Course*), gde jedna funkcija (ili klasa) na osnovu podataka iz baze vraća kompletno popunjene objekte.

Ilustrativan kod (nije obavezno da doslovno postoji u projektu, može da bude "predlog arhitekture" - tj. kako bi se to uradilo):

```
class UserFactory {
    public static function createFromRow(array $row): User {
        $user = new User();
        $user->id = $row['user_id'];
        $user->name = $row['name'];
        $user->email = $row['email'];
        $user->role = $row['role'];
        // ...
        return $user;
    }
}
```

Ovakav pristup bi omogućio da se sav kod za mapiranje podataka iz baze u PHP objekte nalazi na jednom mestu, što povećava preglednost i smanjuje mogućnost greške.

9.5. Logička struktura projekta i organizacija direktorijuma

Kôd aplikacije organizovan je po logičkim celinama kako bi se olakšalo snalaženje:

- **korenski direktorijum projekta** - sadrži osnovne ulazne tačke (npr. `index.php`, `login.php`, `logout.php`).
- **/admin/** - skripte i view-ovi namenjeni administratorskoj ulozi (upravljanje korisnicima, predmetima, rokovima, statistikom).
- **/professor/** - stranice za profesore (predmeti, materijali, ocene, obaveštenja).
- **/student/** - stranice za studente (moji predmeti, prijava ispita, moje prijave, ocene, materijali).
- **/includes/** - zajednički delovi interfejsa (header, footer, navbar, sidebar) i eventualno pomoćne funkcije.
- **/assets/** - statički fajlovi: CSS, JavaScript, slike, ikonice.
- **/config/** ili **/database/** - fajlovi za konekciju sa bazom, podešavanja, konstante.

Ovako organizovan projekat omogućava:

- da se lako pronađe deo koda koji treba menjati (po ulozi ili funkcionalnosti),
- da se jasnije odvoji backend logika (config, database) od frontenda (assets, includes),
- da se u budućnosti projekat lakše prebaci na neki framework (npr. Laravel) koji već podrazumeva MVC strukturu.

10. Bezbednost web aplikacije

Bezbednost web aplikacije fakulteta je posebno važna jer sistem obrađuje osetljive podatke o studentima, profesorima, ispitima i ocenama. U ovom poglavlju opisani su ključni mehanizmi zaštite koji su implementirani ili planirani u okviru aplikacije, kao i preporuke za dalja unapređenja.

10.1. Autentifikacija i zaštita lozinki

Autentifikacija korisnika se zasniva na kombinaciji **email adresa + lozinka**.

Da bi se smanjio rizik od kompromitovanja naloga:

- lozinke se **ne čuvaju u čistom tekstu**,
- prilikom registracije ili kreiranja korisnika lozinka se prolazi kroz funkciju za **hashovanje** (npr. `password_hash` sa `bcrypt` algoritmom),
- prilikom logovanja, unesena lozinka se poredi sa hash vrednošću iz baze pomoću funkcije (npr. `password_verify`).

Ovakav pristup obezbeđuje da čak i u slučaju neovlašćenog pristupa bazi podataka, napadač ne vidi direktne lozinke korisnika, već samo njihove hash vrednosti.

10.2. Kontrola pristupa i uloge (Role-Based Access Control)

Aplikacija koristi **model uloga (Role-Based Access Control - RBAC)** kako bi se ograničio pristup funkcionalnostima:

- **student** - može da vidi samo svoje podatke (sopstvene prijave, ocene, predmete, materijale),
- **profesor** - ima pristup predmetima na kojima je nosilac, listi studenata i unosu ocena za te studente,

- **administrator** - ima proširene privilegije: upravljanje korisnicima, predmetima, ispitnim rokovima i obaveštenjima.

Za svaku zaštićenu stranicu:

- pre iscrtavanja sadržaja proverava se da li je korisnik ulogovan,
- proverava se uloga korisnika (npr. samo `role = 'admin'` može da pristupi admin panelu),
- u slučaju da korisnik nema odgovarajuća prava, prikazuje se poruka o grešci ili se vrši redirekcija na početnu stranicu.

Na ovaj način sprečava se da studenti ili profesori direktnim unosom URL-a pristupe administratorskim stranicama ili funkcijama koje im ne pripadaju.

10.3. Validacija i sanitizacija ulaznih podataka

Svi podaci koje korisnik unosi kroz forme (login, prijava ispita, dodavanje korisnika, kreiranje roka itd.) prolaze kroz **validaciju** i, po potrebi, **sanitizaciju**:

- validira se da su obavezna polja popunjena (email, lozinka, naziv predmeta...),
- proveravaju se formati (email adresa, datumi, numeričke vrednosti),
- ako je unos neispravan, korisnik dobija jasnu poruku o grešci i unos se ne prosleđuje bazi.

Kod tekstualnih polja koja se kasnije prikazuju na stranici (npr. naslov i sadržaj obaveštenja) koristi se:

- escapovanje specijalnih HTML karaktera (npr. `htmlspecialchars`) kako ne bi bilo moguće uneti HTML/JS kod i izvršiti skript na strani pregledača (*XSS zaštita*).

10.4. Zaštita od SQL Injection napada

SQL injection je jedna od najčešćih vrsta napada na web aplikacije koje rade sa bazama podataka.

Da bi se smanjio rizik:

- SQL upiti se ne konkatenuiraju direktno sa neproverenim korisničkim unosom,
- koriste se **pripremljeni upiti (Prepared Statements)** uz bindovanje parametara (npr. PDO ili MySQLi prepared),
- sve vrednosti pre slanja u bazu prolaze kroz validaciju tipa (broj, datum, ograničena dužina stringa itd.).

Na ovaj način, i ako korisnik pokuša da unese zlonameran SQL (npr. ' OR 1=1 - -), on će biti tretiran kao običan tekst, a ne kao deo upita.

10.5. Zaštita od XSS (Cross-Site Scripting) napada

XSS napadi nastaju kada napadač uspe da ubaci sopstveni JavaScript kod u stranicu koja se prikazuje drugim korisnicima (npr. kroz polja za unos teksta).

U aplikaciji se XSS rizik smanjuje na sledeći način:

- prilikom prikaza podataka unetih od strane korisnika (naslovi obaveštenja, opisi, neka tekstualna polja) koristi se funkcija koja escapuje specijalne HTML karaktere,
- ograničava se maksimalna dužina tekstualnih polja,
- ne dopušta se proizvoljan HTML unos (npr. u obaveštenjima i opisima ne podržava se direktno umetanje `<script>` tagova).

Dodatna preporuka za budući rad je da se uvede centralizovana funkcija ili helper za "siguran prikaz" (npr. `e($string)`), koja uvek poziva `htmlspecialchars`.

10.6. Sesije, kolačići i timeout

Aplikacija koristi **PHP sesije** za praćenje ulogovanog korisnika:

- prilikom uspešnog logovanja u sesiju se upisuju bar `user_id` i `role`,
- na svakoj stranici koja zahteva logovanje proverava se da li sesija postoji,
- pri odjavi (Logout) sesija se uništava (`session_destroy`) i relevantne vrednosti se uklanjaju.

Kao dodatno unapređenje, moguće je uvesti:

- **vremenski ograničen session timeout** - npr. automatska odjava korisnika nakon 20-30 minuta neaktivnosti,
- flagove za kolačiće (`HttpOnly`, `Secure`) kada se aplikacija koristi preko HTTPS-a,
- zaštitu od "session fixation" napada regenerisanjem session ID-a nakon logovanja.

10.7. Ostali bezbednosni aspekti i preporuke

Pored navedenog, važni su i sledeći aspekti bezbednosti:

- **HTTPS protokol**
 - u produkcionom okruženju aplikaciju treba koristiti isključivo preko HTTPS protokola, kako bi se šifrovao saobraćaj između browsera i servera (posebno lozinke i sesije).

- **Ograničavanje upload-a fajlova**

- kod upload-a nastavnih materijala, potrebno je ograničiti dozvoljene ekstenzije (npr. samo PDF, DOCX, PPTX),
- fajlovi se čuvaju van direktno izvršivih direktorijuma ili se dodatno štite pravilnom konfiguracijom web servera.

- **Logovanje aktivnosti**

- beleženje ključnih akcija (logovanja, neuspelih logovanja, kreiranja i brisanja korisnika, unošenja ocena) u log fajl ili log tabelu radi kasnije analize.

- **Rate limiting za pokušaje logovanja**

- mogućnost da se ograniči broj neuspelih pokušaja logovanja po IP adresi ili korisničkom nalogu u određenom vremenu, kako bi se umanjio rizik od brute-force napada.

Primena ovih mera čini aplikaciju robusnijom i spremnijom za rad u realnom okruženju, posebno u slučaju da se sistem koristi na nivou cele visokoškolske ustanove.

11. Budući razvoj i unapređenja

Iako trenutna verzija web aplikacije pokriva osnovne potrebe fakultetskog informacionog sistema (upravljanje korisnicima, predmetima, ispitima, materijalima i obaveštenjima), postoji veliki prostor za dalja unapređenja. U nastavku su navedene ideje koje bi značajno poboljšale funkcionalnost, upotrebljivost i inteligenciju sistema.

11.1. Integracija sa mobilnom aplikacijom i REST API

Jedno od najlogičnijih proširenja je razvoj **mobilne aplikacije za studente i profesore** (Android/iOS).

Za to je neophodno dodati:

- **REST API** sloj u postojeću aplikaciju, koji bi omogućio bezbedan pristup podacima (prijave, ocene, materijali) iz mobilne aplikacije,
- endpoint-e za:
 - logovanje,
 - pregled predmeta,
 - prijavu ispita,
 - pregled ocena,
 - pregled obaveštenja.

Na ovaj način, studenti bi mogli da preko telefona:

- prijave ispit u roku od nekoliko sekundi,
- dobiju notifikaciju kada dobiju novu ocenu,

- direktno preuzmu materijale.

11.2. Sistemi obaveštavanja (email / SMS / push notifikacije)

Trenutno obaveštenja studenti vide tek kada se uloguju na sistem. Jedno unapređenje bi bilo uvođenje **aktivnog sistema obaveštavanja**, koji bi automatski slao poruke kada se dese važne promene, npr.:

- nova ocena je uneta,
- objavljen je novi ispitni rok,
- promenjen je termin ispita ili nastave,
- postavljen je novi nastavni materijal.

Mogući kanali:

- **email notifikacije** (najrealnije za početak - slanje automatskih email poruka na adresu korisnika),
- **SMS poruke** (uz korišćenje eksternog SMS gateway servisa),
- **push notifikacije** u mobilnoj aplikaciji (u kombinaciji sa idejom iz 11.1).

11.3. Napredna analitika i izveštavanje

Za administraciju i nastavnike korisno bi bilo da sistem nudi naprednije **statističke prikaze i izveštaje**:

- prolaznost po predmetima i ispitnim rokovima,
- prosečne ocene po generaciji i smeru,
- broj ponavljanja istog ispita po studentu,

- vizuelni grafici (bar chart, line chart, pie chart) koji bi pomogli u donošenju odluka.

U naprednijoj fazi, ove statistike se mogu proširiti i na:

- identifikaciju "rizičnih" predmeta (npr. ekstremno niska prolaznost),
- pomoć pri planiranju kapaciteta ispitnih termina,
- otkrivanje trendova uspeha studenata na različitim smerovima.

11.4. AI chatbot i pametni asistenti

Zanimljivo unapređenje bi bilo uvođenje **chatbot asistenta** integrisanog u studentski i profesorski panel:

- studentu bi chatbot mogao da odgovori na pitanja tipa:
 - "Koje ispite mogu da prijavim ovog meseca?"
 - "Koliko mi je prosečna ocena?"
 - "Kada mi je sledeći ispit iz Serverskih tehnologija?"
- profesorima bi asistent pomagao da brže pronađu informacije o studentima, prijavama i prošlim rezultatima.

Tehnički, chatbot bi koristio API-je aplikacije (ili direktan pristup bazi preko dodatnog servisa) da bi na osnovu upita korisnika generisao odgovore.

11.5. Povezivanje sa LMS sistemima i dodatne funkcionalnosti

U budućnosti je moguće proširiti sistem integracijom sa postojećim **LMS (Learning Management System)** rešenjima (npr. Moodle), tako da:

- studenti osim ocena i prijava ispita imaju i direktan pristup online testovima, zadacima i diskusionim forumima,

- profesori upravljaju kompletnim nastavnim procesom kroz jedan objedinjeni sistem.

Dodatne ideje:

- modul za evidenciju prisustva na vežbama i predavanjima,
- elektronsko izdavanje potvrda (uverenje o studiranju, potvrda o položenim ispitima),
- integracija sa finansijskim sistemom (npr. pregled uplaćenih rata školarine).

12. Zaključak

U ovom radu predstavljena je dokumentacija web aplikacije fakulteta razvijene u okviru predmeta *Serverske tehnologije*, dok je fokus predmeta *Programski alati za razvoj softvera* bio na pravilnom modelovanju, testiranju, dokumentovanju i analizi sistema.

Na početku rada opisana je **motivacija** za razvoj informacionog sistema visokoškolske ustanove i dat je kratak pregled postojećih rešenja na fakultetima i visokim školama. Istaknuto je da savremeni sistemi moraju da objedine procese upisa na predmete, prijave ispita, vođenja ocena, objave materijala i obaveštenja, uz jasnu podelu uloga na studente, profesore i administratore.

Zatim je detaljno analizirana **struktura sajta**, uz prikaz mape sajta i tri odvojena panela (admin, profesorski, studentski). Posebna pažnja posvećena je **bazi podataka**, kroz ER dijagram i opis ključnih entiteta: `users`, `courses`, `enrollments`, `exam_periods`, `exam_applications`, `grades`, `materials`, `announcements`. Na ovaj način jasno je definisan način čuvanja podataka i relacije između studenata, predmeta, ispita, ocena i materijala.

Korišćenjem **Use Case**, **sekvencijalnih** i **process flow** dijagrama, prikazane su funkcionalnosti sistema iz perspektive različitih korisnika i način na koji se zahtev "kreće" kroz aplikaciju - od unosa u browseru do upisa u bazu podataka. Ovi dijagrami olakšavaju razumevanje sistema i predstavljaju dobru osnovu za dalji razvoj i timski rad.

U poglavlju o **test planu i test case-ovima** definisani su testovi za sve tri uloge (student, profesor, administrator), uz kombinaciju pozitivnih i negativnih scenarija. Time je pokazano da aplikacija nije posmatrana samo sa aspekta implementacije, već i sa aspekta kvaliteta i pouzdanosti.

Posebno poglavlje posvećeno je **tehnologijama i resursima**: backend (PHP, MySQL), frontend (HTML, CSS, Bootstrap, JavaScript), alati za dijagrame,

razvojno okruženje, tutorijali i dokumentacija. Time je jasno naglašeno na kojim se tehnologijama sistem zasniva i koji su eksterni izvori korišćeni tokom rada.

U delu o **design pattern-ima i organizaciji koda** prikazano je kako se primenjuju osnovni principi MVC arhitekture, Singleton paterna za rad sa bazom i šablona (template) za zajedničke delove interfejsa. Ovakav pristup doprinosi boljoj strukturi projekta, lakšem održavanju i eventualnom prelasku na naprednije framework-e u budućnosti.

Poglavlje o **bezbednosti web aplikacije** pokazuje da se vodilo računa o ključnim aspektima zaštite: hashovanju lozinki, kontroli pristupa na osnovu uloga, validaciji i sanitizaciji unosa, zaštiti od SQL injection i XSS napada, kao i pravilnom rukovanju sesijama. Dodatno su date preporuke za unapređenja, poput HTTPS-a, ograničavanja upload-a fajlova i logovanja aktivnosti.

Na kraju, u poglavlju "**Budući razvoj i unapređenja**" predložene su konkretne ideje za dalji rad: uvođenje REST API-ja i mobilne aplikacije, sistemi obaveštavanja putem email/SMS/push notifikacija, napredna analitika i izveštavanje, AI chatbot asistent, integracija sa LMS sistemima i dodatni moduli (prisustvo, elektronske potvrde, finansije). Ove ideje pokazuju da projekat ima potencijal da preraste u kompletan informacioni sistem fakulteta.

Kroz izradu ove aplikacije i dokumentacije, autor(i) projekta stekao/la je praktično iskustvo u radu sa web tehnologijama, bazama podataka, modelovanjem sistema, testiranjem i dokumentovanjem softvera. Dokument koji je pred vama može da posluži kao osnova za dalje proširenje sistema, ali i kao referenca za slične projekte u akademskom i profesionalnom okruženju.

13. Literatura

1. **PHP Manual**, The PHP Group, dostupno na: *php.net*
2. **MySQL Reference Manual**, Oracle Corporation, dostupno na: *dev.mysql.com*
3. **Bootstrap Documentation**, dostupno na: *getbootstrap.com*
4. **MDN Web Docs - HTML, CSS, JavaScript**, Mozilla Foundation, dostupno na: *developer.mozilla.org*

13.1. Alati i resursi za dizajn

12. **Draw.io (diagrams.net)** - alat za izradu ER, Use Case, sekvencijalnih i flow dijagrama, *diagrams.net*.
13. **Figma** - online alat za dizajn i skiciranje interfejsa, *figma.com*.
14. **Flaticon** - baza vektorskih ikonica, korišćena za ikonice u interfejsu, *flaticon.com*.
15. **Pexels** - besplatne stock fotografije korišćene za ilustrovanje stranica, *pexels.com*.
16. **Coolors** - generator kolor paleta za izbor boja aplikacije, *coolors.co*.
17. **Google Fonts** - web fontovi (npr. Roboto, Poppins) korišćeni u aplikaciji, *fonts.google.com*.

13.2. AI alati

18. **ChatGPT (OpenAI)** - korišćen kao pomoćni alat pri razradi arhitekture, osmišljavanju test case-ova i pisanju delova dokumentacije, uz proveru i dopunu na osnovu zvaničnih i stručnih izvora.