

Opis web aplikacija za upravljanje fakultetskim ispitima i evidencijom studenata

seminarski rad

Jovan Jovanovic Its 20/24

SADRŽAJ

1. Uvod.....	5
2. Ciljevi i zadaci rada.....	7
3. Opis problema i funkcionalni zahtevi sistema.....	9
3.1. Opis problema i okruženja.....	9
3.2. Funkcionalni zahtevi sistema.....	10
3.2.1. Funkcionalni zahtevi - administrator.....	10
3.2.2. Funkcionalni zahtevi - profesor.....	12
3.2.3. Funkcionalni zahtevi - student.....	13
3.3. Nefunkcionalni zahtevi.....	16
4. Tehnologije i alati.....	18
4.1. Pregled korišćenih tehnologija.....	18
4.2. PHP kao serverski jezik.....	18
4.3. MySQL baza podataka.....	20
4.4. HTML5, CSS3 i Bootstrap 5.....	22
HTML5.....	22
Bootstrap 5.....	22
4.5. JavaScript i Bootstrap skripte.....	23
4.6. Hosting okruženje - Hostinger.....	24
4.7. Pomoćni razvojni alati.....	25
5. Dizajn sistema i arhitektura.....	28
5.1. Logička arhitektura sistema.....	28
5.2. Troslojna arhitektura (klijent - server - baza).....	31
5.3. Tok prijave (login) i kontrole pristupa.....	34
5.4. Organizacija koda i direktorijuma.....	35
5.5. Pregled glavnih tokova korišćenja.....	37
6. Model baze podataka.....	39
6.1. Tabela users - korisnici sistema.....	41
6.2. Tabela courses - predmeti.....	43
6.3. Tabela enrollments - upisi studenata na predmete.....	44
6.4. Tabela exam_periods - ispitni rokovi.....	45
6.5. Tabela exam_applications - prijave ispita.....	45
6.6. Tabela grades - ocene.....	46
6.7. Tabela announcements - obaveštenja.....	47
6.8. Tabela materials - nastavni materijali.....	47
6.9. Veze i integritet podataka.....	48
7. Implementacija aplikacije.....	50
7.1. Početna stranica i proces prijave na sistem.....	50
7.2. Administratorski modul.....	52
7.2.1. Administratorski dashboard.....	53
7.2.2. Upravljanje korisnicima (users.php i povezane stranice).....	54
7.2.3. Upravljanje predmetima (courses.php).....	56
7.2.4. Upravljanje ispitnim rokovima (exam_periods.php).....	57
7.2.5. Obaveštenja (announcements.php).....	58
7.2.6. Statistika sistema (statistics.php).....	59

7.3. Profesorski modul.....	60
7.3.1. Profesorski dashboard.....	61
7.3.2. Predmeti i studenti (courses.php, course_students.php).....	61
7.3.3. Ispiti i ocenjivanje (exams.php, exam_students.php).....	62
7.3.4. Nastavni materijali (materials.php).....	63
7.4. Studentski modul.....	64
7.4.1. Studentski dashboard (dashboard.php).....	64
7.4.2. Upis i pregled predmeta (courses.php, my_courses.php).....	65
7.4.3. Prijava i odjava ispita (exams.php).....	66
7.4.4. Pregled ocena (grades.php).....	67
7.4.5. Profil, obaveštenja i materijali.....	68
7.5. Validacija, poruke i korisničko iskustvo.....	69
8. Bezbednost i validacija podataka.....	70
8.1. Autentifikacija i autorizacija korisnika.....	70
8.1.1. Autentifikacija.....	70
8.1.2. Autorizacija i zaštita ruta.....	71
8.2. Zaštita lozinki.....	72
8.3. Zaštita od SQL injection napada.....	73
8.4. Validacija korisničkog unosa.....	74
8.4.1. Serverska validacija.....	74
8.4.2. Klijentska validacija.....	75
8.5. Bezbedno rukovanje fajlovima.....	75
8.6. Obrada grešaka i zaštita internih informacija.....	76
8.7. Ograničenja i moguća unapređenja bezbednosti.....	77
9. Testiranje i scenariji korišćenja.....	79
9.1. Pristup testiranju.....	79
9.2. Test podaci i test korisnici.....	81
9.3. Scenarijo 1 - Administrator upravlja korisnicima i predmetima.....	82
9.4. Scenarijo 2 - Profesor pregleda studente i unosi ocene.....	84
9.5. Scenarijo 3 - Student upisuje predmete i prati materijale.....	85
9.6. Scenarijo 4 - Student prijavljuje ispit, profesor ga ocenjuje, student vidi ocenu....	86
9.7. Testiranje grešaka i graničnih slučajeva.....	88
9.8. Rezultati testiranja.....	91
10. Moguća unapređenja sistema.....	92
10.1. Funkcionalna proširenja.....	92
10.1.1. Kurs-specifična obaveštenja.....	92
10.1.2. Raspored časova i konsultacija.....	92
10.1.3. Više izlazaka na ispit i istorija ocena.....	93
10.1.4. Evidencija prisustva i vežbi.....	93
10.2. Tehnička poboljšanja.....	93
10.2.1. Naprednije heširanje lozinki.....	93
10.2.2. CSRF zaštita za forme.....	94
10.2.3. API sloj i SPA klijent.....	94
10.2.4. Paginacija i napredna pretraga.....	94
10.3. Unapređenje korisničkog interfejsa.....	95
10.4. Integracije sa drugim sistemima.....	95
10.5. DevOps, backup i monitoring.....	95
11. Zaključak.....	97

12. Literatura.....	99
---------------------	----

1. Uvod

Savremeno visoko obrazovanje zasniva se na velikom broju nastavnih aktivnosti, ispita, rokova i administrativnih obaveza koje je potrebno precizno voditi i sinhronizovati. Čak i na relativno malim visokoškolskim ustanovama, broj studenata, nastavnog osoblja i predmeta vrlo brzo postaje dovoljno veliki da ručno vođenje evidencije u Excel tabelama ili na papiru postane neefikasno, sklono greškama i teško za održavanje.

Zbog toga se većina fakulteta i visokih škola oslanja na specijalizovane informacione sisteme koji omogućavaju centralizovano čuvanje podataka o studentima, predmetima, ispitnim rokovima i položenim ispitima, kao i različite administrativne funkcionalnosti namenjene studijskoj službi, profesorima i samim studentima.

U ovom seminarskom radu prikazana je izrada web aplikacije pod nazivom „**Zmaj University**“, koja predstavlja prototip jednostavnog fakultetskog informacionog sistema. Aplikacija omogućava prijavu na sistem sa tri različita tipa korisnika (administrator, profesor, student) i obezbeđuje svakoj ulozi odgovarajući skup funkcionalnosti. Administratori imaju mogućnost upravljanja korisnicima, predmetima, ispitnim rokovima, obaveštenjima i statistikom. Profesori imaju uvid u sopstvene predmete, prijave studenata na ispit i mogu da unose i menjaju ocene. Studenti se mogu upisivati na predmete, prijavljivati na ispitne rokove, pratiti svoja obaveštenja, materijale i pregledati položene ispite.

Poseban akcenat u projektu stavljen je na jasnu podelu uloga, dosledan dizajn korisničkog interfejsa i korišćenje standardnih web tehnologija koje su dostupne u tipičnom hosting okruženju, kakvo studenti često imaju na raspolaganju. Aplikacija je razvijena u programskom jeziku **PHP**, uz korišćenje **MySQL** baze podataka i frontend okvira **Bootstrap 5**, čime je obezbeđen responzivan dizajn i korektno prikazivanje na savremenim browserima.

Na slici ispod prikazana je početna stranica aplikacije sa login formom na kojoj se korisnik identifikuje e-mail adresom i lozinkom. Nakon uspešne autentifikacije, korisnik se automatski preusmerava na odgovarajući dashboard u zavisnosti od svoje uloge.

Zmaj University
Faculty Information System

E-mail
admin@zmaj.univ

Password
Enter your password

Sign in

Test accounts – click the email or password to copy:

ADMIN admin@zmaj.univ / admin123

PROFESSOR milan.petrovic@zmaj.univ / prof123

STUDENT s001@zmaj.univ / stud123

Click any credential above to copy it to clipboard.

Open project documentation
Seminar paper & system documentation · opens in a new tab

© 2025 Zmaj University

Slika 1: Početna (login) stranica aplikacije

Cilj ovog rada je da detaljno opiše proces analize zahteva, dizajna baze podataka i implementacije aplikacije, kao i da prikaže mogućnosti proširenja sistema u smeru realnog informacionog sistema fakulteta.

2. Ciljevi i zadaci rada

Osnovni cilj rada je razvoj funkcionalnog **web informacionog sistema za upravljanje ispitima i evidencijom studenata** koji obuhvata najčešće procese sa kojima se susreće studentska služba i nastavno osoblje na visokoškolskoj ustanovi. Sistem je zamišljen kao vežbovnički projekat za predmet iz oblasti serverskih tehnologija, ali je dizajniran dovoljno generički da se, uz manje prilagođavanje, može primeniti i u realnom okruženju.

Da bi se ovaj opšti cilj ostvario, definisano je nekoliko konkretnih zadataka:

1. Analiza problema i definisanje uloga u sistemu.

Potrebno je precizno definisati koji tipovi korisnika postoje (administrator, profesor, student), koje informacije su im potrebne i koje operacije su im dozvoljene. Na osnovu toga se formiraju funkcionalni zahtevi sistema.

2. Dizajn baze podataka za čuvanje ključnih informacija.

Baza treba da obuhvati podatke o korisnicima, studijskim programima, predmetima, ispitnim rokovima, prijavama za ispit, ocenama, obaveštenjima i nastavnim materijalima. Posebna pažnja posvećena je doslednoj upotrebi primarnih i stranih ključeva, kao i referencijalnom integritetu između tabela.

3. Implementacija serverske logike u PHP-u.

Neophodno je realizovati logiku prijave na sistem, rad sa sesijama, kontrolu pristupa po ulozi, kao i kontrolere (PHP skripte) za kreiranje, čitanje, izmenu i brisanje podataka (CRUD operacije) nad odgovarajućim tabelama u bazi.

4. Izrada korisničkog interfejsa za sve tri uloge.

Za administratora je potrebno obezbediti pregledne tabele sa svim korisnicima, predmetima, ispitnim rokovima, obaveštenjima i statistikom. Za profesore treba obezbediti stranice za pregled sopstvenih predmeta, listi prijavljenih studenata i formu za unos ocena. Za studente je cilj jednostavan i intuitivan interfejs za upis na predmete, prijavu na ispite, pregled materijala i pratioca sopstvenih rezultata.

5. Implementacija osnovnih bezbednosnih mehanizama.

Sistem mora da spreči pristup za neautorizovane korisnike, da ograniči funkcionalnosti u skladu sa ulogom i da minimizira rizik od tipičnih napada kao što su SQL injection i

neovlašćeno menjanje tuđih podataka (npr. unos ocena za drugi predmet ili drugog studenta).

6. Testiranje funkcionalnosti kroz realne scenarije korišćenja.

Potrebno je popuniti bazu realističnim test podacima (više profesora, veći broj studenata, više predmeta i ispitnih rokova) i proveriti da li se sistem ponaša očekivano u različitim situacijama: upis studenata na predmete, prijave i odjave ispita, unos ocena, promene lozinke, kreiranje novih rokova i sl.

7. Dokumentovanje rešenja i mogućih unapređenja.

Završni zadatak obuhvata izradu dokumentacije koja opisuje strukturu sistema, korišćene tehnologije, ključne delove implementacije i predloge za dalji razvoj - npr. automatizovano generisanje izveštaja, online plaćanje prijave ispita, integraciju sa postojećim studentskim servisom i sl.

Realizacijom navedenih zadataka dobijen je kompaktan, ali funkcionalno bogat sistem koji studentu omogućava rad na konkretnom, za praksu relevantnom primeru, a istovremeno demonstrira tipičnu upotrebu serverskih tehnologija u okruženju web aplikacija.

3. Opis problema i funkcionalni zahtevi sistema

3.1. Opis problema i okruženja

Na većini visokoškolskih ustanova u Srbiji i regionu i dalje se mogu sresti različiti nivoi digitalizacije. U nekim slučajevima evidencija studenata, ispita i ocena vodi se kroz kombinaciju:

- ručno popunjavanih formulara,
- Excel tabela,
- odvojenih desktop aplikacija,
- i različitih “internih fajlova” koji kruže između službi i nastavnika.

Ovakav pristup ima niz nedostataka:

- **Dupliranje podataka** - podaci o istom studentu često postoje u više fajlova (npr. jedna tabela za upis, druga za ispite, treća za školarinu), što lako dovodi do neusaglašenosti.
- **Rizik od grešaka** - ručni unos i prepisivanje podataka (indeks, JMBG, ocene) povećava mogućnost tipografskih grešaka i pogrešnih evidencija.
- **Težak uvid u celinu** - nastavnik često ne može na jednom mestu da vidi sve studente koji slušaju predmet, njihove prijave za ispit i konačne ocene, već mora da kombinuje više lista.
- **Ograničen pristup studentima** - student neretko mora fizički da dođe do studentske službe ili da čeka objavu spiskova na oglasnoj tabli kako bi proverio da li je prijava ispita prošla ili koja mu je konačna ocena.
- **Nedostatak standardizacije** - svaki nastavnik ili služba može koristiti sopstveni format (drugačije nazive kolona, različite fajlove), što otežava razmenu i objedinjavanje podataka.

Sa druge strane, moderne web tehnologije omogućavaju relativno jednostavnu izradu centralizovanih sistema koji su:

- dostupni sa bilo kog uređaja koji ima internet i browser,
- lako proširivi novim funkcionalnostima,
- sposobni da istovremeno opsluže više uloga (administrator, profesor, student),
- i mogu da obezbede jedinstvenu “istinu” o podacima (jedna baza, više pogleda).

Zbog toga se javlja potreba za aplikacijom koja će:

- objединiti evidenciju studenata, predmeta, prijava i ocena na jednom mestu,
- jasno definisati različite uloge i njihove nadležnosti,
- i omogućiti da studenti samostalno obavljaju najčešće radnje (upis predmeta, prijava ispita, pregled ocena) bez stalnog odlaska u studentsku službu.

U okviru ovog rada, taj problem je adresiran razvojem web aplikacije „**Zmaj University**“, koja simulira informacioni sistem jednog fakulteta i omogućava opisane procese uz minimalne tehničke zahteve na strani korisnika (dovoljan je savremeni web preglednik).

U nastavku su detaljno opisani funkcionalni zahtevi koji su postavljeni pred sistem.

3.2. Funkcionalni zahtevi sistema

Funkcionalni zahtevi opisuju **šta** sistem treba da radi, tj. koje funkcije korisnicima pruža. U ovom projektu oni su grupisani prema ulogama korisnika: administrator, profesor i student.

3.2.1. Funkcionalni zahtevi - administrator

Administrator predstavlja korisnika zaduženog za tehničko i organizaciono održavanje sistema. Njegove glavne funkcionalnosti su:

1. Upravljanje korisnicima (User management)

- kreiranje novih korisnika (admin, profesor, student),
- izmena postojećih korisnika (ime, prezime, e-mail, uloga, studijski smer, godina studija),
- brisanje korisnika koji više nisu aktivni,
- pretraga korisnika po imenu, prezimenu ili e-mail adresi,
- obezbeđivanje da e-mail adresa bude jedinstvena u sistemu.

2. Upravljanje predmetima (Course management)

- kreiranje novih predmeta sa nazivom, šifrom, brojem ESPB bodova i opisom,
- dodela odgovornog profesora predmetu,
- izmena podataka o predmetu (npr. promena nosioca predmeta),
- brisanje predmeta uz poštovanje ograničenja baze (npr. kaskadno brisanje vezanih podataka ili zabrana ako postoje ocene).

3. Upravljanje ispitnim rokovima (Exam period management)

- definisanje ispitnih rokova (npr. „Januar 2025“, „Jun 2025“) sa datumom početka i završetka,
- označavanje jednog ili više rokova kao aktivnih,
- mogućnost izmene ili deaktiviranja rokova,
- mogućnost da se u datom trenutku jasno vidi koji je rok trenutno aktuelan.

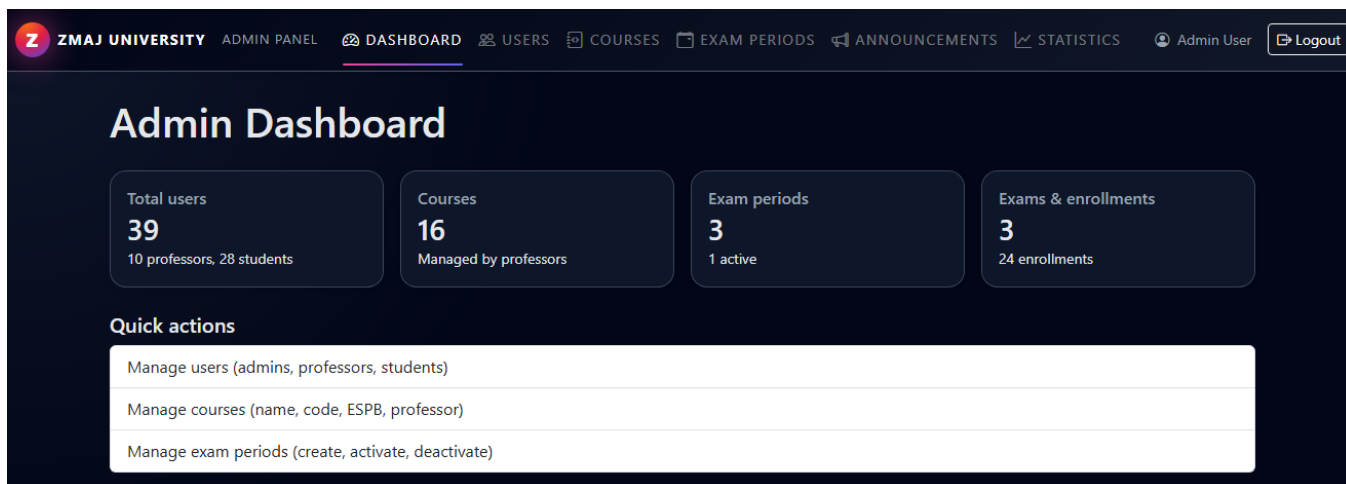
4. Upravljanje obaveštenjima (Announcements)

- kreiranje globalnih obaveštenja vidljivih svim korisnicima (npr. „Upis godine traje do...“),
- izmena i brisanje postojećih obaveštenja,
- prikaz liste objavljenih obaveštenja, sortiranih po datumu.

5. Uvid u statistiku (Statistics)

- pregled broja korisnika po ulogama (broj studenata, profesora, administratora),
- pregled broja predmeta, prijava na ispit, položenih ispita,
- potencijalno filtriranje po roku ili predmetu (u projektu je prikazan jednostavan pregled kroz tabele).

Na slici ispod može se prikazati primer administratorskog kontrolnog panela (dashboard) sa najvažnijim brojevima i linkovima ka glavnim modulima



Slika 2: Administratorski dashboard sa sažetkom sistema

3.2.2. Funkcionalni zahtevi - profesor

Profesor je korisnik koji je odgovoran za izvođenje nastave na određenim predmetima i za ocenjivanje studenata. Njegove ključne funkcionalnosti su:

1. Pregled sopstvenih predmeta

- prikaz liste predmeta na kojima je profesor nosilac,
- mogućnost pregleda upisanih studenata na svakom predmetu (preko tabele sa indeksom, imenom, prezimenom i smerom).

2. Rad sa prijavama i ocenjivanjem (Exams & grades)

- pregled aktivnih ispitnih rokova,
- izbor roka i predmeta za koji želi da vidi prijavljene studente,
- lista prijavljenih studenata sa informacijama o statusu prijave i eventualno već unetoj oceni,
- unos i izmena ocene za studenta (u opsegu 5-10),
- automatsko ažuriranje statusa prijave (npr. položio za ocenu ≥ 6 , pao za ocenu 5),

- evidentiranje datuma kada je ocena uneta ili izmenjena.

3. Upravljanje nastavnim materijalima (Materials)

- dodavanje nastavnih materijala za predmete koje profesor drži (npr. prezentacije, skripte, zadaci),
- unos naslova i opisa materijala,
- postavljanje fajla na server i čuvanje putanje u bazi,
- brisanje zastarelih ili pogrešno postavljenih materijala.

4. Pregled globalnih obaveštenja i osnovne statistike

- prikaz važnih obaveštenja koja je objavio administrator,
- uvid u osnovne statističke informacije, koliko je studenata prijavljeno na ispit, koliko ih je položilo i sl. (u okviru dostupnih funkcionalnosti).

#	Index	Name	Program	Application status	Grade	Date	Action
1	IT001	Marko Kostić	Internet Tehnologije	Passed	7	2025-11-16	7 ▾ Grade 5 6 7 8 9 10 Save

Slika 3: Pregled prijava na ispit i unos ocena u profesorskom panelu

3.2.3. Funkcionalni zahtevi - student

Student je centralna uloga u sistemu, jer se ceo proces obrazovanja i polaganja ispita vrti upravo oko njegovih aktivnosti. Za studenta su predviđene sledeće funkcionalnosti:

1. Upis na predmete (Courses & enrollments)

- pregled liste dostupnih predmeta,
- mogućnost filtriranja i pretrage predmeta (prema nazivu ili šifri),

- upis (enrollment) na željeni predmet,
- odjava sa predmeta (u okviru ograničenja koje sistem definiše - npr. ne dozvoliti odjavu nakon što je položena ocena).

2. Pregled sopstvenih predmeta (My courses)

- tabela sa svim predmetima na koje je student upisan,
- prikaz osnovnih informacija o predmetu (šifra, naziv, ESPB, profesor),
- pristup nastavnim materijalima koje je profesor objavio.

3. Prijava i odjava ispita (Exams)

- prikaz aktivnih ispitnih rokova,
- izbor roka na koji student želi da se prijavi za određeni predmet,
- prijava ispita za predmet na koji je student upisan (kreiranje zapisa u tabeli `exam_applications`),
- odjava ispita, pod uslovom da za tu prijavu još nije uneta ocena,
- jasan prikaz statusa za svaki predmet (nije prijavljen, prijavljen, položen, pao).

4. Pregled ocena i istorije polaganja (Grades)

- lista položenih i nepoloženih ispita,
- prikaz konačne ocene, datuma polaganja i predmeta,
- uvid u to koliko je ESPB bodova ostvareno.

5. Pregled obaveštenja i materijala

- pregled globalnih obaveštenja koja je objavio administrator,
- detaljan prikaz materijala po predmetima na koje je student upisan.

6. Upravljanje sopstvenim profilom (Profile)

- izmena kontakt podataka (adresa, telefon i dr.),
- promena lozinke, uz proveru postojeće lozinke,

- dodavanje ili ažuriranje profilne slike, koja se čuva kao fajl na serveru a referencira u bazi.

Na slici ispod može se prikazati primer studentskog dashboard-a, sa osnovnim statistikama (broj upisanih predmeta, broj prijavljenih ispita, broj položenih ispita) i listom obaveštenja.

Code	Name	ESPB	Professor	Status	Action
ALG1	Algorithms and Data Structures	6	Ana Jovanović	Enrolled	Unenroll
CA1	Computer Architecture	6	Marko Ristić	Not enrolled	Enroll
DB1	Databases	6	Milan Petrović	Enrolled	Unenroll
DB2	Databases 2	6	Ivana Savić	Not enrolled	Enroll
ENG1	English for IT	4	Tamara Milošević	Not enrolled	Enroll
IOT1	Internet of Things	6	Dejan Blagojević	Not enrolled	Enroll
ML1	Machine Learning Basics	6	Nataša Bogdanović	Not enrolled	Enroll
MOB1	Mobile Development	6	Tamara Milošević	Not enrolled	Enroll
NET1	Computer Networks	6	Nikola Ilić	Enrolled	Unenroll
OS1	Operating Systems	6	Ana Jovanović	Enrolled	Unenroll
PM1	Project Management in IT	6	Vladimir Kostić	Not enrolled	Enroll
SE1	Software Engineering	6	Milan Petrović	Not enrolled	Enroll
SEC1	Computer Security	6	Goran Janković	Not enrolled	Enroll
TEST1	Software Testing	6	Ivana Savić	Not enrolled	Enroll
WP1	Web Programming 1	6	Milan Petrović	Not enrolled	Enroll

Slika 4: Studentski dashboard sa pregledom kursa i ispita

3.3. Nefunkcionalni zahtevi

Pored funkcionalnih zahteva, sistem mora da ispuni i određene **nefunkcionalne zahteve**, koji se odnose na način rada aplikacije, performanse, bezbednost i održavanje.

Najvažniji nefunkcionalni zahtevi su:

1. Dostupnost i jednostavnost korišćenja

- Aplikacija treba da bude dostupna putem standardnog web browser-a bez dodatne instalacije,
- Interfejs treba da bude dovoljno intuitivan da novi korisnik, uz minimalno objašnjenje, može da pronađe osnovne funkcije (prijava ispita, unos ocena, pregled predmeta).

2. Performanse

- Sistem treba da odgovori u razumnom vremenu i kada radi sa većim brojem korisnika i podataka (stotine studenata, desetine predmeta i ispita),
- Upiti ka bazi treba da koriste indekse na ključnim kolonama (primarni i strani ključevi), kako bi se izbegla nepotrebna usporavanja.

3. Bezbednost

- Neophodno je koristiti sesije za autentifikaciju i proveru uloge korisnika,
- Pristup administrativnim i profesorskim funkcijama mora biti onemogućen studentima i anonimnim korisnicima,
- Svi upiti prema bazi realizuju se preko pripremljenih izraza (prepared statements) kako bi se minimizirao rizik od SQL injection napada,
- Lozinke se u bazi čuvaju u hešovanom obliku (npr. SHA-256), a ne u čistom tekstu.

4. Modularnost i mogućnost proširenja

- Kôd je organizovan tako da se lako mogu dodavati nove funkcionalnosti (npr. dodatni izveštaji, nove uloge, API endpoint-i),

- Struktura baze podataka je dovoljno fleksibilna da se, bez radikalnih promena, mogu dodati novi entiteti (npr. finansijska evidencija, prijava laboratorijskih vežbi i sl.).

5. Kompatibilnost sa hosting okruženjem

- Aplikacija je prilagođena radu na standardnom deljenom hostingu (u ovom slučaju Hostinger),
- Koriste se standardne verzije PHP-a i MySQL-a, bez oslanjanja na specijalizovane ekstenzije koje možda nisu dostupne na svakom serveru.

Ispunjavanje ovih nefunkcionalnih zahteva omogućava da sistem ne bude samo demonstracija programerskih vežbi, već realan primer kako se serverske tehnologije mogu iskoristiti za izradu stabilne i upotrebljive web aplikacije.

4. Tehnologije i alati

U realizaciji projekta „Zmaj University“ korišćen je skup pažljivo odabranih tehnologija i alata, sa ciljem da se postigne dobra ravnoteža između jednostavnosti, dostupnosti i realističnosti u odnosu na tipično produkciono okruženje. Osnovna ideja bila je da se sistem izgradi na platformi koja je studentima lako dostupna (shared hosting, standardan PHP i MySQL), ali da istovremeno poštuje dobre prakse serverskog programiranja i rada sa bazama podataka.

4.1. Pregled korišćenih tehnologija

Na najvišem nivou, sistem se sastoji od tri glavne tehnološke celine:

- **Backend (serverska strana)**
 - programski jezik: **PHP**,
 - baza podataka: **MySQL**.
- **Frontend (klijentska strana)**
 - struktura: **HTML5**,
 - stilovi: **CSS3** + CSS framework **Bootstrap 5**,
 - osnovne interakcije: JavaScript (uglavnom u okviru Bootstrap komponenti).
- **Hosting i administracija**
 - web server i baza hostovani su na platformi **Hostinger**,
 - za upravljanje bazom koristi se **phpMyAdmin**,
 - za rad sa fajlovima koristi se Hostinger **File Manager** (i po potrebi FTP).

Pored ovih tehnologija korišćeni su i razvojni alati na strani klijenta (IDE/urednik koda, pregledač sa developerskim alatima), o čemu će biti reči u nastavku.

4.2. PHP kao serverski jezik

Za implementaciju serverske logike izabran je **PHP**, jedan od najraširenijih jezika za razvoj dinamičkih web aplikacija. Razlozi za izbor PHP-a su sledeći:

- **Široka podrška na hosting servisima** - skoro svaki shared hosting, uključujući Hostinger, nudi PHP i MySQL kao standardnu kombinaciju.
- **Jednostavnost pokretanja** - nije potrebno posebno podešavanje aplikacionog servera; .php fajlovi se izvršavaju direktno na web serveru.
- **Dobra integracija sa MySQL-om** - postoji ugrađena podrška kroz PDO (PHP Data Objects), koja omogućava korišćenje pripremljenih upita i fleksibilan rad sa bazom.

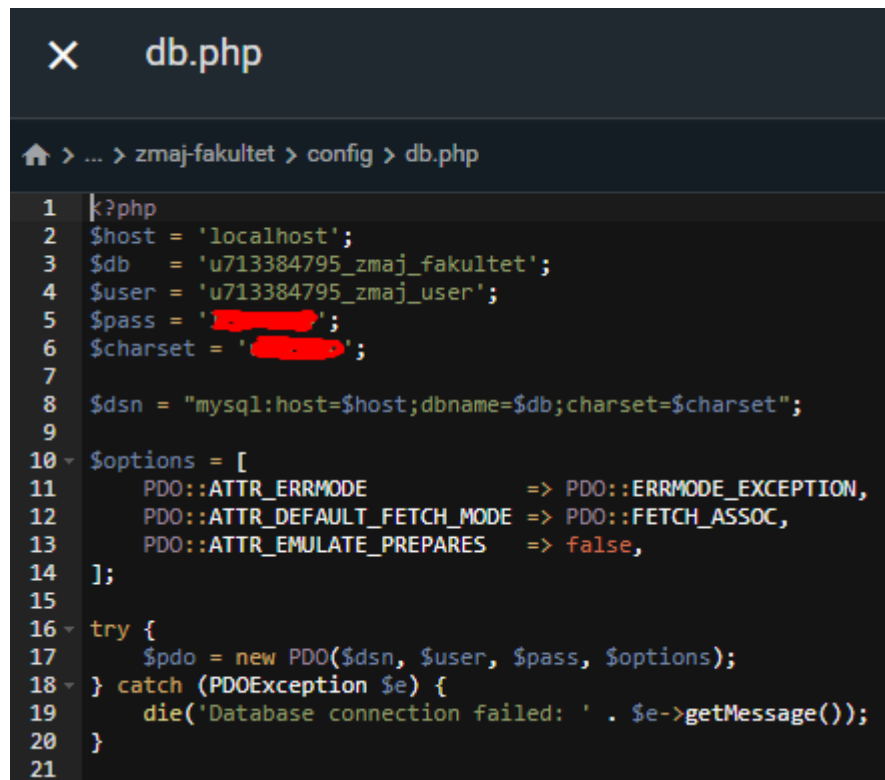
U okviru projekta PHP je korišćen za sledeće zadatke:

- obrada login forme,
- rad sa sesijama i čuvanje informacija o prijavljenom korisniku,
- proveru uloge (admin/profesor/student) i preusmeravanje na odgovarajući dashboard,
- realizaciju CRUD operacija nad tabelama (users, courses, exam_periods, exam_applications, grades, announcements, materials, enrollments),
- primenu osnovnih bezbednosnih mehanizama (heširanje lozinki, pripremljeni SQL upiti, proveru parametara u URL-u).

U projektu se koristi **PDO** kao način komunikacije sa bazom podataka. To omogućava:

- lak prelazak na drugi tip baze (teoretski),
- jednostavno i sigurno vezivanje parametara (bind param),
- jasnije razdvajanje logike upita i vrednosti koje se unose.

Konfiguracija povezivanja sa bazom izdvojena je u poseban fajl `config/db.php`, čime se olakšava održavanje i eventualno menjanje parametara (host, ime baze, korisnik, lozinka) bez prepravke svih skripti.

A screenshot of a code editor window titled 'db.php'. The editor shows a file path 'zmaj-fakultet > config > db.php'. The code is a PHP script for establishing a database connection using PDO. It defines variables for host, database name, user, password, and charset. It then constructs a DSN and an array of PDO options. Finally, it attempts to create a PDO instance and catches any exceptions to display an error message.

```
1 <?php
2 $host = 'localhost';
3 $db = 'u713384795_zmaj_fakultet';
4 $user = 'u713384795_zmaj_user';
5 $pass = ' ';
6 $charset = ' ';
7
8 $dsn = "mysql:host=$host;dbname=$db;charset=$charset";
9
10 $options = [
11     PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
12     PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
13     PDO::ATTR_EMULATE_PREPARES => false,
14 ];
15
16 try {
17     $pdo = new PDO($dsn, $user, $pass, $options);
18 } catch (PDOException $e) {
19     die('Database connection failed: ' . $e->getMessage());
20 }
21
```

Slika 5: Primer PHP koda za konekciju na bazu i korišćenje PDO-a

4.3. MySQL baza podataka

Za čuvanje svih podataka koristi se **MySQL** baza podataka, koja je de facto standard na velikom broju hosting platformi. MySQL je relacionala baza podataka koja obezbeđuje:

- rad sa tabelama povezanih preko primarnih i stranih ključeva,
- referencijalni integritet,
- podršku za transakcije (u InnoDB storage engine-u),
- adekvatne performanse za tipične akademske projekte i male do srednje sisteme.

U sistemu „**Zmaj University**“ definišeno je više međusobno povezanih tabela (detaljno u sledećem poglavlju), kao što su:

- `users` - čuva podatke o svim korisnicima sistema,
- `courses` - opis predmeta,
- `enrollments` - upisi studenata na predmete,
- `exam_periods` - ispitni rokovi,

- exam_applications - prijave ispita,
- grades - ostvarene ocene,
- announcements - obaveštenja,
- materials - nastavni materijali.

U MySQL-u je korišćen **InnoDB** storage engine kako bi se obezbedila podrška za strane ključeve i kaskadno ažuriranje/brisanje (npr. ako se predmet obriše, vezani materijali se brišu automatski, ako je tako definisano). Takođe su postavljeni odgovarajući indeksi nad primarnim i stranim ključevima, što povećava brzinu izvršavanja upita koji spajaju više tabela (JOIN).

Upravljanje strukturom baze podataka i podacima vršeno je preko alata **phpMyAdmin**, koji omogućava:

- kreiranje i izmenu tabela kroz grafički interfejs ili SQL upite,
- izvršavanje kompleksnih SQL skripti (npr. kreiranje više test korisnika odjednom),
- pregled i filtriranje postojeće evidencije,
- izvoženje i uvoz baze (backup/restore).

Table	Action	Rows	Type	Collation	Size	Overhead
announcements	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
courses	Browse Structure Search Insert Empty Drop	16	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
enrollments	Browse Structure Search Insert Empty Drop	24	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
exam_applications	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_unicode_ci	64.0 KiB	-
exam_periods	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
grades	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	64.0 KiB	-
materials	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_uca1400_ai_ci	16.0 KiB	-
users	Browse Structure Search Insert Empty Drop	39	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
8 tables	Sum	90	InnoDB	utf8mb4_unicode_ci	336.0 KiB	0 B

Slika 6: Prikaz strukture baze u phpMyAdmin-u sa tabelama sistema

4.4. HTML5, CSS3 i Bootstrap 5

Za prikaz korisničkog interfejsa korišćene su standardne web tehnologije:

- **HTML5** - za strukturu stranice, formi i tabela,
- **CSS3** - za osnovno stilizovanje,
- **Bootstrap 5** - kao frontend framework za responzivni dizajn i gotove komponente.

HTML5

HTML5 je korišćen za izradu:

- formi za login, registraciju/upis, unos i izmenu podataka,
- tabela za prikaz korisnika, predmeta, ispita i materijala,
- navigacionih traka (navbar) za admin/professor/student panele,
- kartica (cards) na dashboard stranicama sa osnovnim statistikama.

Struktura je organizovana tako da svaka uloga ima svoj poddirektorijum (/admin, /professor, /student), što olakšava odvajanje prikaza i kontrolu pristupa.

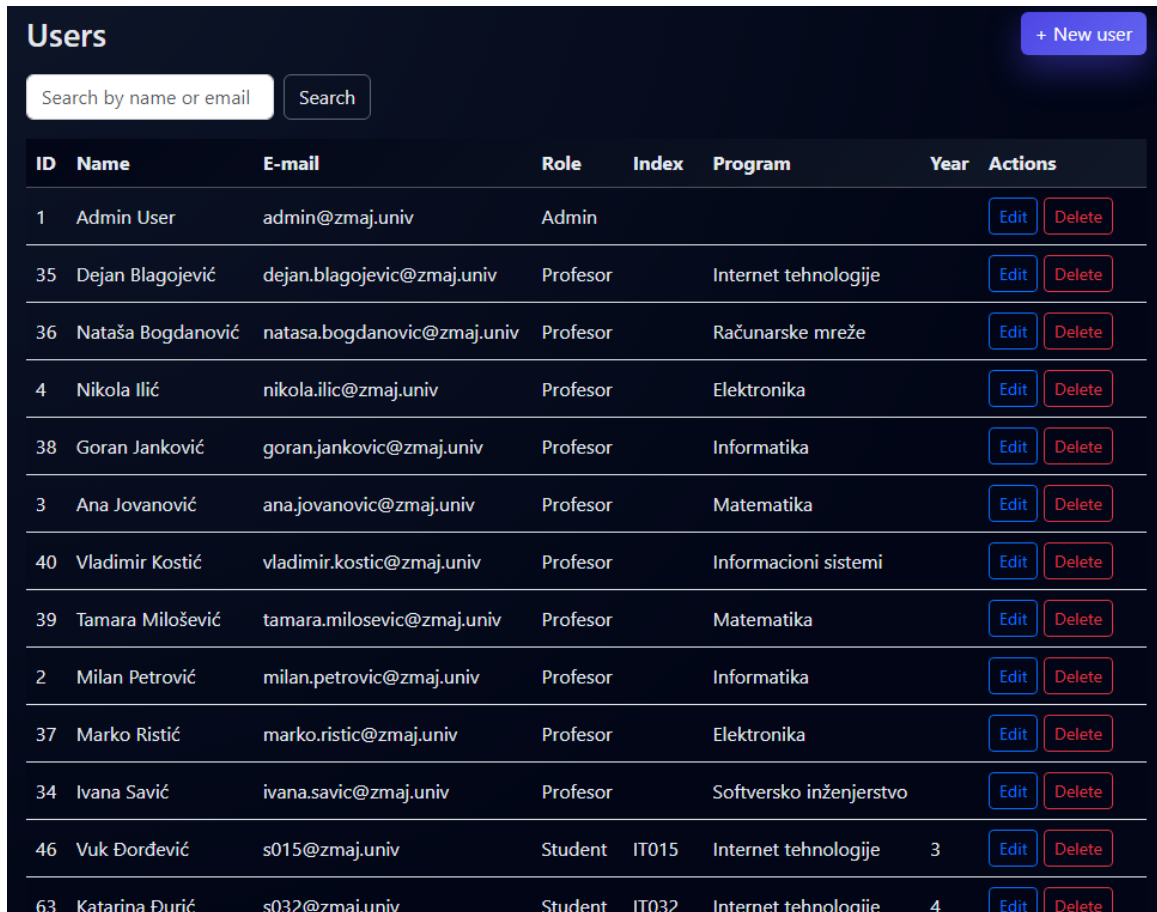
Bootstrap 5

Bootstrap 5 je korišćen iz CDN-a, što znači da nije bilo potrebe da se dodatno preuzimaju fajlovi na server. Korišćene su različite komponente:

- **Navbar** - za gornju navigacionu traku, prilagođenu svakoj ulozi (boja, naslov, linkovi),
- **Cards** - za prikaz sažetih informacija na dashboard stranama (broj studenata, predmeta, ispita itd.),
- **Tables** - za pregledne liste korisnika, predmeta, prijava, ocena,
- **Forms** - za unos podataka (Bootstrap klase za form-control, form-label, grid sistem),
- **Badges i alerts** - za statusne poruke (npr. prijavljen/položio/pao, uspešno snimljeno, greška).

Pored osnovne Bootstrap teme, napravljen je i sopstveni CSS fajl (assets/css/custom.css) koji dodaje:

- tamnu temu (dark mode) prilagođenu celom sistemu,
- prilagođene boje za brend „Zmaj University“ (indigo, pink, orange akcenti),
- vizuelno naglašavanje aktivne stavke u meniju,
- moderniji izgled kartica i tabela (gradient pozadine, blagi glow efekti).



ID	Name	E-mail	Role	Index	Program	Year	Actions
1	Admin User	admin@zmaj.univ	Admin				Edit Delete
35	Dejan Blagojević	dejan.blagojevic@zmaj.univ	Profesor		Internet tehnologije		Edit Delete
36	Nataša Bogdanović	natasa.bogdanovic@zmaj.univ	Profesor		Računarske mreže		Edit Delete
4	Nikola Ilić	nikola.ilic@zmaj.univ	Profesor		Elektronika		Edit Delete
38	Goran Janković	goran.jankovic@zmaj.univ	Profesor		Informatika		Edit Delete
3	Ana Jovanović	ana.jovanovic@zmaj.univ	Profesor		Matematika		Edit Delete
40	Vladimir Kostić	vladimir.kostic@zmaj.univ	Profesor		Informacioni sistemi		Edit Delete
39	Tamara Milošević	tamara.milosevic@zmaj.univ	Profesor		Matematika		Edit Delete
2	Milan Petrović	milan.petrovic@zmaj.univ	Profesor		Informatika		Edit Delete
37	Marko Ristić	marko.ristic@zmaj.univ	Profesor		Elektronika		Edit Delete
34	Ivana Savić	ivana.savic@zmaj.univ	Profesor		Softversko inženjerstvo		Edit Delete
46	Vuk Đorđević	s015@zmaj.univ	Student	IT015	Internet tehnologije	3	Edit Delete
63	Katarina Đurić	s032@zmaj.univ	Student	IT032	Internet tehnologije	4	Edit Delete

Slika 7: Primer admin tabele sa Bootstrap stilizacijom i tamnom temom

4.5. JavaScript i Bootstrap skripte

U ovom projektu nije bilo potrebe za složenim JavaScript logikom na klijentskoj strani. Ipak, korišćene su osnovne JS funkcionalnosti koje obezbeđuje sam Bootstrap:

- otvaranje i zatvaranje navigacije na manjim ekranima (collapse),
- modalni dijalozi (gde je to bilo potrebno),
- potvrda (confirm) pre brisanja određenih entiteta (npr. korisnika, materijala), u kombinaciji sa `onclick="return confirm('...')"` u HTML-u.

Bootstrap je uključen preko CDN skripte:

```
<script  
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.  
min.js"></script>
```

Ovakav pristup smanjuje potrebu za ručnim održavanjem biblioteka na serveru i omogućava da pregledač kešira resurse, što poboljšava performanse.

Ukoliko bi se sistem dalje razvijao (npr. uvođenje “live” pretrage, AJAX poziva, dinamičkih grafika za statistiku), tada bi bilo prirodno uključiti dodatne JS biblioteke (poput jQuery ili nekog modernijeg framework-a kao što je Vue ili React). U okviru ovog seminarskog rada, fokus je zadržan na serverskoj logici i klasičnom web obrascu “request-response”.

4.6. Hosting okruženje - Hostinger

Aplikacija je postavljena na komercijalni hosting servis **Hostinger**, što dodatno približava projekat realnim uslovima rada.

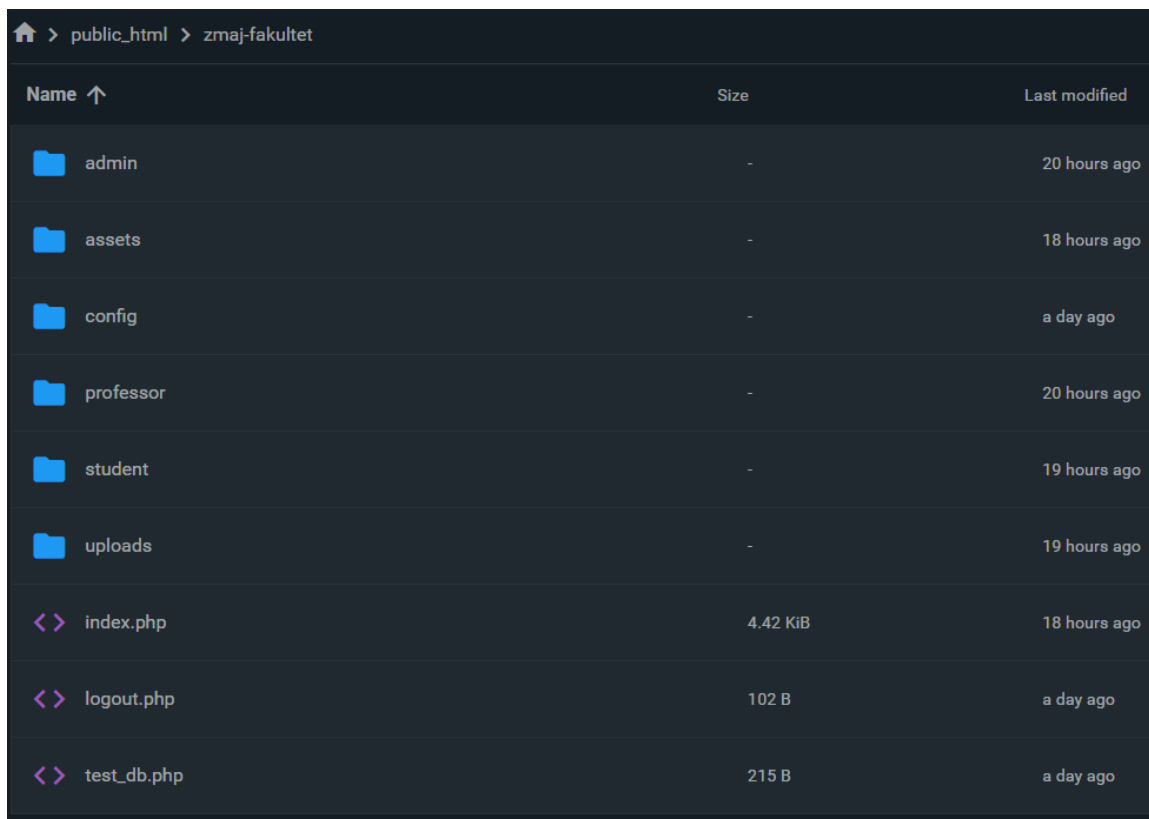
Hostinger obezbeđuje:

- podršku za **PHP** i **MySQL**,
- web interfejs za kreiranje baza, korisnika i lozinki,
- **File Manager** za rad sa fajlovima direktno u browser-u,
- mogućnost korišćenja poddomena (npr. `f.jovanhq.tech`) za projekat.

Postupak postavljanja aplikacije na Hostinger obuhvatio je sledeće korake:

1. Kreiranje nove MySQL baze podataka i korisnika putem Hostinger kontrolnog panela.
2. Upis podataka o hostu, imenu baze, korisniku i lozinki u `config/db.php` fajl na serveru.
3. Kreiranje poddirektorijuma `zmaj-fakultet` unutar `public_html`, u koji je smešten ceo projekat.
4. Prenos PHP fajlova, CSS fajla, kao i ostalih resursa putem Hostinger File Manager-a.
5. Kreiranje strukture baze podataka i inicijalnih zapisa (admin nalog, nekoliko profesora i studenata, predmeti, ispitni rokovi i sl.) preko phpMyAdmin-a.

6. Testiranje svih funkcionalnosti direktno na URL adresi projekta, kao da je reč o produkciomom sistemu.



The screenshot shows a File Manager interface with a dark theme. The breadcrumb path at the top is 'public_html > zmaj-fakultet'. Below this is a table listing files and folders. The table has three columns: 'Name' with an upward arrow, 'Size', and 'Last modified'. There are six folders (admin, assets, config, professor, student, uploads) and three PHP files (index.php, logout.php, test_db.php). Each folder is represented by a blue folder icon, and each file by a purple file icon. The 'Last modified' column shows relative times like '20 hours ago' or 'a day ago'.

Name ↑	Size	Last modified
admin	-	20 hours ago
assets	-	18 hours ago
config	-	a day ago
professor	-	20 hours ago
student	-	19 hours ago
uploads	-	19 hours ago
index.php	4.42 KiB	18 hours ago
logout.php	102 B	a day ago
test_db.php	215 B	a day ago

Slika 8: File Manager-a na Hostingeru sa strukturom projekta zmaj - fakultet

4.7. Pomoćni razvojni alati

Tokom razvoja aplikacije korišćeno je nekoliko pomoćnih alata koji nisu deo same aplikacije, ali značajno olakšavaju rad:

1. Urednik koda (IDE / editor)

Na strani klijenta korišćen je moderan editor koda (npr. Visual Studio Code), koji pruža:

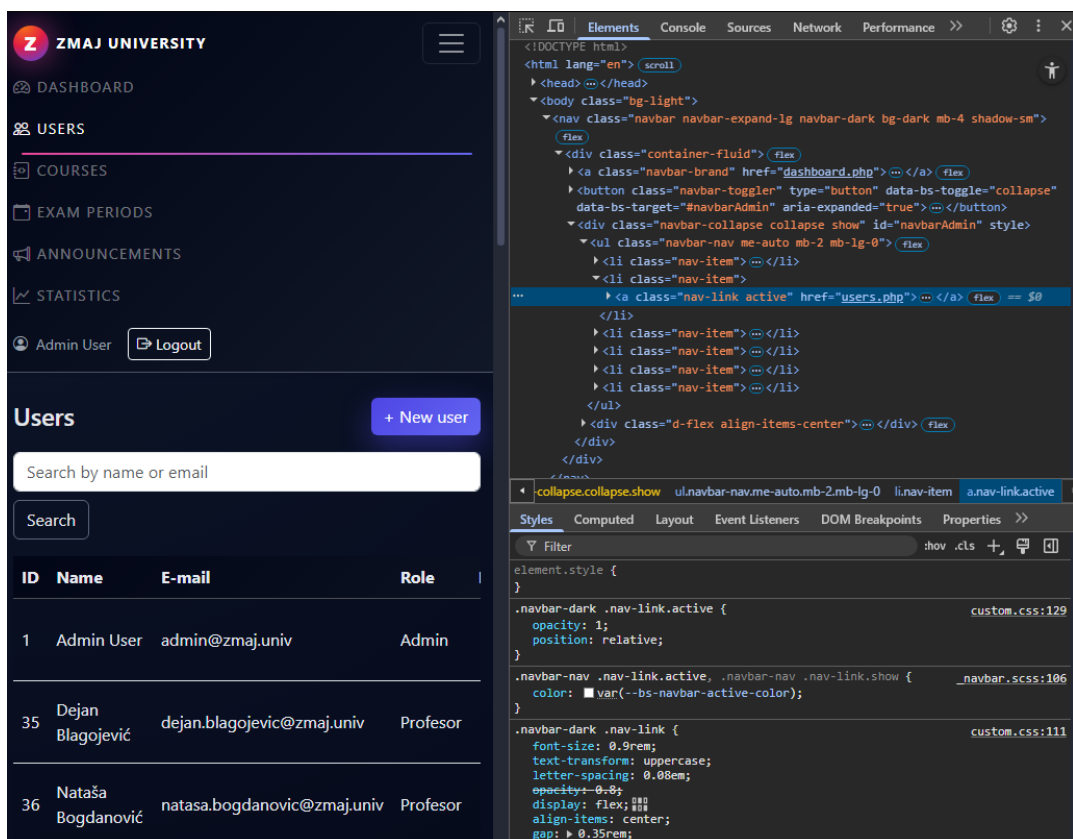
- isticanje sintakse za PHP, HTML, CSS i JavaScript,
- automatsko uvlačenje koda (indentation),
- brzu navigaciju po fajlovima i pretragu po projektu.

2. Web browser i developerski alati

Moderni pregledači (Chrome, Edge, Opera itd.) nude developerske alate koji omogućavaju:

- inspekciju HTML strukture i CSS stilova,
- pregled mrežnih zahteva (Network tab),
- brzu izmenu CSS pravila “uživo” radi testiranja dizajna,
- uvid u JavaScript greške na konzoli.

Tokom rada na dizajnu tamne teme upravo su developerski alati korišćeni da se identifikuju Bootstrap varijable koje utiču na boje teksta i pozadine (npr. `--bs-table-bg`, `--bs-secondary-color`), koje su zatim pravilno prepisane u `custom.css`.



Slika 9: Primer korišćenja developerskih alata u browseru za podešavanje stilova

3. **phpMyAdmin**

Već pomenuti phpMyAdmin korišćen je ne samo za kreiranje tabela i unosa početnih podataka, već i za:

- debugovanje SQL upita,
- ručni pregled zapisa kada treba proveriti da li je insert/updejt urađen kako treba,
- izvoženje `.sql` dump-a baze kao bekap pre većih izmena.

5. Dizajn sistema i arhitektura

Dizajn sistema predstavlja ključnu fazu između analize zahteva i same implementacije. U ovoj fazi donose se odluke o tome kako će se sistem logički podeliti na module, kako će izgledati komunikacija između klijenta (browser-a), web servera i baze podataka, kao i kako će se organizovati uloge korisnika i njihove mogućnosti.

U projektu „Zmaj University“ korišćen je klasičan pristup **troslojne arhitekture** (three-tier):

1. **Prezentacioni sloj** - web preglednik (HTML, CSS, Bootstrap, osnovni JS),
2. **Aplikacioni sloj** - PHP skripte koje obrađuju zahteve,
3. **Sloj podataka** - MySQL baza podataka.

Pored toga, sistem je logički podeljen na tri funkcionalna modula u skladu sa ulogama: **administratorski**, **profesorski** i **studentski** modul.

5.1. Logička arhitektura sistema

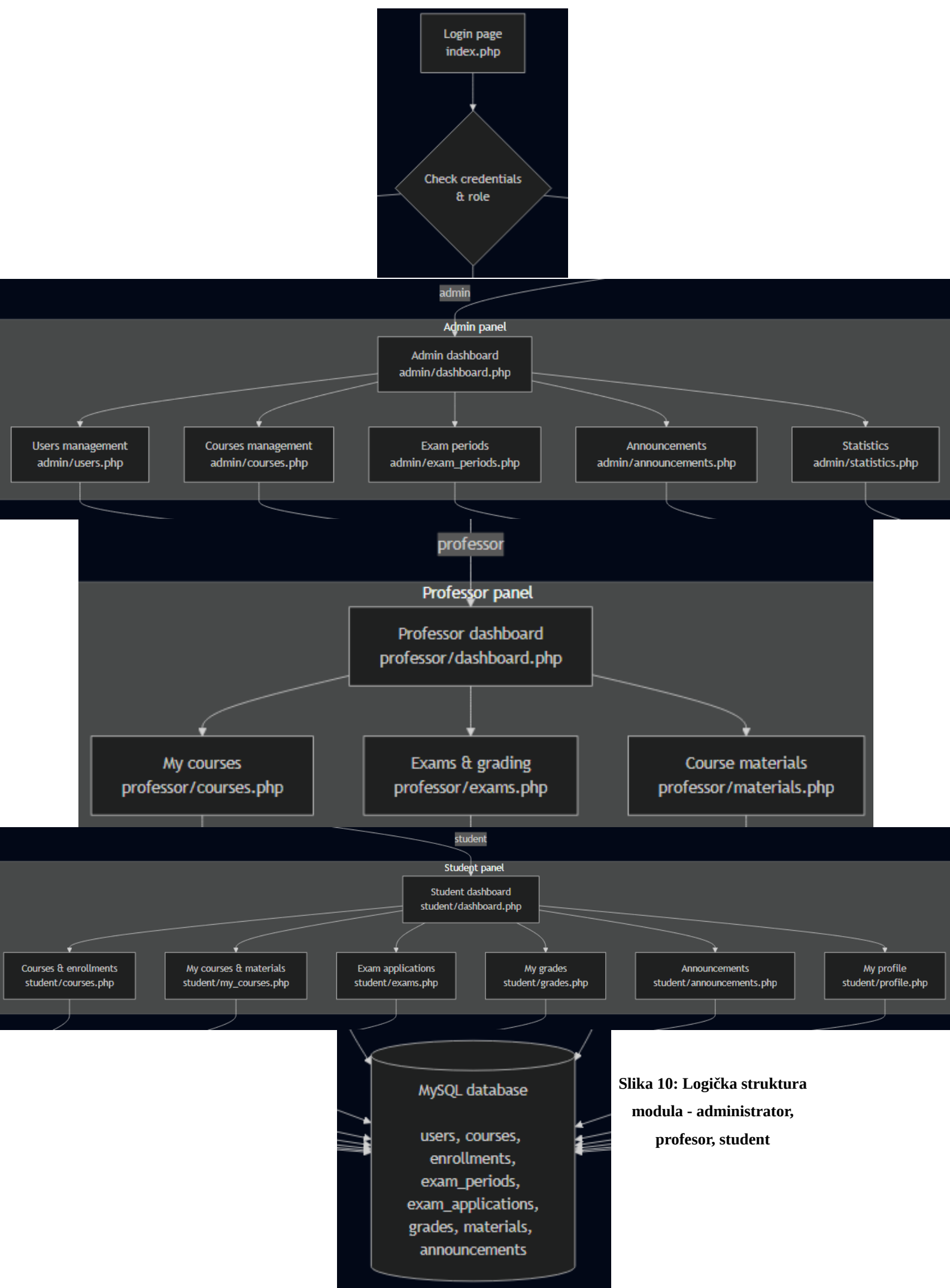
Na logičkom nivou, sistem se može posmatrati kao skup modula organizovanih oko tri tipa korisnika:

- **Administrator**
- **Profesor**
- **Student**

Svaki modul koristi zajedničku bazu podataka, ali ima različit skup ekrana i operacija. Logička arhitektura prikazuje:

- koje stranice se nalaze u okviru kog modula,
- kako korisnik prelazi sa login stranice na odgovarajući dashboard,
- koje funkcionalnosti su vezane za koje uloge.

Na dijagramu ispod (generisanom u Mermaid-u, a u seminarskom prikazanom kao slika), prikazana je logička podela sistema na module i osnovne funkcionalnosti.



Slika 10: Logička struktura modula - administrator, profesor, student

Na ovom dijagramu:

- centralnu ulogu ima **Login modul**, koji na osnovu korisničkih kredencijala (e-mail i lozinka) određuje ulogu korisnika i preusmerava ga na odgovarajući **dashboard**,
- **Administratorski modul** obuhvata:
 - upravljanje korisnicima (Users CRUD),
 - upravljanje predmetima (Courses CRUD),
 - upravljanje ispitnim rokovima (Exam periods CRUD),
 - obaveštenja (Announcements),
 - statistiku sistema (Statistics),
- **Profesorski modul** obuhvata:
 - pregled sopstvenih predmeta,
 - pregled upisanih studenata po predmetima,
 - pregled prijava na ispit po ispitnim rokovima,
 - unos i izmenu ocena,
 - rad sa nastavnim materijalima,
- **Studentski modul** obuhvata:
 - upis na predmete i pregled sopstvenih predmeta,
 - prijavu i odjavu ispita,
 - pregled ocena,
 - pregled obaveštenja i materijala,
 - osnovne funkcije profila (lični podaci, lozinka, slika).

Ovakva podela omogućava da se funkcionalnosti po ulozi jasno organizuju i da se u kodu lako kontroliše pristup pojedinim delovima sistema (npr. kroz `session` promenljive i proveru uloge na početku svake PHP skripte).

5.2. Troslojna arhitektura (klijent - server - baza)

Sa tehničkog stanovišta, aplikacija prati klasičan model web sistema sa troslojnom arhitekturom:

1. Klijentski sloj (Client / Presentation layer)

- predstavlja web preglednik (browser) studenta, profesora ili administratora,
- prikazuje HTML stranice stilizovane CSS-om i Bootstrap-om,
- šalje HTTP GET i POST zahteve ka serveru (npr. slanje forme za login, slanje forme za unos ocene).

2. Aplikacioni sloj (Application layer)

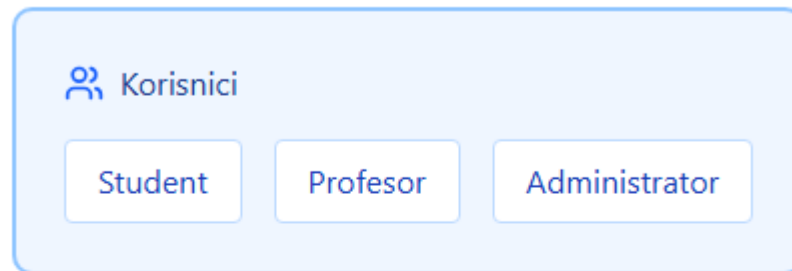
- čine ga PHP skripte koje se izvršavaju na web serveru,
- obrađuju korisnički unos, komuniciraju sa bazom podataka preko PDO-a,
- odlučuju koju HTML stranicu ili rezultat treba vratiti klijentu,
- sadrže poslovnu logiku (validacija podataka, provera uloge, definisanje statusa prijava, računanje statistike).

3. Sloj podataka (Data layer)

- realizovan je pomoću MySQL baze podataka,
- čuva sve trajne podatke (korisnici, predmeti, prijave, ocene, materijali itd.),
- obezbeđuje konzistentnost i integritet podataka kroz ograničenja (PRIMARY KEY, FOREIGN KEY).

Na sledećoj slici prikazan je dijagram arhitekture sistema, koji vizuelno opisuje tok zahteva i podataka između ova tri sloja.

1 Klijentski sloj (browser)



HTTP GET/POST zahtevi
(login, formulari, akcije)



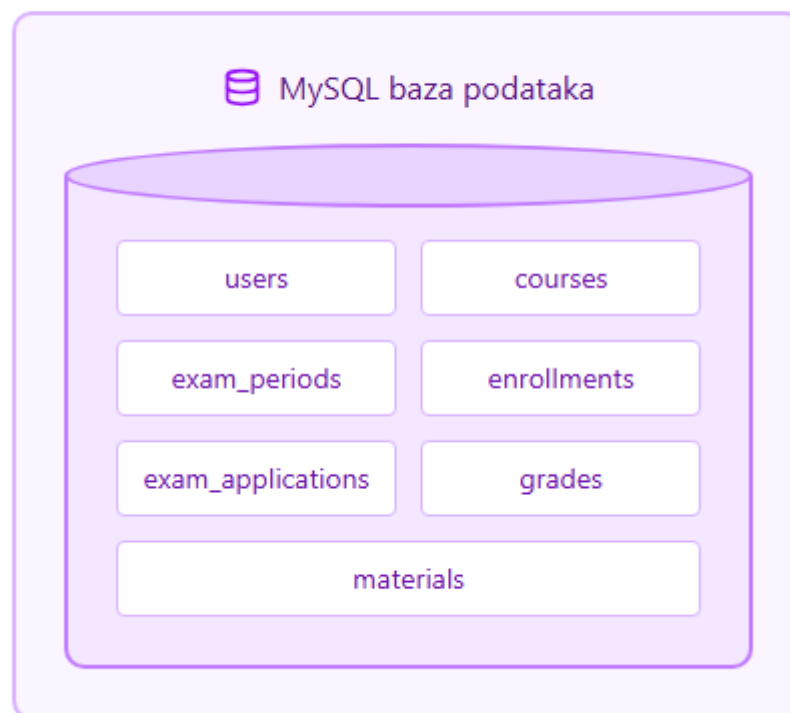
2 Aplikacioni sloj (PHP aplikacija)



PDO upiti
(SELECT/INSERT/UPDATE/DELETE)



3 Sloj podataka (MySQL baza)



Slika 11: Arhitektura sistema - browser, PHP aplikacija, MySQL baza

Na dijagramu se može uočiti:

- da svi tipovi korisnika koriste isti **URL** (domen i poddomen),
- da login formu obrađuje centralni PHP skript (`index.php` + login logika),
- da se, nakon uspešnog prijavljivanja, rade redirekcije ka odgovarajućim poddirektorijumima:
 - `/admin/` za administratore,
 - `/professor/` za profesore,
 - `/student/` za studente,
- da sve PHP skripte koriste zajednički fajl za konekciju na bazu (`config/db.php`), čime se postiže jedinstveno mesto za podešavanje parametara konekcije,
- da je baza podataka jedinstvena za ceo sistem, ali da se podaci filtriraju prema ulozi i kontekstu (npr. profesor ima uvid samo u svoje predmete, student samo u sopstvene prijave i ocene).

5.3. Tok prijave (login) i kontrole pristupa

Jedan od ključnih aspekata arhitekture sistema je kontrola pristupa. Osnovna ideja je da se svaka stranica na serverskoj strani zaštiti proverom sesije i uloge korisnika.

Tok prijave korisnika može se opisati sledećim koracima:

1. Korisnik otvara početnu stranicu aplikacije (`index.php`), na kojoj se nalazi **login forma** sa poljima:
 - E-mail adresa
 - Lozinka
2. Nakon popunjavanja forme i slanja (HTTP POST zahtev), PHP skript:
 - preuzima vrednosti iz `$_POST`,
 - pronade korisnika u tabeli `users` na osnovu e-mail adrese,
 - proveriti da li se heš lozinke poklapa sa hešom u bazi (`SHA2 / hash('sha256', ...)`),
 - ukoliko je kombinacija validna, upisuje u `$_SESSION`:
 - `user_id`,
 - `uloga` (`admin/profesor/student`),
 - eventualno `ime` i `prezime` radi prikaza u navigaciji.
3. Na osnovu uloge (`uloga`) vrši se redirekcija:
 - `admin - /admin/dashboard.php`,
 - `profesor - /professor/dashboard.php`,
 - `student - /student/dashboard.php`.
4. Na svakoj stranici koja pripada određenom modulu, na samom početku PHP fajla nalazi se provera, na primer:
`session_start();`

```

if (!isset($_SESSION['user_id']) || ($_SESSION['uloga'] ?? '') !==
'admin') {
    header('Location: ../index.php');
    exit;
}

```

Ovim se obezbeđuje da:

- neprijavljeni korisnici ne mogu direktno da pristupe URL-ovima aplikacije,
- korisnik sa pogrešnom ulogom (npr. student) ne može da pristupi administratorskim stranicama.

5. U slučaju odjave, poziva se skript `logout.php` koji:

- poziva `session_start()`,
- uklanja podatke iz sesije pomoću `session_unset()` i `session_destroy()`,
- preusmerava korisnika nazad na `index.php`.

Ovaj tok osigurava da je svaki modul jasno odvojen i da se prava pristupa proveravaju na serverskoj strani, a ne samo vizuelno u interfejsu.

5.4. Organizacija koda i direktorijuma

Radi preglednosti i lakšeg održavanja, kod aplikacije je organizovan u više direktorijuma unutar `public_html/zmaj-fakultet`:

- `index.php` - početna stranica i login forma,
- `config/` - konfiguracioni fajlovi:
 - `db.php` - konekcija na bazu podataka (PDO),
- `admin/` - sve stranice i skripte za administratorski modul:
 - `dashboard.php`,
 - `users.php`, `user_create.php`, `user_edit.php`,
`user_delete.php`,

- `courses.php`, `course_create.php`, ...
- `exam_periods.php`,
- `announcements.php`,
- `statistics.php`, itd.
- `professor/` - stranice za profesore:
 - `dashboard.php`,
 - `courses.php`, `course_students.php`,
 - `exams.php`, `exam_students.php`,
 - `materials.php`, itd.
- `student/` - stranice za studente:
 - `dashboard.php`,
 - `courses.php`, `my_courses.php`,
 - `exams.php`,
 - `grades.php`,
 - `announcements.php`,
 - `profile.php`, itd.
- `assets/` - statički resursi:
 - `css/custom.css` (ili `zma-j-theme.css`) - prilagođena tema,
 - eventualno `img/` sa favicon-om i profilnim slikama.

Ovakva struktura jasno razdvaja:

- konfiguraciju (konekcija na bazu),
- logiku po ulogama (admin/professor/student),
- statičke resurse (CSS, slike).

To vrlo olakšava i studijsko ocenjivanje projekta, jer profesor može jednostavno da “ode” u odgovarajući folder i vidi sav kôd koji se tiče jedne uloge.

5.5. Pregled glavnih tokova korišćenja

Na osnovu prethodnih dijagrama i opisa, mogu se izdvojiti tri ključna toka (use-case scenarija):

1. Tok administratora:

- prijava na sistem kao admin,
- otvaranje dashboard-a,
- kreiranje novog studenta (unos osnovnih podataka, e-maila i lozinke),
- kreiranje novog predmeta i dodela profesora,
- definisanje ispitnog roka,
- objava obaveštenja (npr. „Otvoren je jun 2025 rok“),
- pregled systemske statistike (broj studenata, prijava po predmetu itd.).

2. Tok profesora:

- prijava na sistem kao profesor,
- pregled sopstvenih predmeta,
- pregled liste upisanih studenata po predmetu,
- prelazak na stranicu za ispite,
- izbor aktivnog roka i konkretnog predmeta,
- pregled prijavljenih studenata na ispit,
- unos ocena i automatsko ažuriranje statusa prijave,
- postavljanje materijala za naredno predavanje.

3. Tok studenta:

- prijava na sistem kao student,

- prikaz dashboard-a sa osnovnim informacijama (broj upisanih predmeta, aktivnih rokova, položenih ispita),
- upis na nove predmete ili odjava sa predmeta (ako je dozvoljeno),
- pregled dostupnih ispitnih rokova,
- prijava na ispit za određeni predmet,
- pregled obaveštenja i materijala,
- nakon unosa ocene od strane profesora - pregled istorije polaganja i ostvarenih ocena.

Ovi tokovi demonstriraju kako arhitektura sistema i podela na module direktno podržavaju praktične scenarije iz realnog fakultetskog okruženja.

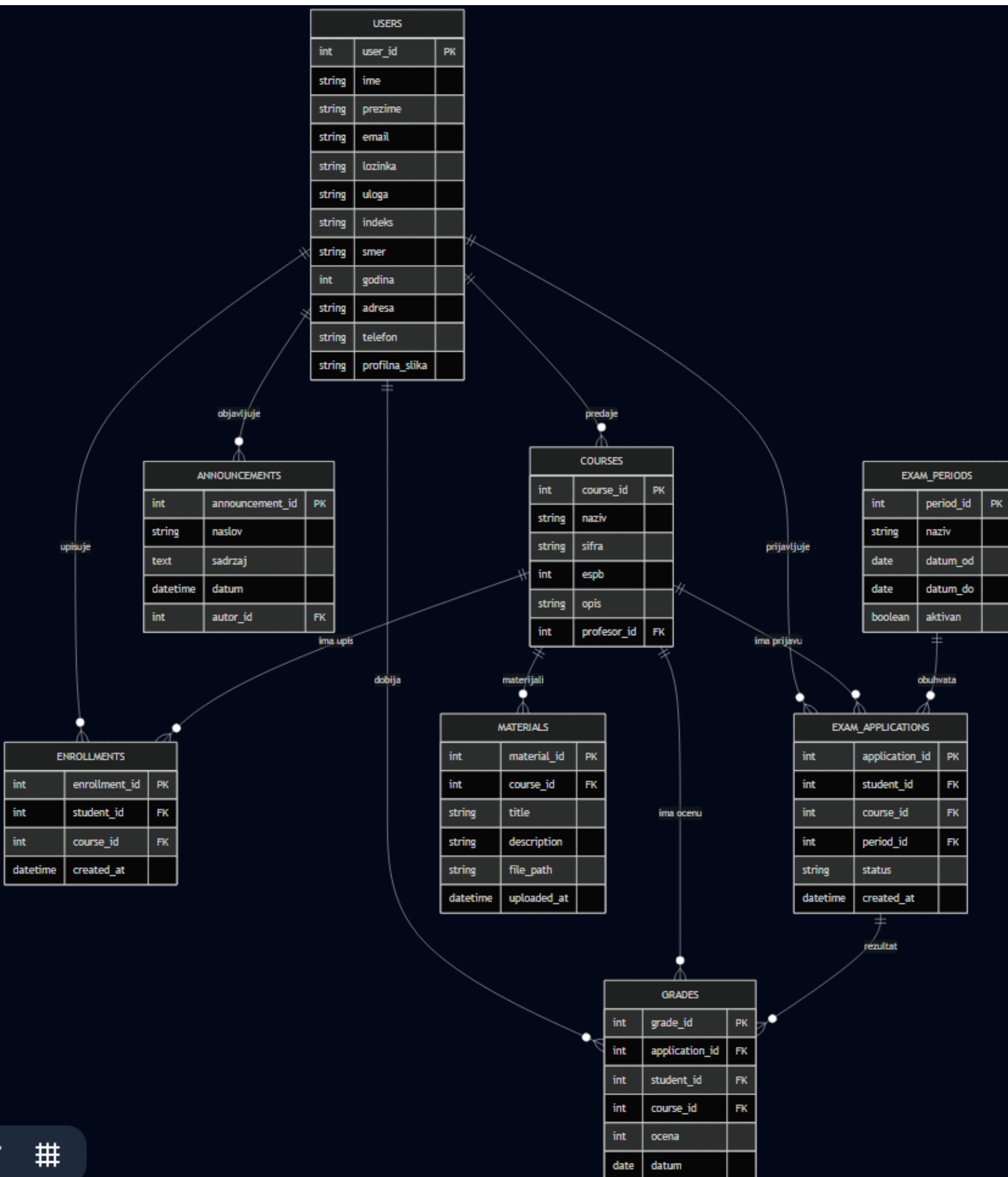
6. Model baze podataka

Baza podataka predstavlja „srce“ sistema „**Zmaj University**“, jer u njoj se trajno čuvaju svi ključni podaci: informacije o korisnicima, predmetima, ispitnim rokovima, prijavama, ocenama i materijalima. Kvalitetan model baze direktno utiče na stabilnost i jednostavnost razvoja aplikacije - ako je baza dobro osmišljena, PHP kod je jednostavniji, logika je jasnija i postoji manja verovatnoća za nekonzistentne podatke.

Model baze podataka u ovom projektu zasnovan je na relacionom modelu, gde su podaci organizovani u tabele povezane preko primarnih i stranih ključeva. Posebna pažnja posvećena je:

- normalizaciji podataka (izbegavanje dupliranja),
- jasno definisanim vezama (1:n, n:m),
- i referencijalnom integritetu (FOREIGN KEY ograničenja).

Na sledećoj slici prikazan je ER dijagram baze podataka koji je izrađen u skladu sa funkcionalnim zahtevima aplikacije.



Slika 13: ER dijagram baze podataka sistema „Zmaj University“

Na dijagramu se jasno vide glavne tabele:

- `users`
- `courses`
- `enrollments`
- `exam_periods`
- `exam_applications`
- `grades`
- `announcements`
- `materials`

uvezane preko primarnih i stranih ključeva u konzistentnu celinu.

6.1. Tabela **users** - korisnici sistema

Tabela **users** predstavlja centralno mesto za čuvanje svih korisnika sistema - administratora, profesora i studenata. Umesto da postoje odvojene tabele za svakog, koristi se jedna tabela, a polje `u_log` (role) određuje tip korisnika.

Tipična struktura tabele `users` obuhvata sledeća polja:

- `user_id` - primarni ključ (INT, AUTO_INCREMENT), jedinstveni identifikator korisnika;
- `ime` - ime korisnika;
- `prezime` - prezime korisnika;
- `adresa` - adresa stanovanja (opciono);
- `telefon` - kontakt telefon (opciono);
- `email` - e-mail adresa koja se koristi za login; postavljeno je ograničenje da bude **jedinstvena** (UNIQUE), jer dva korisnika ne smeju imati isti login;
- `lozinka` - heš lozinka korisnika (npr. SHA-256 heš), nikada se ne čuva u običnom tekstu;

- `uloga` - tekstualno polje (npr. `admin`, `profesor`, `student`);
- `indeks` - broj indeksa (primenjuje se prvenstveno za studente);
- `smer` - naziv studijskog smera (npr. "Internet tehnologije");
- `godina` - godina studija (1, 2, 3, ...);
- opciono: `profile_image` - putanja do profilne slike (ako je student/profesor postavi).

Ova tabela je povezana sa drugim tabelama:

- kao **profesor**: preko polja `profesor_id` u tabeli `courses`;
- kao **student**: preko polja `student_id` u tabelama `enrollments`, `exam_applications` i `grades`;
- kao **autor**: može biti povezan sa tabelom `announcements` (npr. `created_by`).

id	ime	prezime	adresa	telefon	profilna_slika	email	lozinka	uloga	indeks	smer	godina	crea
1	Admin	User	NULL	NULL	NULL	admin@zmaj.univ	240be518fabd2724ddb6f04eeb1da5967448d7e831c08c8fa8...	admin	NULL	NULL	NULL	2025
2	Milan	Petrović	NULL	NULL	NULL	milan.petrovic@zmaj.univ	00624b02e1f9b996a3278f559d5d55313552ad2c0bafc82adf...	profesor	NULL	Informatika	NULL	2025
3	Ana	Jovanović	NULL	NULL	NULL	ana.jovanovic@zmaj.univ	00624b02e1f9b996a3278f559d5d55313552ad2c0bafc82adf...	profesor	NULL	Matematika	NULL	2025
4	Nikola	Ilić	NULL	NULL	NULL	nikola.ilic@zmaj.univ	00624b02e1f9b996a3278f559d5d55313552ad2c0bafc82adf...	profesor	NULL	Elektronika	NULL	2025
5	Marko	Kostić	Homeless	+49 999	uploads/avatars/student_5_1763269226.png	s001@zmaj.univ	19b9dd3e24fad97f47400340f81e118ca3f88be2ee3503b34b...	student	IT001	Internet Tehnologije	1	2025
6	Jovana	Milovanović	NULL	NULL	NULL	s002@zmaj.univ	19b9dd3e24fad97f47400340f81e118ca3f88be2ee3503b34b...	student	IT002	Internet Tehnologije	2	2025
7	Petar	Stojanović	NULL	NULL	NULL	s003@zmaj.univ	19b9dd3e24fad97f47400340f81e118ca3f88be2ee3503b34b...	student	IT003	Softversko Inženjerstvo	3	2025
8	Sara	Nikolić	NULL	NULL	NULL	s004@zmaj.univ	19b9dd3e24fad97f47400340f81e118ca3f88be2ee3503b34b...	student	IT004	Računarske Mreže	1	2025
34	Ivana	Savić	NULL	NULL	NULL	ivana.savic@zmaj.univ	00624b02e1f9b996a3278f559d5d55313552ad2c0bafc82adf...	profesor	NULL	Softversko Inženjerstvo	NULL	2025
35	Dejan	Blagojević	NULL	NULL	NULL	dejan.blagojevic@zmaj.univ	00624b02e1f9b996a3278f559d5d55313552ad2c0bafc82adf...	profesor	NULL	Internet tehnologije	NULL	2025
36	Nataša	Bogdanović	NULL	NULL	NULL	natasa.bogdanovic@zmaj.univ	00624b02e1f9b996a3278f559d5d55313552ad2c0bafc82adf...	profesor	NULL	Računarske mreže	NULL	2025
37	Marko	Ristić	NULL	NULL	NULL	marko.ristic@zmaj.univ	00624b02e1f9b996a3278f559d5d55313552ad2c0bafc82adf...	profesor	NULL	Elektronika	NULL	2025
38	Goran	Janković	NULL	NULL	NULL	goran.jankovic@zmaj.univ	00624b02e1f9b996a3278f559d5d55313552ad2c0bafc82adf...	profesor	NULL	Informatika	NULL	2025
39	Tamara	Milošević	NULL	NULL	NULL	tamara.milosevic@zmaj.univ	00624b02e1f9b996a3278f559d5d55313552ad2c0bafc82adf...	profesor	NULL	Matematika	NULL	2025
40	Vladimir	Kostić	NULL	NULL	NULL	vladimir.kostic@zmaj.univ	00624b02e1f9b996a3278f559d5d55313552ad2c0bafc82adf...	profesor	NULL	Informacioni sistemi	NULL	2025
41	Luka	Petrović	NULL	NULL	NULL	s010@zmaj.univ	19b9dd3e24fad97f47400340f81e118ca3f88be2ee3503b34b...	student	IT010	Internet tehnologije	1	2025
42	Mina	Jovanović	NULL	NULL	NULL	s011@zmaj.univ	19b9dd3e24fad97f47400340f81e118ca3f88be2ee3503b34b...	student	IT011	Internet tehnologije	1	2025
43	Nikola	Stefanović	NULL	NULL	NULL	s012@zmaj.univ	19b9dd3e24fad97f47400340f81e118ca3f88be2ee3503b34b...	student	IT012	Internet tehnologije	2	2025
44	Sara	Pavlović	NULL	NULL	NULL	s013@zmaj.univ	19b9dd3e24fad97f47400340f81e118ca3f88be2ee3503b34b...	student	IT013	Internet tehnologije	2	2025
45	Jelena	Mitić	NULL	NULL	NULL	s014@zmaj.univ	19b9dd3e24fad97f47400340f81e118ca3f88be2ee3503b34b...	student	IT014	Internet tehnologije	3	2025
46	Vuk	Đorđević	NULL	NULL	NULL	s015@zmaj.univ	19b9dd3e24fad97f47400340f81e118ca3f88be2ee3503b34b...	student	IT015	Internet tehnologije	3	2025
sole	Stjepan	Štambur	NULL	NULL	NULL	s016@zmaj.univ	10b6d43e2346467f47400340f81e118ca3f88be2ee3503b34b...	student	IT016	Softversko	1	2025

Slika 14: Struktura tabele users u phpMyAdmin-u

Ovakav pristup (jedna tabela za sve uloge) pojednostavljuje login logiku i rad sa sesijama, jer se nakon prijave uz ulogu iz iste tabele odlučuje da li je korisnik admin, profesor ili student.

6.2. Tabela courses - predmeti

Tabela **courses** sadrži informacije o svim predmetima koji postoje na fakultetu u okviru sistema. Svaki predmet ima svog odgovornog profesora i može imati određeni broj ESPB bodova.

Glavna polja:

- **course_id** - primarni ključ, jedinstveni identifikator predmeta;
- **naziv** - puni naziv predmeta (npr. “Web programiranje 1”);
- **sifra** - kratka šifra predmeta (npr. “WP1”);
- **espb** - broj ESPB bodova (tipično 4, 6, 8, ...);

- `opis` - tekstualni opis predmeta (opciono);
- `profesor_id` - strani ključ ka `users.user_id`, koji označava profesora nosioca predmeta.

Veza između `courses` i `users` (kao profesora) je tipa **n:1** - više predmeta može imati istog profesora, ali svaki predmet ima tačno jednog nosioca.

Tabela `courses` je dalje povezana sa:

- `enrollments` (upis studenata na predmet),
- `exam_applications` (prijave ispita za određeni predmet i rok),
- `grades` (ocena studenta iz tog predmeta),
- `materials` (nastavni materijali za taj predmet).

6.3. Tabela `enrollments` - upisi studenata na predmete

Tabela **`enrollments`** predstavlja vezu između studenata i predmeta i realizuje tipičan **n:m odnos** (više studenata može slušati više predmeta). Umesto da se ovaj odnos direktno modeluje u dve tabele, uvodi se posebna relaciona tabela.

Glavna polja:

- `enrollment_id` - primarni ključ (INT, AUTO_INCREMENT);
- `student_id` - strani ključ ka `users.user_id`, gde je uloga korisnika student;
- `course_id` - strani ključ ka `courses.course_id`;
- `created_at` - datum i vreme kada se student upisao na predmet.

Da bi se obezbedilo da se student ne može **duplo** upisati na isti predmet, tipično se uvodi jedinstveno ograničenje (UNIQUE) nad kombinacijom:

- (`student_id`, `course_id`).

Na ovaj način, baza sama sprečava dupliranje podataka, što olakšava logiku u PHP kodu. Kada student upiše predmet, kreira se novi zapis u `enrollments`. Kada se odjavi, taj zapis

može biti obrisan, uz eventualna ograničenja (npr. zabrana brisanja ako već postoji ocena u tabeli `grades`).

6.4. Tabela `exam_periods` - ispitni rokovi

Tabela `exam_periods` sadrži informacije o ispitnim rokovima kroz koje se ispiti polažu (npr. „Januar 2025“, „Jun 2025“).

Polja:

- `period_id` - primarni ključ, identifikator roka;
- `naziv` - naziv roka (npr. „Januar 2025“);
- `datum_od` - datum početka roka;
- `datum_do` - datum završetka roka;
- `aktivan` - logičko polje (npr. `TINYINT`), koje označava da li je rok trenutno aktivan.

Administrator može kreirati više rokova unapred, ali studentima i profesorima su obično interesantni oni rokovi koji su **trenutno aktivni** ili koji uključuju današnji datum (zbog prijave ispita i unosa ocena).

Tabela `exam_periods` je povezana sa tabelom `exam_applications`, gde se za svaki zapis navodi i `period_id`.

6.5. Tabela `exam_applications` - prijave ispita

Tabela `exam_applications` predstavlja prijavu studenta na ispit iz određenog predmeta u odabranom ispitnom roku.

Glavna polja:

- `application_id` - primarni ključ;
- `student_id` - strani ključ ka `users.user_id` (student);
- `course_id` - strani ključ ka `courses.course_id`;
- `period_id` - strani ključ ka `exam_periods.period_id`;
- `status` - status prijave (npr. `prijavljen`, `polozio`, `pao`);

- `created_at` - datum i vreme kada je prijava kreirana.

U većini slučajeva, želi se ograničiti da student može imati **najviše jednu** prijavu po predmetu u datom roku. To se postiže jedinstvenim ograničenjem:

- `(student_id, course_id, period_id)`.

Veza:

- jedan `exam_periods` može imati više `exam_applications`,
- jedan `course` može imati više prijava po različitim periodima,
- jedan `student` može imati više prijava (za različite predmete ili rokove).

Ova tabela je ključna za profesorski modul, jer profesor na osnovu nje vidi ko je prijavljen na ispit i unosi ocenu.

6.6. Tabela **grades** - ocene

Tabela **grades** čuva informacije o konačnim ocenama studenata na predmetima.

Konačna verzija tabele (nakon dorada tokom razvoja) povezana je i sa tabelom `exam_applications`, što obezbeđuje jasan trag o tome iz koje prijave je proizašla konkretna ocena.

Polja:

- `grade_id` - primarni ključ;
- `application_id` - strani ključ ka `exam_applications.application_id` (povezuje ocenu sa konkretnom prijavom);
- `student_id` - strani ključ ka `users.user_id`;
- `course_id` - strani ključ ka `courses.course_id`;
- `ocena` - numerička ocena (celobrojno, najčešće u opsegu 5-10);
- `datum` - datum kada je ocena upisana (ili potvrđena).

Veza sa `exam_applications` je posebno bitna:

- obezbeđuje referencijalni integritet - ne može se uneti ocena za nepostojeću prijavu,

- omogućava da se eventualno kasnije prati istorija više prijava (ako se model dodatno razvija - npr. više izlazaka na ispit),
- u projektu je primenjen princip da jedna prijava ima najviše jednu konačnu ocenu.

Pored toga, preko `student_id` i `course_id` moguće je brzo pretražiti sve ocene jednog studenta ili sve ocene iz jednog predmeta.



	grade_id	application_id	student_id	course_id	ocena	datum	created_at
	4	3	5	2	7	2025-11-16	2025-11-16 03:29:40

Slika 15: Primer sadržaja tabele grades u phpMyAdmin-u

6.7. Tabela **announcements** - obaveštenja

Tabela **announcements** služi za čuvanje globalnih obaveštenja u sistemu, koja su najčešće vidljiva svim korisnicima (studentima, profesorima, administratorima).

Tipična struktura:

- `announcement_id` - primarni ključ;
- `title` - naslov obaveštenja (kratak i jasan);
- `content` - sadržaj obaveštenja (tekstualni opis);
- `created_at` - datum i vreme kreiranja obaveštenja;
- `created_by` - strani ključ ka `users.user_id` (najčešće administrator).

U ovoj verziji sistema obaveštenja su **globalna**, što znači da nisu vezana za konkretan predmet. U potencijalnim proširenjima moglo bi se dodati polje `course_id` kako bi se omogućila **kurs-specifična obaveštenja** vidljiva samo studentima upisanim na dati predmet.

6.8. Tabela **materials** - nastavni materijali

Tabela **materials** omogućava da se nastavni materijali (npr. PDF prezentacije, skripte, zadaci) organizovano čuvaju i povežu sa određenim predmetima.

Polja:

- `material_id` - primarni ključ;

- `course_id` - strani ključ ka `courses.course_id`;
- `title` - naslov materijala (npr. “Predavanje 1 - Uvod u HTML”);
- `description` - kratki opis (opciono);
- `file_path` - putanja do fajla na disku (npr. `/uploads/materials/wp1_predavanje1.pdf`);
- `uploaded_at` - datum i vreme postavljanja materijala.

Veza sa `courses` je tipa **n:1** - više materijala može biti vezano za jedan predmet, ali svaki materijal pripada tačno jednom predmetu. Kroz korisnički interfejs:

- profesori i/ili administratori imaju mogućnost dodavanja materijala,
- studenti mogu da ih pregledaju i preuzimaju, ali ne i da menjaju ili brišu.

6.9. Veze i integritet podataka

Sumarno, veze između tabela mogu se opisati na sledeći način:

- **users - courses**
 - 1:n (jedan profesor - više predmeta) preko `courses.profesor_id`.
- **users - enrollments - courses**
 - n:m (više studenata prati više predmeta) realizovano preko tabele `enrollments`.
- **exam_periods - exam_applications**
 - 1:n (jedan ispitni rok - više prijava).
- **users - exam_applications - courses**
 - student prijavljuje ispit iz konkretnog predmeta u konkretnom roku (kombinacija `student_id`, `course_id`, `period_id`).
- **exam_applications - grades**
 - 1:1 ili 1:n u zavisnosti od modela; u ovoj implementaciji je jednoznačno povezano tako da jedna prijava ima najviše jednu ocenu.

- **courses - materials**

- 1:n (jedan predmet - više nastavnih materijala).

Korišćenjem **FOREIGN KEY** ograničenja u MySQL-u obezbeđeno je da:

- ne može postojati prijava za nepostojećeg studenta ili nepostojeći predmet,
- ne može postojati ocena za prijavu koja ne postoji,
- ne može postojati materijal za nepostojeći predmet,
- eventualno brisanje korisnika/predmeta/roka može biti ograničeno ili kaskadno, u zavisnosti od poslovne logike.

Ovakav model baze je dovoljno robustan da podrži sve funkcionalnosti koje aplikacija pruža. Istovremeno je i dovoljno jednostavan da se lako razume i proširi dodatnim entitetima (npr. raspored, prisustvo na vežbama, finansijska evidencija).

7. Implementacija aplikacije

U prethodnim poglavljima definisani su funkcionalni zahtevi, arhitektura sistema i model baze podataka. U ovom poglavlju fokus je na praktičnoj implementaciji aplikacije „**Zmaj University**“: organizaciji PHP fajlova, načinu na koji se obrađuju zahtevi, i ključnim ekranima za svaku ulogu (administrator, profesor, student).

Aplikacija je implementirana tako da:

- svaki modul ima svoj skup PHP skripti u posebnom direktorijumu (admin, professor, student),
- sve skripte dele zajedničku konfiguraciju konekcije na bazu (config/db.php),
- kontrola pristupa se vrši na početku svake skripte preko sesije i polja uloga,

- korisnički interfejs koristi Bootstrap 5 komponente, uz dodatni `custom.css` koji uvodi tamnu temu i vizuelni identitet „Zmaj University“.

7.1. Početna stranica i proces prijave na sistem

Prvi kontakt korisnika sa aplikacijom dešava se na početnoj stranici `index.php`, koja prikazuje login formu. Ova stranica je dostupna svim posetiocima, dok su ostale skripte zaštićene proverom sesije.

Login forma sadrži:

- polje za **e-mail adresu**,
- polje za **lozinku**,
- dugme za slanje forme.

Zmaj University
Faculty Information System

E-mail
admin@zmaj.univ

Password
Enter your password

Sign in

Test accounts – click the email or password to copy:

ADMIN admin@zmaj.univ / admin123

PROFESSOR milan.petrovic@zmaj.univ / prof123

STUDENT s001@zmaj.univ / stud123

Click any credential above to copy it to clipboard.

 **Open project documentation**
Seminar paper & system documentation · opens in a new tab

© 2025 Zmaj University

Slika 16: Početna stranica sa login formom

Nakon što korisnik unese svoje podatke i pošalje formu, PHP skript:

1. Poziva `session_start()` kako bi se omogućio rad sa sesijama.
2. Učitava `config/db.php` kako bi dobio PDO konekciju ka MySQL bazi.
3. Preuzima `email` i `password` iz `$_POST` niza.
4. Pronađe korisnika u tabeli `users` na osnovu e-mail adrese, na primer:

```

$stmt = $pdo->prepare("SELECT * FROM users WHERE email = ?");

$stmt->execute([$email]);

$user = $stmt->fetch();

```

5. Ako korisnik postoji, poredi se heš lozinke iz baze sa hešom lozinke koju je korisnik uneo.

6. Ukoliko se heševi poklapaju, u sesiju se upisuju ključne informacije:

- `$_SESSION['user_id']`,
- `$_SESSION['uloga']` (admin/profesor/student),
- `$_SESSION['ime']`, `$_SESSION['prezime']`.

7. U zavisnosti od uloge vrši se redirekcija:

- administrator - `admin/dashboard.php`,
- profesor - `professor/dashboard.php`,
- student - `student/dashboard.php`.

U slučaju da kredencijali nisu ispravni, korisniku se na istoj stranici prikazuje poruka o grešci (npr. “Pogrešan e-mail ili lozinka”), pri čemu se polje e-mail obično ostavlja popunjeno, a lozinka se briše.

Ovaj mehanizam obezbeđuje da se sve dalje funkcionalnosti izvršavaju samo za autentifikovane korisnike, uz jasnu podelu uloga.

7.2. Administratorski modul

Administratorski modul predstavlja “kontrolni centar” sistema. Svi fajlovi ovog modula smešteni su u direktorijum `admin/` i na početku sadrže proveru da li je korisnik prijavljen i da li ima ulogu `admin`:

```

session_start();

if (!isset($_SESSION['user_id']) || ($_SESSION['uloga'] ?? '') !== 'admin')
{
    header('Location: ../index.php');
    exit;
}

```

```
}  
require __DIR__ . '/../config/db.php';
```

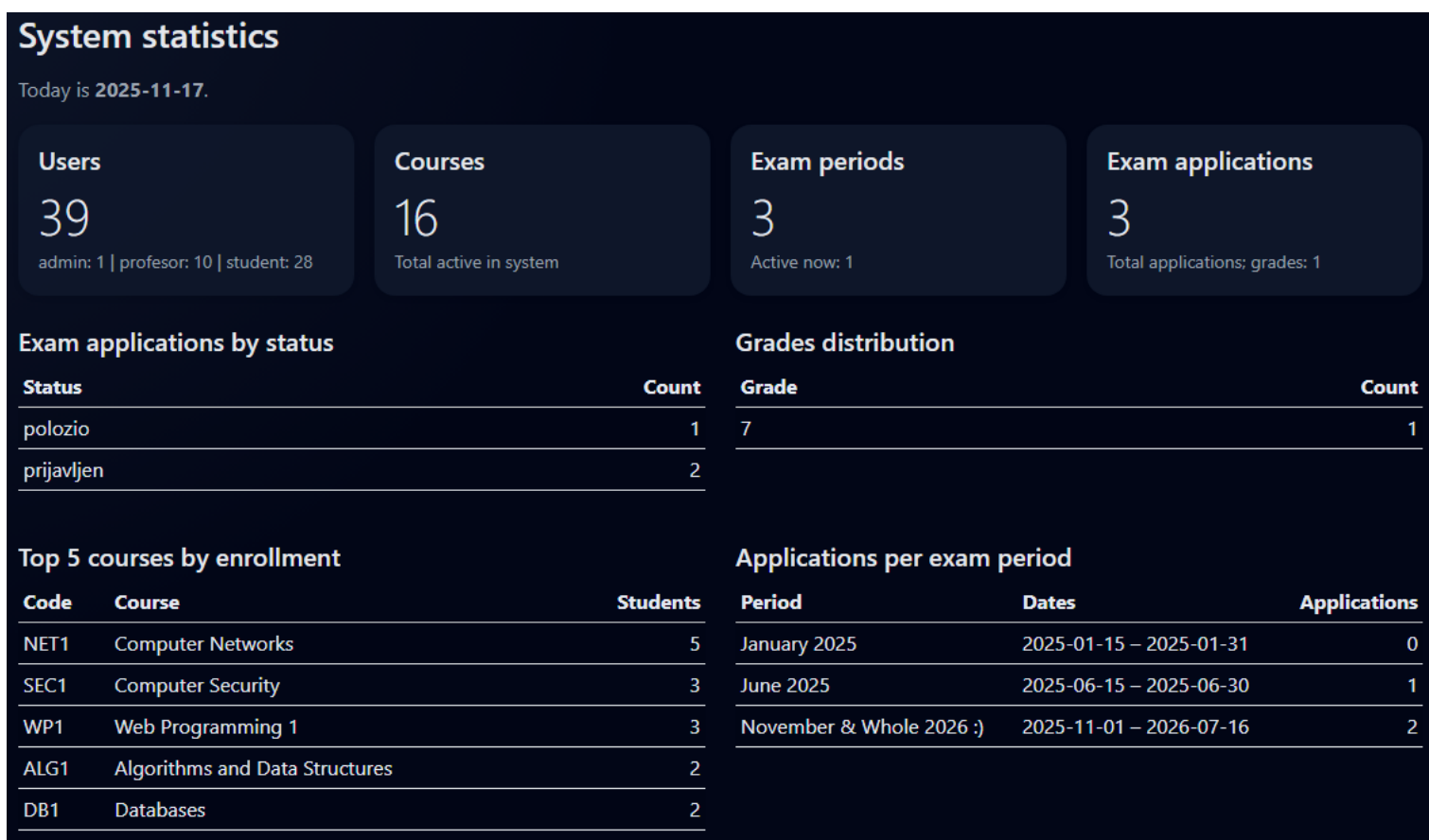
Na ovaj način sprečava se da studenti ili profesori direktnim unosom URL-a pristupe administratorskim funkcijama.

7.2.1. Administratorski dashboard

Stranica `admin/dashboard.php` prikazuje sažet pregled stanja sistema. Tipično sadrži:

- broj registrovanih korisnika po ulogama (studenti, profesori, administratori),
- broj predmeta,
- broj aktivnih ispitnih rokova,
- broj trenutnih prijava na ispite,
- kratku listu najnovijih obaveštenja ili materijala.

Ove informacije prikazane su kroz Bootstrap **kartice (cards)** raspoređene u grid, čime se dobija moderan i pregledan izgled.



Slika 17: Administratorski dashboard sa karticama i statistikom

Upiti na bazi koriste agregatne funkcije (npr. `COUNT( *)`) i grupisanje po ulozi, čime se lako dobijaju brojčani prikazi bez potrebe za dodatnom logikom na klijentu.

7.2.2. Upravljanje korisnicima (`users.php` i povezane stranice)

Stranica `admin/users.php` prikazuje tabelarni pregled svih korisnika uz mogućnost:

- pretrage po imenu, prezimenu ili e-mail-u,
- dodavanja novog korisnika (`user_create.php`),
- izmene postojećeg korisnika (`user_edit.php`),
- brisanja korisnika (`user_delete.php`).

Tabela koristi Bootstrap klase (`table`, `table-striped`, `table-hover`) i ima kolone:

- ID,
- ime i prezime,

- e-mail,
- uloga,
- indeks, smer, godina (za studente),
- akcije (Edit/Delete).

Users

+ New user

Search by name or email

Search

ID	Name	E-mail	Role	Index	Program	Year	Actions
1	Admin User	admin@zmaj.univ	Admin				Edit Delete
35	Dejan Blagojević	dejan.blagojevic@zmaj.univ	Profesor		Internet tehnologije		Edit Delete
36	Nataša Bogdanović	natasa.bogdanovic@zmaj.univ	Profesor		Računarske mreže		Edit Delete
4	Nikola Ilić	nikola.ilic@zmaj.univ	Profesor		Elektronika		Edit Delete
38	Goran Janković	goran.jankovic@zmaj.univ	Profesor		Informatika		Edit Delete
3	Ana Jovanović	ana.jovanovic@zmaj.univ	Profesor		Matematika		Edit Delete
40	Vladimir Kostić	vladimir.kostic@zmaj.univ	Profesor		Informacioni sistemi		Edit Delete
39	Tamara Milošević	tamara.milosevic@zmaj.univ	Profesor		Matematika		Edit Delete
2	Milan Petrović	milan.petrovic@zmaj.univ	Profesor		Informatika		Edit Delete
37	Marko Ristić	marko.ristic@zmaj.univ	Profesor		Elektronika		Edit Delete
34	Ivana Savić	ivana.savic@zmaj.univ	Profesor		Softversko inženjerstvo		Edit Delete
46	Vuk Đorđević	s015@zmaj.univ	Student	IT015	Internet tehnologije	3	Edit Delete

Slika 18: Stranica za upravljanje korisnicima - lista korisnika

Dodavanje korisnika realizovano je kroz formu na stranici `user_create.php`. Kada se forma pošalje:

1. PHP skript validira sva obavezna polja (ime, prezime, e-mail, lozinka, uloga).
2. Proverava se jedinstvenost e-mail adrese u bazi.
3. Lozinka se hešuje (npr. `hash( 'sha256' , $password )`).
4. Podaci se upisuju u tabelu `users` pomoću pripremljenih upita.

Slično, stranica `user_edit.php` preuzima podatke o korisniku po ID-u, popunjava formu i omogućava izmenu. Lozinka se menja samo ako je korisnik unese; u suprotnom, stara lozinka ostaje neizmenjena.

Brisanje korisnika obavlja `user_delete.php`, pri čemu se može dodatno sprečiti da administrator obriše sam sebe (provera `if ($id == $_SESSION['user_id'])`).

7.2.3. Upravljanje predmetima (`courses.php`)

Stranica `admin/courses.php` omogućava:

- pregled svih predmeta,
- dodavanje novih predmeta,
- izmenu i brisanje postojećih predmeta.

Za svaki predmet prikazuju se:

- šifra,
- naziv,
- broj ESPB,
- profesor nosilac.

Courses					+ New course	
Search by name, code or ID					Search	
ID	Code	Name	ESPB	Professor	Actions	
3	ALG1	Algorithms and Data Structures	6	Ana Jovanović	Edit	Delete
10	CA1	Computer Architecture	6	Marko Ristić	Edit	Delete
2	DB1	Databases	6	Milan Petrović	Edit	Delete
9	DB2	Databases 2	6	Ivana Savić	Edit	Delete
17	ENG1	English for IT	4	Tamara Milošević	Edit	Delete
14	IOT1	Internet of Things	6	Dejan Blagojević	Edit	Delete
13	ML1	Machine Learning Basics	6	Nataša Bogdanović	Edit	Delete
12	MOB1	Mobile Development	6	Tamara Milošević	Edit	Delete
4	NET1	Computer Networks	6	Nikola Ilić	Edit	Delete
5	OS1	Operating Systems	6	Ana Jovanović	Edit	Delete
16	PM1	Project Management in IT	6	Vladimir Kostić	Edit	Delete
6	SE1	Software Engineering	6	Milan Petrović	Edit	Delete
11	SEC1	Computer Security	6	Goran Janković	Edit	Delete
15	TEST1	Software Testing	6	Ivana Savić	Edit	Delete

Slika 19: Administratorski pregled predmeta

Prilikom kreiranja i izmene predmeta koristi se padajuća lista (select) za izbor profesora, čiji podaci dolaze iz tabele `users` filtriranjem po ulozi `profesor`. To sprečava ručno unošenje ID-a profesora i smanjuje mogućnost greške.

7.2.4. Upravljanje ispitnim rokovima (`exam_periods.php`)

Stranica `admin/exam_periods.php` sadrži:

- tabelu sa svim definisanim ispitnim rokovima,
- formular za dodavanje novog roka,
- opcije izmene i brisanja.

Za svaki rok čuvaju se naziv, datum početka i datum završetka, kao i informacija da li je rok aktivan (`aktivan = 1`). Administratori mogu:

- definisati više rokova za budućnost,
- aktivirati ili deaktivirati rok,
- voditi računa da istovremeno može biti aktivan samo određeni broj rokova (često jedan ili dva, u zavisnosti od politike).

Exam periods

Search by name

Search

+ New exam period

ID	Name	From	To	Active	Actions
4	November & Whole 2026 :)	2025-11-01	2026-07-16	Active	Edit Delete Deactivate
2	June 2025	2025-06-15	2025-06-30	Inactive	Edit Delete Set active
1	January 2025	2025-01-15	2025-01-31	Inactive	Edit Delete Set active

Slika 20: Stranica za upravljanje ispitnim rokovima

Ove informacije se kasnije koriste u studentskom i profesorskom modulu za prikaz samo aktuelnih rokova u kojima je dozvoljena prijava ispita ili unos ocena.

7.2.5. Obaveštenja (announcements . php)

Administratori objavljuju globalna obaveštenja pomoću stranice `admin/announcements . php`. Interfejs omogućava:

- unos naslova obaveštenja,
- unos sadržaja (kraći tekst),
- pogled na listu već objavljenih obaveštenja,
- brisanje ili izmenu postojećih obaveštenja.

Announcements					+ New announcement	
Search by title or text					Search	
ID	Title	Author	Date	Preview	Actions	
4	Deutschland Uber allen!!	Admin User	2025-11-16 05:08:08	Mein Herz in flammen Will dich lieben und verdammm...	Edit	Delete
1	Dobrodošli na Zmaj fakultet	Admin User	2025-11-16 04:24:43	Dobrodošli u informacioni sistem Zmaj fakulteta. O...	Edit	Delete
2	Početak januarskog ispitnog roka	Admin User	2025-11-16 04:24:43	Januarski ispitni rok počinje 15. januara. Prijava...	Edit	Delete

Slika 21: Administratorski prikaz liste obaveštenja

Objavljena obaveštenja vidljiva su u studentskom i profesorskom dashboard-u, čime se obezbeđuje da sve važne informacije (otvaranje roka, promene termina, administrativna obaveštenja) dođu do korisnika na jednom mestu.

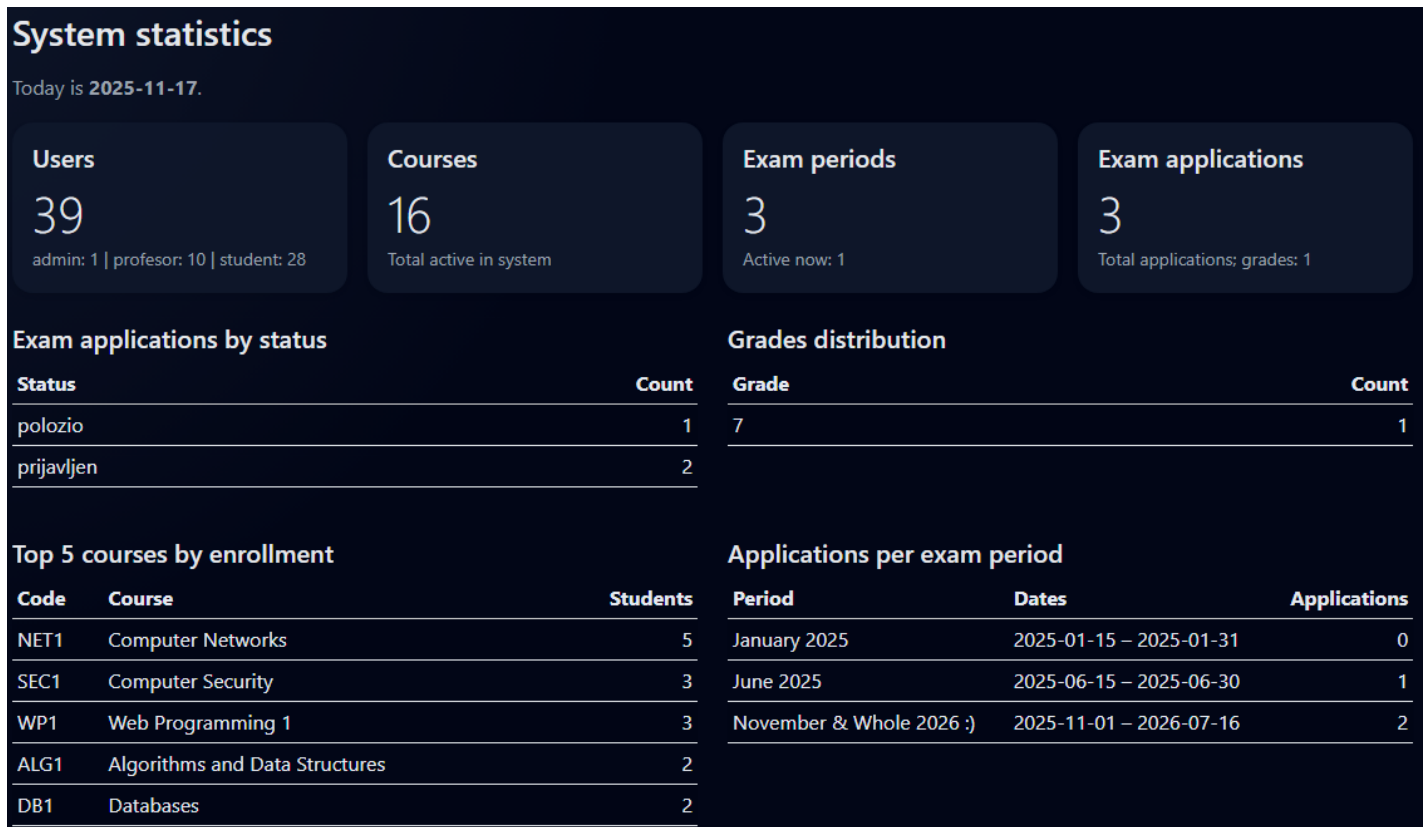
7.2.6. Statistika sistema (**statistics.php**)

Stranica `admin/statistics.php` daje administratoru pregled statističkih podataka o sistemu. Tipični prikazi uključuju:

- ukupan broj korisnika po ulozi,
- broj predmeta,
- broj upisa na predmete (**enrollments**),
- broj prijava na ispit po rokovima,

- broj položenih ispita (na osnovu tabele **grades**).

Podaci se prikazuju u formi tabela, a po potrebi i kartica ili jednostavnih grafikona (u ovoj verziji sistema naglasak je na tabelarnom prikazu). Statistika pomaže administratoru da stekne uvid kako se sistem koristi i koliko je opterećen.



Slika 22: Administratorska stranica sa statistikom sistema

7.3. Profesorski modul

Profesorski modul nalazi se u direktorijumu `professor/` i obuhvata sve funkcionalnosti koje su potrebne nastavniku - od pregleda studenata po predmetima do unosa konačnih ocena.

Na početku svake stranice nalazi se provera:

```
session_start();

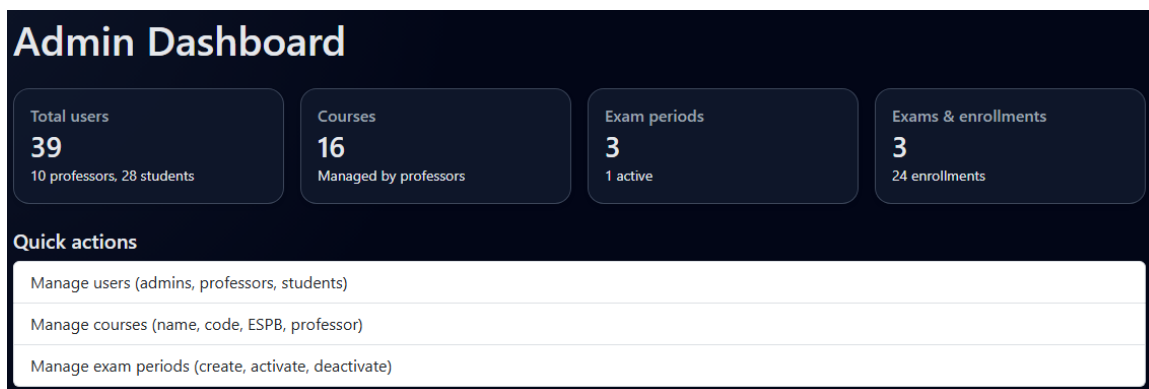
if (!isset($_SESSION['user_id']) || ($_SESSION['uloga'] ?? '') !== 'profesor') {
    header('Location: ../index.php');
    exit;
```

```
}  
require __DIR__ . '/../config/db.php';
```

7.3.1. Proforski dashboard

Stranica `professor/dashboard.php` prikazuje kratak pregled:

- broja predmeta na kojima je profesor nosilac,
- broja studenata upisanih na njegove predmete,
- broja prijava na ispit u aktivnim rokovima.



Slika 23: Proforski dashboard sa osnovnim statistikama

Ovde se takođe mogu prikazivati globalna obaveštenja koja je objavio administrator.

7.3.2. Predmeti i studenti (`courses.php`, `course_students.php`)

Stranica `professor/courses.php` prikazuje listu svih predmeta koje drži dati profesor. Podaci se učitavaju iz tabele `courses` filtriranjem po `profesor_id = $_SESSION['user_id']`.

Za svaki predmet moguća je akcija „View students“ ili slično, koja vodi na stranicu `course_students.php?course_id=...` Na toj stranici prikazana je tabela:

- indeks studenta,
- ime i prezime,
- smer i godina,
- eventualne dodatne informacije (e-mail).

Students in course: WP1 - Web Programming 1

ESPB: 6 | Total students: 3

« Back to my courses

#	Index	Name	E-mail	Program	Year	Enrolled on
1	IT010	Luka Petrović	s010@zmaj.univ	Internet tehnologije	1	2025-11-16 05:39:04
2	IT011	Mina Jovanović	s011@zmaj.univ	Internet tehnologije	1	2025-11-16 05:39:04
3	IT032	Katarina Đurić	s032@zmaj.univ	Internet tehnologije	4	2025-11-16 05:39:04

Slika 24: Pregled studenata po predmetu u profesorskom modulu

Ovaj pogled omogućava profesoru da brzo vidi ko prati njegov predmet i da, po potrebi, proveri da li su svi studenti pravilno upisani.

7.3.3. Ispiti i ocenjivanje (exams.php, exam_students.php)

Stranica professor/exams.php kombinuje informacije o:

- aktivnim ispitnim rokovima,
- predmetima koje profesor drži,
- broju prijavljenih studenata po predmetu u izabranom roku.

Profesor bira:

1. ispitni rok iz padajuće liste (filtrirani samo aktivni rokovi),
2. predmet za koji želi da vidi prijave.

Klikom na dugme „View students“ otvara se stranica exam_students.php, koja prikazuje tabelu studenata prijavljenih na ispit za taj predmet i rok:

- indeks,
- ime i prezime,
- smer,
- status prijave (prijavljen, položio, pao),
- eventualno postojeća ocena i datum.

Za svakog studenta profesor ima formu u kojoj može izabrati ocenu (od 5 do 10) i snimiti je. Nakon snimanja:

- zapis u tabeli `grades` se kreira ili ažurira,
- status u tabeli `exam_applications` postavlja se na `polozio` (ako je ocena ≥ 6) ili `pao` (ako je ocena 5),
- stranica se ponovo učitava (POST/Redirect/GET pattern) kako bi se izbegla ponovna obrada iste forme pri osvežavanju.



Code	Course	ESPB	Applications in this period	Action
DB1	Databases	6	1	View students
SE1	Software Engineering	6	0	View students
WP1	Web Programming 1	6	0	View students
WP2	Web Programming 2	6	0	View students

Slika 25: Stranica za pregled prijava i unos ocena u profesorskom modulu

Dodatno, u kodu se proverava da li je trenutni datum unutar opsega izabranog roka (`datum_od` $\leq$ `danas` $\leq$ `datum_do`). Ukoliko nije, unos ocena se onemogućava i korisniku se prikazuje poruka da ocenjivanje nije dozvoljeno van trajanja roka.

7.3.4. Nastavni materijali (`materials.php`)

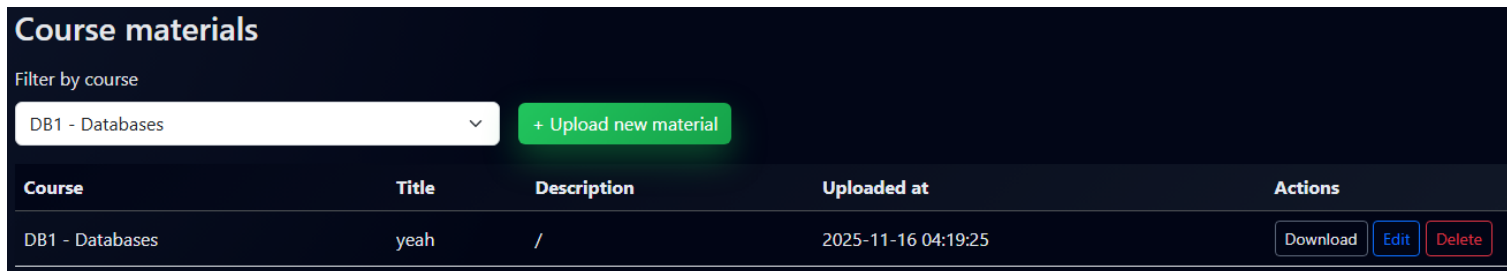
Stranica `professor/materials.php` omogućava profesorima da dodaju, pregledaju i brišu materijale po predmetima.

Tipičan tok rada:

1. Profesor odabere predmet iz padajuće liste.
2. Popunjava naslov materijala i opcioni opis.
3. Izabere fajl sa lokalnog računara (npr. PDF, PPTX).

4. Klikom na „Upload“ fajl se šalje na server, smešta u zadati direktorijum, a u tabelu `materials` upisuje se: `course_id`, `title`, `description`, `file_path` i `uploaded_at`.

Na istoj stranici nalazi se tabela sa već postavljenim materijalima za izabrani predmet, uz linkove za preuzimanje i dugmad za brisanje.



Course	Title	Description	Uploaded at	Actions
DB1 - Databases	yeah	/	2025-11-16 04:19:25	Download Edit Delete

Slika 26: Stranica za postavljanje i pregled nastavnih materijala

7.4. Studentski modul

Studentski modul nalazi se u direktorijumu `student/` i predstavlja deo sistema koji je najčešće korišćen u realnoj upotrebi, jer obuhvata svakodnevne aktivnosti studenata (upis predmeta, prijava ispita, pregled ocena, materijala i obaveštenja).

Na početku svake stranice sledi standardna provera:

```
session_start();

if (!isset($_SESSION['user_id']) || ($_SESSION['uloga'] ?? '') !== 'student') {
    header('Location: ../index.php');
    exit;
}

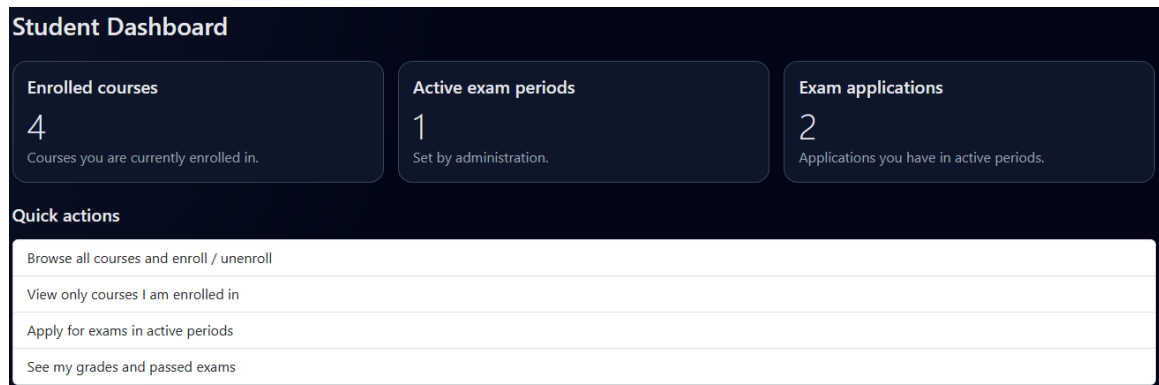
require __DIR__ . '/../config/db.php';
```

7.4.1. Studentski dashboard (`dashboard.php`)

Na stranici `student/dashboard.php` student dobija:

- kratku statistiku: broj upisanih predmeta, broj aktivnih rokova, broj prijavljenih ispita, broj položenih predmeta,
- listu najnovijih obaveštenja (preuzetu iz `announcements`),

- eventualno linkove ka najčešće korišćenim funkcijama (upis na predmete, prijava ispita, pregled ocena).



Slika 27: Studentski dashboard - sažet pregled informacija

Ovaj ekran služi kao “početna tačka” za sve ostale studentske aktivnosti.

7.4.2. Upis i pregled predmeta (courses.php, my_courses.php)

Stranica student/courses.php prikazuje listu svih predmeta dostupnih u sistemu.

Student može:

- pretraživati predmete po nazivu ili šifri,
- klikom na dugme „Upiši predmet“ (Enroll) kreirati zapis u tabeli enrollments,
- videti koji predmeti su već upisani (npr. kroz status ili onemogućeno dugme).

Courses & enrollments

Code	Name	ESPB	Professor	Status	Action
ALG1	Algorithms and Data Structures	6	Ana Jovanović	Enrolled	Unenroll
CA1	Computer Architecture	6	Marko Ristić	Not enrolled	Enroll
DB1	Databases	6	Milan Petrović	Enrolled	Unenroll
DB2	Databases 2	6	Ivana Savić	Not enrolled	Enroll
ENG1	English for IT	4	Tamara Milošević	Not enrolled	Enroll
IOT1	Internet of Things	6	Dejan Blagojević	Not enrolled	Enroll
ML1	Machine Learning Basics	6	Nataša Bogdanović	Not enrolled	Enroll
MOB1	Mobile Development	6	Tamara Milošević	Not enrolled	Enroll
NET1	Computer Networks	6	Nikola Ilić	Enrolled	Unenroll
OS1	Operating Systems	6	Ana Jovanović	Enrolled	Unenroll
PM1	Project Management in IT	6	Vladimir Kostić	Not enrolled	Enroll
SE1	Software Engineering	6	Milan Petrović	Not enrolled	Enroll
SEC1	Computer Security	6	Goran Janković	Not enrolled	Enroll
TEST1	Software Testing	6	Ivana Savić	Not enrolled	Enroll
WP1	Web Programming 1	6	Milan Petrović	Not enrolled	Enroll

Slika 28: Lista dostupnih predmeta za upis - studentski modul

Stranica `student/my_courses.php` prikazuje samo one predmete na koje je student upisan, uz osnovne informacije (šifra, naziv, profesor, ESPB) i linkove ka pripadajućim materijalima.

7.4.3. Prijava i odjava ispita (`exams.php`)

Stranica `student/exams.php` je ključna za rad sa ispitnim rokovima. Na njoj student:

1. vidi listu aktivnih ispitnih rokova (preuzetu iz `exam_periods` sa `aktivan = 1` i datim opsegom datuma),

2. bira rok iz padajuće liste,
3. vidi svoje upisane predmete i status prijave ispita za svaki od njih u izabranom roku,
4. može da se prijavi na ispit (kreira se zapis u `exam_applications`),
5. može da se odjavi sa ispita, sve dok za tu prijavu nije uneta ocena.

Statusi su jasno označeni bojama (Bootstrap `badge` komponente):

- siva („Not applied“) - student nije prijavljen,
- plava („Applied“) - student je prijavljen,
- zelena („Passed“) - položeno,
- crvena („Failed“) - nije položeno.

Exam registration					
Exam period		Search course			
November & Whole 2026 :) (2025-11-01 – 2026-01-01) ▼		By name or code		Filter	
Code	Course	ESPB	Professor	Status	Action
ALG1	Algorithms and Data Structures	6	Ana Jovanović	Not applied	Apply for exam
DB1	Databases	6	Milan Petrović	Passed	(Cannot change after grading)
NET1	Computer Networks	6	Nikola Ilić	Registered	Cancel application
OS1	Operating Systems	6	Ana Jovanović	Not applied	Apply for exam

Slika 29: Stranica za prijavu i odjavu ispita u studentskom modulu

Ovakav vizuelni prikaz omogućava studentu da na jednom mestu vidi svoj napredak i da ne zaboravi da se prijavi na ispit u okviru roka.

7.4.4. Pregled ocena (`grades.php`)

Stranica `student/grades.php` prikazuje listu svih ocena koje je student ostvario. Podaci se preuzimaju spajanjem tabela `grades`, `courses` i `exam_applications` (po potrebi), čime se dobija:

- naziv predmeta,

- šifra predmeta,
- ocena,
- datum polaganja,
- eventualno ispitni rok.

My grades						
Code	Course	ESPB	Grade	Status	Exam period	Date
DB1	Databases	6	7	Passed	November & Whole 2026 :)	2025-11-16

Slika 30: Stranica sa pregledom ocena studenta

Student na ovaj način ima jasan uvid u svoj akademski napredak i može da prati koliko je ESPB bodova do sada ostvario (ovo se može dodatno izračunati, ako je predviđeno u specifikaciji).

7.4.5. Profil, obaveštenja i materijali

U studentskom modulu obično se nalaze i dodatne stranice:

- `profile.php` - omogućava studentu da promeni kontakt podatke i lozinku (uz proveru stare lozinke), kao i da eventualno postavi profilnu sliku,
- `announcements.php` prikazuje listu globalnih obaveštenja iz `announcements`,
- `materials.php` ili materijali dostupni preko `my_courses.php` - prikaz materijala po predmetima koje student sluša.

Announcements	
Deutschland Uber allen!! By Admin User Mein Herz in flammen Will dich lieben und verdammen Deutschland Dein atem kalt so jung und doch so alt Deutschland	2025-11-16 05:08:08
Dobrodošli na Zmaj fakultet By Admin User Dobrodošli u informaciوني sistem Zmaj fakulteta. Ovde možete pratiti predmete, ispite i obaveštenja.	2025-11-16 04:24:43
Početak januarskog ispitnog roka By Admin User Januarski ispitni rok počinje 15. januara. Prijava ispita je obavezna najkasnije 48h pre polaganja.	2025-11-16 04:24:43

Slika 31: Stranica za pregled obaveštenja iz perspektive studenta

7.5. Validacija, poruke i korisničko iskustvo

Kroz ceo sistem implementirana je osnovna validacija podataka:

- na serverskoj strani se proverava da li su obavezna polja popunjena,
- proverava se format e-mail adrese,
- uloga se ograničava na predefinisane vrednosti (admin, profesor, student),
- onemogućava se brisanje ili izmena zapisa koji bi narušili integritet baze (preko FOREIGN KEY ograničenja ili kroz logiku u PHP-u).

Korisnicima se prikazuju jasne poruke:

- zelene „success“ poruke nakon uspešnih operacija (npr. “User successfully created.”),
- crvene „danger“ poruke pri greškama (npr. “E-mail already in use.”),
- žute „warning“ poruke kod specifičnih situacija (npr. “No active exam periods.”).

Za to se koriste Bootstrap **alert** komponente (alert-success, alert-danger, alert-warning), čime se dodatno unapređuje korisničko iskustvo i jasno komunicira šta se desilo nakon neke akcije.

8. Bezbednost i validacija podataka

Bezbednost web aplikacija je jedan od najvažnijih aspekata modernih informacionih sistema. Iako je projekat „Zmaj University“ razvijen u obrazovne svrhe, pri dizajnu i implementaciji vodilo se računa da se primene osnovne, ali značajne bezbednosne prakse:

- kontrola pristupa i upravljanje sesijama,
- zaštita lozinki,
- zaštita od SQL injection napada,
- validacija korisničkog unosa,
- bezbedno rukovanje fajlovima koje korisnici postavljaju,
- korektno prikazivanje grešaka.

U nastavku su opisane ključne mere koje su primenjene, kao i ograničenja trenutnog rešenja.

8.1. Autentifikacija i autorizacija korisnika

Prvi sloj bezbednosti u sistemu zasniva se na **autentifikaciji** (provera identiteta korisnika) i **autorizaciji** (provera dozvola, tj. šta korisnik sme da radi).

8.1.1. Autentifikacija

Autentifikacija je realizovana pomoću:

- login forme na početnoj stranici (`index.php`),
- tabele `users` u bazi, gde se čuvaju e-mail adrese i heš lozinke,
- PHP sesija (`$_SESSION`) koje čuvaju informacije o prijavljenom korisniku.

Nakon uspešnog logovanja:

- u sesiju se upisuju identifikator korisnika i njegova uloga (`admin`, `profesor` ili `student`),
- korisnik se preusmerava na odgovarajući dashboard (`/admin/dashboard.php`, `/professor/dashboard.php` ili `/student/dashboard.php`).

8.1.2. Autorizacija i zaštita ruta

Svaka stranica u aplikaciji, koja je namenjena samo određenoj ulozi, na samom početku sadrži proveru sesije. Na primer, u administratorskom modu:

```
session_start();

if (!isset($_SESSION['user_id']) || ($_SESSION['uloga'] ?? '') !== 'admin')
{
    header('Location: ../index.php');
    exit;
}
```

Sličan obrazac koristi se i u profesorskom i studentskom modulu, sa odgovarajućom ulogom:

- `($_SESSION['uloga'] ?? '') !== 'profesor'`
- `($_SESSION['uloga'] ?? '') !== 'student'`

Na ovaj način:

- **anonimni korisnici** (koji nisu ulogovani) ne mogu direktno pristupiti unutrašnjim URL-ovima,
- **korisnici sa pogrešnom ulogom** ne mogu pristupiti panelima koji im ne pripadaju (npr. student ne može u admin panel).

Odjava iz sistema realizovana je skriptom `logout.php`, koji:

- pokreće sesiju,
- briše sve podatke iz `$_SESSION`,
- poziva `session_destroy()`,
- preusmerava korisnika nazad na `index.php`.

Ovim se obezbeđuje da ostali korisnici istog računara (npr. u biblioteci) ne mogu da „naslede“ tuđu sesiju.

8.2. Zaštita lozinke

Lozinke su jedan od najosetljivijih tipova podataka u sistemu. Iz tog razloga primenjene su sledeće mere:

1. Lozinke se ne čuvaju u običnom tekstu (plain text)

Kod kreiranja korisnika, uneta lozinka se **hešuje** pre upisa u bazu podataka.

U projektu se koristi SHA-256 heširanje:

```
$hashedPassword = hash('sha256', $password);
```

U bazi se čuva samo rezultat heš funkcije, što znači da čak i u slučaju kompromitovanja baze, originalne lozinke nisu direktno vidljive.

2. Provera lozinke pri logovanju

Kada se korisnik loguje, uneta lozinka se ponovo prolazi kroz istu heš funkciju i poredi sa hešom sačuvanim u bazi:

```
if (hash('sha256', $inputPassword) === $user['lozinka']) {  
    // logovanje uspešno  
}
```

3. Promena lozinke

U okviru studentskog (i potencijalno profesorskog) profila postoji mogućnost promene lozinke, pri čemu se:

- najpre proverava stara lozinka,
- zatim se nova lozinka hešuje na isti način i upisuje u bazu.

Napomena: Za produkcijske sisteme preporučuje se korišćenje naprednijih funkcija kao što su `password_hash()` i `password_verify()` uz algoritme poput bcrypt/argon2, dok je u kontekstu ovog projektnog zadatka primena SHA-256 dovoljna da pokaže razumevanje osnovnog koncepta heširanja.

8.3. Zaštita od SQL injection napada

Jedan od najčešćih napada na web aplikacije sa bazom podataka je **SQL injection**, gde napadač pokušava da ubaci zlonameran SQL kod preko ulaznih polja (formi, parametara u URL-u).

Da bi se ovaj rizik sveo na minimum, u aplikaciji se koriste:

- **pripremljeni upiti (prepared statements),**
- **parametrizovani upiti** preko PDO-a.

Umesto da se vrednosti direktno ubacuju u SQL string preko konkatencije, koristi se sledeći obrazac:

```
$sql = "SELECT * FROM users WHERE email = :email";  
$stmt = $pdo->prepare($sql);  
$stmt->execute(['email' => $email]);  
$user = $stmt->fetch();
```

ili:

```
$stmt = $pdo->prepare("  
    INSERT INTO exam_applications (student_id, course_id, period_id,  
    status, created_at)  
    VALUES (?, ?, ?, 'prijavljen', NOW())  
");  
$stmt->execute([$studentId, $courseId, $periodId]);
```

Prednosti ovog pristupa:

- PDO automatski pravilno „escapuje“ parametre,
- sprečava se da korisnički unos bude interpretiran kao deo SQL sintakse,
- isti upit se može ponovo koristiti sa različitim vrednostima (performanse, preglednost).

Tokom implementacije, vodi se računa da se **svaki upit koji koristi vrednosti iz \$_GET ili \$_POST podataka** izvršava isključivo kao pripremljeni upit, bez direktnog ubacivanja tih vrednosti u string.

8.4. Validacija korisničkog unosa

Pored tehničke zaštite upita, važna je i **validacija podataka** koje korisnici unose. U sistemu je primenjena:

- **serverska validacija** - u PHP kodu,
- opcionalna **osnovna klijentska validacija** - kroz HTML5 attribute i Bootstrap komponente.

8.4.1. Serverska validacija

Na serverskoj strani, pri obradi svake forme primenjuju se sledeće provere:

- obavezna polja ne smeju biti prazna (npr. ime, prezime, e-mail, uloga pri kreiranju korisnika),
- format e-mail adrese proverava se funkcijom `filter_var($email, FILTER_VALIDATE_EMAIL)`,
- uloga (uloga) se proverava da li je jedna od dozvoljenih vrednosti (admin, profesor, student),
- numeričke vrednosti (npr. godina studija, ocena 5-10) proveravaju se da li su u očekivanom opsegu,
- pri prijavi na ispit proverava se da li je student zaista upisan na taj predmet (provera u tabeli `enrollments`),
- pri unosu ocene proverava se da li prijava postoji i pripada datom predmetu, studentu i roku.

Ukoliko neka od provera padne, korisniku se prikazuju odgovarajuće poruke o grešci, obično preko Bootstrap `alert` komponenti, a operacija se prekida.

8.4.2. Klijentska validacija

HTML forme sadrže attribute kao što su:

- `required` - za obavezna polja,
- `type="email"` - za e-mail adrese,
- `min` i `max` za numeričke vrednosti (npr. ocena od 5 do 10),
- `maxlength` - ograničenje dužine unosa.

Ove provere izvršava browser pre slanja forme, čime se poboljšava korisničko iskustvo (korisnik odmah dobija informaciju da je nešto propustio da popuni). Ipak, serverska validacija je neophodna, jer klijentsku validaciju napadač može zaobići.

8.5. Bezbedno rukovanje fajlovima

Sistem omogućava postavljanje fajlova u okviru modula nastavnih materijala (profesorski modul), kao i potencijalno postavljanje profilnih slika (studentski profil). Fajl upload je osetljiva operacija jer napadač može pokušati da postavi zlonamerni skript umesto regularnog dokumenta.

Da bi se minimizovao rizik, primenjene su sledeće mere:

1. Ograničavanje tipa fajla

Pri upload-u materijala dozvoljeni su samo određeni tipovi (npr. PDF, PPTX, DOCX).

Proverava se ekstenzija fajla, a po potrebi i MIME tip:

- `.pdf`, `.ppt`, `.pptx`, `.doc`, `.docx`, `.txt` itd.

2. Ograničavanje veličine fajla

Korišćenjem PHP direktiva i dodatnih provera kontroliše se maksimalna veličina fajla (npr. nekoliko megabajta), kako bi se izbeglo preopterećenje servera.

3. Generisanje jedinstvenog imena fajla

Umesto da se koristi originalno ime fajla (koje može sadržati specijalne karaktere ili konflikte), generiše se jedinstveno ime - npr. spajanjem vrednosti `time()` sa random stringom i originalnom ekstenzijom.

Primer:

```
$newFileName = time() . '_' . bin2hex(random_bytes(4)) . '.' . $ext;
```

4. Smeštanje fajlova u poseban direktorijum

Upload fajlovi se čuvaju u posebnom folderu (npr. `/uploads/materials/`), a u bazu se upisuje samo relativna putanja (`file_path`). Time se odvajaju izvorni PHP skriptovi od sadržaja koji postavljaju korisnici.

5. Onemogućavanje izvršavanja skripti u upload folderu

Na produkcionim sistemima preporučuje se dodatna zaštita (npr. `.htaccess` ili podešavanje servera) koja sprečava izvršavanje PHP skripti iz foldera u koji korisnici mogu da upload-uju fajlove. Iako u okviru zadatka ovo nije detaljno podešavano, arhitektura sistema je osmišljena tako da se fajlovi tretiraju isključivo kao dokumenti za preuzimanje.

8.6. Obrada grešaka i zaštita internih informacija

Web aplikacije treba da budu oprezne prilikom prikaza grešaka korisnicima. Detaljni „stack trace“ ili poruke koje sadrže SQL upite mogu odati previše informacija o unutrašnjoj strukturi sistema.

U ovom projektu primenjen je sledeći princip:

- u produkcionom okruženju (Hostinger) detaljne PHP greške nisu prikazane krajnjem korisniku, već se, u slučaju ozbiljnije greške (npr. 500 Internal Server Error), prikazuje generička poruka,
- kritične situacije (npr. neuspešna konekcija sa bazom) hvataju se i prikazuje se kratka, korisniku razumljiva poruka, dok se detalji po potrebi mogu logovati.

Takođe, sistemi obično koriste:

- `try/catch` blokove oko kritičnih operacija sa bazom,
- onemogućavanje `display_errors` u produkciji, uz eventualno logovanje u fajl.

U kontekstu ovog projektnog zadatka, fokus je na tome da korisnik ne vidi sirove poruke PDO-a ili PHP-a, već samo informaciju da je došlo do problema i da pokuša kasnije.

8.7. Ograničenja i moguća unapređenja bezbednosti

Iako su implementirane osnovne mere zaštite, postoje i elementi koji bi se u ozbiljnoj produkcionoj verziji mogli i trebalo unaprediti:

1. Jača zaštita lozinki

Umesto ručnog hash('sha256', . . .), korišćenje funkcija `password_hash()` i `password_verify()` sa `bcrypt/argon2` algoritmima, uz automatsko dodavanje soli (salt) i opcije za menjanje „cost“ parametara.

2. CSRF zaštita (Cross-Site Request Forgery)

Trenutno, forme (npr. brisanje korisnika, dodavanje prijave, unos ocene) ne koriste CSRF tokene. U budućnosti bi svaka kritična forma trebalo da ima skriveni token povezan sa sesijom, kako bi se sprečilo da napadač natera korisnika da nenamerno izvrši neku akciju.

3. Ograničavanje pokušaja logovanja

Implementacija mehanizma koji posle određenog broja neuspelih pokušaja logovanja:

- privremeno blokira nalog, ili
- zahteva dodatnu proveru (npr. CAPTCHA),
bi značajno otežala brute-force napade.

4. Logovanje bezbednosno relevantnih događaja

Kreiranje log fajla ili posebne tabele za čuvanje:

- neuspelih pokušaja logovanja,
- brisanja važnih entiteta (npr. korisnika, predmeta),
- promene privilegija,
bi olakšalo detekciju potencijalno sumnjivih aktivnosti.

5. HTTPS i sigurne kolačiće

Na ozbiljnom produkcionom okruženju neophodno je koristiti HTTPS, kao i:

- `session.cookie_secure = 1,`
- `session.cookie_httponly = 1,`
kako bi se smanjio rizik od krađe kolačića i session hijacking napada.

6. Detaljnija kontrola ovlašćenja

Iako uloga (u loga) dobro razdvaja admin/profesor/student funkcionalnosti, u velikim sistemima se uvode granularnije dozvole (npr. razdvajanje sistemskog administratora, šefa katedre, studentske službe itd.). Model korisnika i prava se već nalazi u stanju koje omogućava takvo širenje.

Zaključak ovog poglavlja je da, iako je reč o studentskom projektu, u izradi sistema „**Zmaj University**“ primenjen je veliki deo standardnih bezbednosnih praksi koje se koriste u realnim web aplikacijama: heširanje lozinki, pripremljeni SQL upiti, striktna kontrola pristupa, validacija podataka i pažljivo rukovanje fajlovima. Ovo postavlja dobru osnovu za dalje unapređenje sistema ka još višem nivou bezbednosti.

9. Testiranje i scenariji korišćenja

Nakon što je implementacija osnovnih modula završena, sprovedeno je sistematsko **funkcionalno testiranje** aplikacije „Zmaj University“. Cilj testiranja bio je da se proveriti:

- da li sve ključne funkcionalnosti rade u skladu sa specifikacijom,
- da li su uloge (admin / profesor / student) pravilno razdvojene,
- da li su podaci u bazi konzistentni,
- da li se greške pravilno obrađuju i prikazuju korisniku.

Testiranje je pre svega rađeno kao **ručno testiranje** (manual testing), kroz realnu upotrebu sistema u browser-u, na pravom hosting okruženju (Hostinger), sa kreiranim test korisnicima i podacima.

9.1. Pristup testiranju

Primijenjen je sledeći pristup:

1. Postavljanje aplikacije na Hostinger

cela aplikacija je postavljena na poddomen `f.jovanhq.tech`, sa MySQL bazom napravljenom u Hostinger panelu.

time je testiranje rađeno u realnom okruženju, vrlo slično produkcijom.

2. Kreiranje test podataka

u bazi su SQL skriptama kreirani:

- jedan administratorski nalog,
- više profesora,
- više studenata,
- nekoliko predmeta,
- bar dva ispitna roka (npr. „January 2025“, „June 2025“),
- probne prijave i ocene.

3. Testiranje po ulogama

svaki modul (admin, professor, student) je posebno testiran:

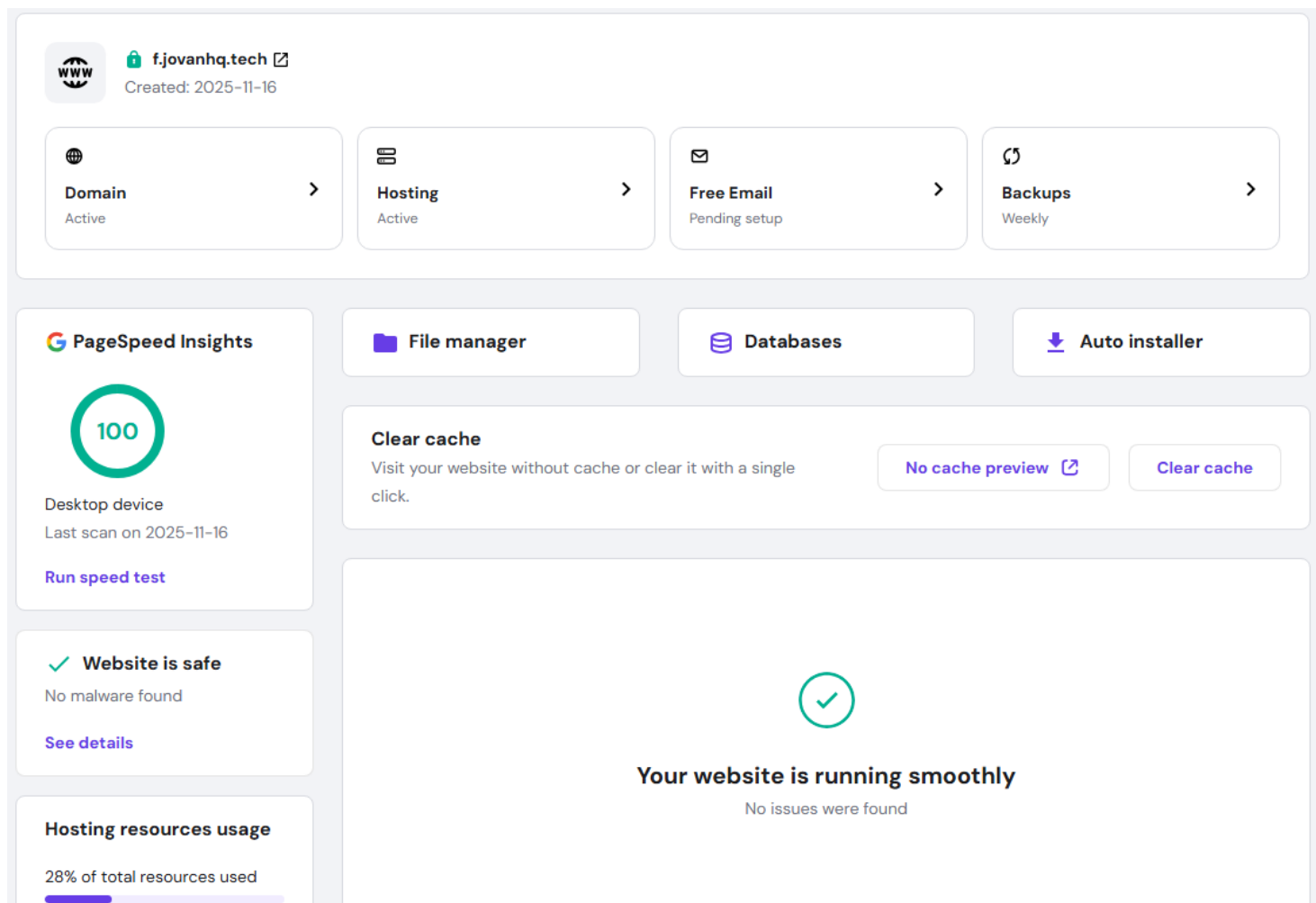
- prvo login i dashboard,
- zatim sve bitne funkcionalnosti (CRUD operacije, prijava ispita, ocenjivanje itd.).

4. Testiranje graničnih slučajeva

pogrešni logini, nevalidni podaci u formama, pokušaj pristupa tuđim stranicama preko direktnih URL-ova, itd.

5. Iterativno ispravljanje grešaka

sve uočene greške (500 greške, problemi sa foreign ključevima, pogrešni join-ovi u SQL upitima) postepeno su otklanjane, uz ponovno testiranje nakon svake izmene.



Slika 32: Primer okruženja - aplikacija pokrenuta na Hostinger poddomenu

9.2. Test podaci i test korisnici

Za testiranje sistema korišćeni su korisnici i podaci kreirani u bazi preko SQL skripti (u phpMyAdmin-u). To omogućava da se scenariji realno isprobaju bez potrebe za ručnim kreiranjem svakog korisnika.

Primer test korisnika:

- **Administrator**
 - e-mail: `admin@zmaj.univ`
 - lozinka: `admin123`
- **Profesor** (npr. Web programming)
 - e-mail: `milan.petrovic@zmaj.univ`
 - lozinka: `prof123`
- **Student** (npr. IT smer)
 - e-mail: `s001@zmaj.univ`
 - lozinka: `stud123`

Pored korisnika, u bazi su unapred uneti:

- predmeti (npr. „Web Programming 1“, „Databases“, „Computer Networks“),
- ispitni rokovi (npr. „January 2025“, „June 2025“),
- nekoliko veza između studenata i predmeta (`enrollments`),
- ispitne prijave i probne ocene.

Ovi test podaci korišćeni su u scenarijima u nastavku.

course_id	naziv	sifra	espb	opis	profesor_id	created_at	updated_at
1	Web Programming 1	WP1	6	Introduction to web technologies and HTML/CSS/JS.	2	2025-11-16 01:25:15	2025-11-16 01:25:15
2	Databases	DB1	6	Relational databases and SQL basics.	2	2025-11-16 01:25:15	2025-11-16 01:25:15
3	Algorithms and Data Structures	ALG1	6	Fundamental algorithms and data structures.	3	2025-11-16 01:25:15	2025-11-16 01:25:15
4	Computer Networks	NET1	6	Basics of computer networks and protocols.	4	2025-11-16 01:25:15	2025-11-16 01:25:15
5	Operating Systems	OS1	6	Concepts of modern operating systems.	3	2025-11-16 01:25:15	2025-11-16 01:25:15
6	Software Engineering	SE1	6	Software development lifecycle and methodologies.	2	2025-11-16 01:25:15	2025-11-16 01:25:15
8	Web Programming 2	WP2	6	Advanced web technologies: modern JS frameworks an...	2	2025-11-16 05:38:08	2025-11-16 05:38:08
9	Databases 2	DB2	6	Advanced SQL, normalization, indexing and transact...	34	2025-11-16 05:38:08	2025-11-16 05:38:08
10	Computer Architecture	CA1	6	Basic digital logic, CPU organization and memory h...	37	2025-11-16 05:38:08	2025-11-16 05:38:08
11	Computer Security	SEC1	6	Introduction to information security, cryptography...	38	2025-11-16 05:38:08	2025-11-16 05:38:08
12	Mobile Development	MOB1	6	Basics of mobile app development and mobile UI/UX.	39	2025-11-16 05:38:08	2025-11-16 05:38:08
13	Machine Learning Basics	ML1	6	Fundamentals of supervised and unsupervised learni...	36	2025-11-16 05:38:08	2025-11-16 05:38:08
14	Internet of Things	IOT1	6	Embedded devices, sensors and IoT communication pr...	35	2025-11-16 05:38:08	2025-11-16 05:38:08
15	Software Testing	TEST1	6	Unit testing, integration testing and test automat...	34	2025-11-16 05:38:08	2025-11-16 05:38:08
16	Project Management in IT	PM1	6	Planning, tracking and managing software projects.	40	2025-11-16 05:38:08	2025-11-16 05:38:08
17	English for IT	ENG1	4	Technical English for IT professionals.	39	2025-11-16 05:38:08	2025-11-16 05:38:08

Slika 33: Prikaz test predmeta u phpMyAdmin-u

9.3. Scenarijo 1 - Administrator upravlja korisnicima i predmetima

Cilj: Proveriti da li administrator može da kreira nove korisnike i predmete, izmeni i obriše postojeće podatke.

Koraci:

1. Administrator otvara početnu stranicu i loguje se sa svojim kredencijalima (admin@zmaj.univ / admin123).
2. Nakon prijave prikazuje se admin/dashboard.php sa sažetom statistikom.
3. Administrator otvara stranicu **Users** (admin/users.php).
4. Klikne na dugme „New user“ i otvara se forma za kreiranje korisnika.
5. Popunjava podatke za novog studenta:
 - ime, prezime, e-mail, lozinka, uloga = student, indeks, smer, godina.
6. Čuva korisnika - na stranici se pojavljuje zelena poruka o uspehu.
7. U tabeli korisnika proverava da li je novi student prikazan.
8. Administrator zatim otvara stranicu **Courses** (admin/courses.php).

9. Dodaje novi predmet, bira profesora iz liste i unosi ESPB i opis.

10. Predmet se pojavljuje u tabeli predmeta i može se kasnije izmeniti ili obrisati.

Očekivani rezultat:

- sistem prihvata validne podatke,
- odbija duplirani e-mail,
- za nove korisnike i predmete automatski se dodeljuju ID vrednosti,
- administrator može da vrši izmene bez SQL grešaka i bez narušavanja integriteta baze.

The screenshot shows a 'Create new user' form with a dark blue header. The form contains several input fields: 'First name' (with 'kkkkk' entered), 'Last name' (with 'kkkk' entered), 'E-mail' (with 'kkk' entered), 'Password' (empty), 'Role' (a dropdown menu showing '-- choose role --'), 'Index (for students)' (empty), 'Year of study' (empty), and 'Study program' (empty). At the bottom left, there are two buttons: 'Save' (blue) and 'Cancel' (grey).

Slika 34: Test - kreiranje novog studenta od strane administratora

The screenshot shows a 'Create new course' form with a dark blue header. The form contains several input fields: 'Course name' (empty), 'Code' (empty), 'ESPB' (with '6' entered), 'Professor' (a dropdown menu showing 'Jovanović Ana'), and 'Description' (empty). At the bottom left, there are two buttons: 'Save' (blue) and 'Cancel' (grey).

Slika 35: Test - kreiranje novog predmeta i dodela profesora

9.4. Scenarijo 2 - Profesor pregleda studente i unosi ocene

Cilj: Proveriti da li profesor može da:

- vidi svoje predmete i studente na njima,
- vidi prijave na ispit za određeni rok,
- unese i izmeni ocene.

Koraci:

1. Profesor se loguje na sistem (`milan.petrovic@zmaj.univ / prof123`).
2. Otvara se `professor/dashboard.php` sa pregledom njegovih predmeta i osnovnih statistika.
3. Profesor otvara stranicu **My courses** (`professor/courses.php`).
4. U tabeli vidi listu predmeta kojima je dodeljen.
5. Klikne na „View students“ za jedan predmet - otvara se `course_students.php`, gde vidi studente upisane na taj predmet.
6. Profesor zatim otvara stranicu **Exams & grades** (`professor/exams.php`).
7. Iz liste rokova bira aktivni rok (npr. „June 2025“).
8. Za dati rok i predmet klikne na „View students“ - otvara se `exam_students.php` sa spiskom prijavljenih studenata.
9. U padajućem meniju pored studenta izabere ocenu (npr. 9) i klikne na „Save“.
10. Stranica se osvežava, status prijave postaje „Passed“, a ocena i datum se prikazuju u tabeli.

Očekivani rezultat:

- profesor vidi samo **svoje** predmete i prijave (ne tuđe),
- unos ocene je moguć samo za aktivan rok i postojeću prijavu,
- u tabeli `grades` se kreira ili ažurira zapis za odabranog studenta i predmet,
- u tabeli `exam_applications` status prelazi sa `prijavljen` u `polozio` ili `pao`.

Students in DB1 - Databases

Exam period: **November & Whole 2026** :) (2025-11-01 – 2026-07-16)

Today: **2025-11-17** Grading is allowed

[← Back to exams](#)

#	Index	Name	Program	Application status	Grade	Date	Action
1	IT001	Marko Kostić	Internet Tehnologije	Passed	7	2025-11-16	7 ▼ Save

Slika 36: Test - pregled studenata prijavljenih na ispit i unos ocene

9.5. Scenarijo 3 - Student upisuje predmete i prati materijale

Cilj: Proveriti tok rada studenta tokom semestra - upis predmeta i pregled nastavnih materijala.

Koraci:

1. Student se loguje na sistem (s001@zmaj.univ / stud123).
2. Otvara se student/dashboard.php sa kratkim pregledom (broj predmeta, aktivnih rokova, prijava, položenih ispita).
3. Student otvara stranicu **Courses** (student/courses.php).
4. U tabeli vidi listu svih predmeta sa nazivima, šiframa, profesorima i ESPB-om.
5. Za predmet koji još nije upisan klikne na dugme „Apply / Enroll“ (ili slično) - kreira se zapis u enrollments.
6. Otvara stranicu **My courses** (student/my_courses.php) i proverava da li se taj predmet sada nalazi na listi.
7. Na istom ekranu ili preko dodatne stranice pregledava materijale (materials) vezane za jedan od predmeta (npr. listu PDF fajlova).

Očekivani rezultat:

- student ne može dva puta upisati isti predmet (UNIQUE ograničenje u enrollments),
- materijali su vidljivi samo za predmete na koje je student upisan (u realnoj implementaciji to se lako doda),

- linkovi ka materijalima omogućavaju preuzimanje fajlova.

Code	Name	ESPB	Students	Actions
DB1	Databases	6	2	View students
SE1	Software Engineering	6	2	View students
WP1	Web Programming 1	6	3	View students
WP2	Web Programming 2	6	1	View students

Slika 37: Test - pregled liste “My courses”

Course	Title	Description	Uploaded at	Actions
DB1 - Databases	yeah	/	2025-11-16 04:19:25	Download Edit Delete

Slika 38: Test - pregled materijala za odabrani predmet

9.6. Scenarijo 4 - Student prijavljuje ispit, profesor ga ocenjuje, student vidi ocenu

Ovo je najkompletniji scenario, koji povezuje sve tri uloge i više tabela u bazi.

Cilj: Proveriti da se ceo „životni ciklus“ ispita odvija ispravno - od prijave do konačne ocene.

Koraci:

1. **Administrator** proverava da je bar jedan ispitni rok aktivan (`exam_periods` sa `aktivan = 1` i odgovarajućim datumima).
2. **Student** se loguje i otvara `student/exams.php`.
 - bira aktivan ispitni rok iz padajuće liste,
 - vidi svoje upisane predmete i trenutni status (Not applied / Applied).
3. Student klikne na „Apply for exam“ za jedan od predmeta.

- u tabeli `exam_applications` nastaje novi zapis sa statusom prijavljen.
- na stranici status prelazi u „Applied“.

4. **Profesor** se loguje i otvara `professor/exams.php`.

- bira isti ispitni rok,
- u tabeli vidi broj prijavljenih studenata za svoje predmete.

5. Profesor otvara `exam_students.php` za isti predmet i rok.

- vidi studenta na listi, status `Registered`.

6. Profesor unosi ocenu (npr. 8) i klikne „Save“.

- u tabeli `grades` kreira se (ili ažurira) odgovarajući zapis,
- u `exam_applications` status prelazi u `polozio` (jer je ocena ≥ 6).

7. **Student** se ponovo loguje i otvara `student/grades.php`.

- u tabeli ocena sada vidi novi predmet sa ocenom 8 i datumom polaganja.

8. Student otvara ponovo `student/exams.php` za isti rok i predmet.

- status više nije „Applied“, već „Passed (8)“,
- nema mogućnost da otkaže prijavu ili ponovo prijavi isti ispit u istom roku.

Očekivani rezultat:

- svi podaci su konzistentni kroz `enrollments`, `exam_applications` i `grades`,
- nema dupliranih prijava,
- ocene su vidljive studentu odmah nakon unosa,
- statusi (badge-ovi) na studentskim i profesorskim stranicama su usklađeni.

Exam registration					
Exam period		Search course			
November & Whole 2026 :) (2025-11-01 – 21-11-2025) ▾		By name or code		Filter	
Code	Course	ESPB	Professor	Status	Action
ALG1	Algorithms and Data Structures	6	Ana Jovanović	Not applied	Apply for exam
DB1	Databases	6	Milan Petrović	Passed	(Cannot change after grading)
NET1	Computer Networks	6	Nikola Ilić	Registered	Cancel application
OS1	Operating Systems	6	Ana Jovanović	Not applied	Apply for exam

Slika 39: Test - student prijavljen na ispit

My grades						
Code	Course	ESPB	Grade	Status	Exam period	Date
DB1	Databases	6	7	Passed	November & Whole 2026 :) (2025-11-01 – 21-11-2025)	2025-11-16

Slika 40: Test - student vidi novu ocenu nakon unosa od strane profesora

9.7. Testiranje grešaka i graničnih slučajeva

Pored „pozitivnih“ scenarija, testirani su i **negativni** scenariji i granični slučajevi:

1. Pogrešan login

- unos nepostojeće e-mail adrese ili pogrešne lozinke vodi do poruke o grešci,
- ne kreira se sesija, nema pristupa internim stranicama.

2. Pristup bez autorizacije

- pokušaj da se direktno otvori npr. `admin/users.php` bez logovanja, ili kao ulogovani student,
- rezultat: preusmeravanje na `index.php`.
- potvrđuje da provere uloge rade ispravno.

3. Nevalidni parametri u URL-u

- ručno menjanje `id` parametara (npr. `course_id` na vrednost koja ne pripada profesoru),
- sistem proverava vlasništvo (da li profesor zaista drži taj predmet);
- u slučaju neuspeha, vrši se preusmeravanje nazad na listu.

4. Duplirani unos

- student pokušava da se dvaput prijavi na isti ispit u istom roku,
- upit prvo proverava postojanje prijave; duplikat se ne kreira.

5. Invalidne ocene

- ručno slanje vrednosti ocene van opsega 5-10,
- PHP validacija odbija unos i prikazuje poruku.

6. Greške u bazi i integritet

- tokom razvoja i testiranja uočene su greške vezane za foreign key ograničenja (npr. pokušaj upisa ocene bez validne prijave),
- model tabele `grades` usklađen je tako da sadrži `application_id` i odgovarajuće ključne veze, čime je obezbeđen integritet.



Zmaj University

Faculty Information System

Invalid email or password.

E-mail

admin@zmaj.univ

Password

Enter your password

Sign in

Test accounts – click the email or password to copy:

ADMIN

admin@zmaj.univ

/

admin123

PROFESSOR

milan.petrovic@zmaj.univ

/

prof123

STUDENT

s001@zmaj.univ

/

stud123

Click any credential above to copy it to clipboard.

**Open project documentation**
Seminar paper & system documentation · opens in a new tab

© 2025 jovanhq.tech

Slika 41: Primer poruke o grešci pri nevažećem logovanju

9.8. Rezultati testiranja

Nakon završetka iterativnog testiranja može se zaključiti:

- sve glavne funkcionalnosti definisane u zahtevima rade u skladu sa očekivanjima,
- login i kontrola pristupa po ulozi funkcionišu korektno,
- administratori mogu da upravljaju korisnicima, predmetima, rokovima i obaveštenjima,
- profesori mogu da vide svoje predmete, studente, prijave na ispit i da unose ocene,
- studenti mogu da upisuju predmete, prijavljuju ispite, prate materijale i pregledaju ocene,
- eventualne greške tokom razvoja (npr. SQL greške, 500 greške) su otkrivene kroz testiranje na stvarnom serveru i sukcesivno otklonjene.

Time je potvrđeno da je sistem „**Zmaj University**“ stabilan za demonstraciju i korišćenje u okviru projektnog zadatka, kao i da ispunjava ciljeve postavljene u uvodnim poglavljima.

10. Moguća unapređenja sistema

Iako sistem „Zmaj University“ u trenutnoj verziji ispunjava sve definisane funkcionalne zahteve i omogućava osnovan rad studentske službe, profesora i studenata, postoje brojna unapređenja koja bi dodatno podigla kvalitet, skalabilnost i bezbednost rešenja. U nastavku su navedene najvažnije ideje podeljene u nekoliko celina.

10.1. Funkcionalna proširenja

10.1.1. Kurs-specifična obaveštenja

Trenutno tabela `announcements` čuva globalna obaveštenja, vidljiva svim korisnicima. Logičan sledeći korak jeste uvođenje:

- dodatnog polja `course_id` u tabelu `announcements`,
- logike da profesor može da objavi obaveštenje vezano za konkretan predmet,
- filtriranja obaveštenja na studentskoj strani - student vidi:
 - globalna obaveštenja (administracija),
 - obaveštenja za predmete na koje je upisan.

Na ovaj način studenti bi imali centralno mesto sa svim relevantnim informacijama, bez mešanja obaveštenja koja se na njih ne odnose.

10.1.2. Raspored časova i konsultacija

U okviru predmeta trenutno se čuvaju osnovne informacije (naziv, šifra, profesor, ESPB). Kao proširenje može se dodati:

- polje `termin_predavanja` - vreme i dan predavanja,
- polje `termin_vezbi` - termin vežbi, ako postoje,
- polje `termin_konsultacija` - kada profesor drži konsultacije.

Studentski i profesorski modul mogli bi da prikazuju pregled rasporeda, a sistem bi mogao da generiše kalendar (npr. nedeljni raspored). To bi značajno unapredilo praktičnu upotrebu sistema.

10.1.3. Više izlazaka na ispit i istorija ocena

Trenutni model `grades` je pojednostavljen i vezan je za jednu prijavu po predmetu. U realnim uslovima student može više puta izlaziti na ispit, pa su moguća proširenja:

- čuvanje *istorije* svih izlazaka i ocena (npr. jedna prijava po izlasku),
- označavanje koja ocena je konačna (npr. najviša ili poslednja),
- pregled istorije polaganja po predmetu u studentskom profilu.

Ovo bi omogućilo detaljniju analizu uspeha studenata i statistike po predmetima i rokovima.

10.1.4. Evidencija prisustva i vežbi

Dodatna funkcionalnost koja bi bila korisna u nastavi:

- evidencija prisustva na predavanjima i vežbama,
- evidencija bodova sa kolokvijuma, domaćih zadataka i projekata,
- automatski izračun „prelaznih“ bodova tokom semestra i uslova izlaska na ispit.

To podrazumeva uvođenje novih tabela (npr. `attendance`, `assignments`, `assignment_grades`) i odgovarajućih ekrana u profesorskom i studentskom modulu.

10.2. Tehnička poboljšanja

10.2.1. Naprednije heširanje lozinki

U narednoj verziji sistema preporučuje se prelazak sa manuelnog `hash( 'sha256', ... )` na funkcije:

- `password_hash( )` - za generisanje heš lozinke,
- `password_verify( )` - za proveru.

Ove funkcije automatski dodaju **salt**, koriste moderne algoritme (`bcrypt`, `Argon2`), i jednostavnije je menjati parametre jačine (`cost`). Time bi se bezbednost autentifikacije podigla na nivo koji je u skladu sa industrijskim standardima.

10.2.2. CSRF zaštita za forme

Trenutno forme za kritične operacije (brisanje korisnika, prijava/odjava ispita, unos ocena) nemaju CSRF tokene. U budućnosti bi se moglo:

- u sesiju čuvati nasumični token (`$_SESSION['csrf_token']`),
- u svaku formu ubaciti skriveno polje `<input type="hidden" name="csrf_token" ...>`,
- pri obradi POST zahteva proveravati da li se token poklapa.

Na taj način sprečavaju se Cross-Site Request Forgery napadi, gde napadač pokušava da navede žrtvu da nenamerno pošalje validan POST zahtev.

10.2.3. API sloj i SPA klijent

Trenutno je aplikacija klasičan „server-rendered“ PHP sistem (HTML generiše PHP). Kao sledeći korak, može se uvesti:

- REST API sloj (npr. `/api/courses`, `/api/students`, `/api/exams`),
- frontend napisan u modernom JavaScript okviru (React, Vue, Angular),
- Single Page Application (SPA) pristup za brže i interaktivnije korisničko iskustvo.

Backend (PHP + MySQL) tada bi funkcionisao kao servis koji vraća JSON podatke, dok bi većina UI logike bila na klijentskoj strani.

10.2.4. Paginacija i napredna pretraga

Kod velikog broja korisnika, predmeta i prijava, liste bi postale predugačke. Kao unapređenje može se dodati:

- paginacija (npr. po 20 zapisa po strani),
- napredniji filteri (po smeru, godini, ESPB, profesoru, roku),
- sortiranje po koloni klikom na zaglavlje tabele.

Ovo bi značajno poboljšalo performanse i korisničko iskustvo na većim dataset-ovima.

10.3. Unapređenje korisničkog interfejsa

Iako trenutni dizajn već koristi Bootstrap 5 i dodatni `custom.css` sa tamnom temom i brendiranjem „Zmaj University“, postoje brojna potencijalna unapređenja:

- **odgovarajući (responsive) dizajn za mobilne uređaje** - dodatna testiranja i prilagođavanje kako bi sve tabele i forme lepo izgledale na telefonu,
- **ikonice i vizuelni elementi** - npr. Font Awesome ili Bootstrap Icons za jasnije razlikovanje akcija (edit, delete, view),
- **grafikoni na dashboard-u** - prikaz broja studenata po smerovima, prolaznosti po predmetima, broja prijava po rokovima (npr. kroz biblioteku poput Chart.js),
- **tamni/svetli režim** - mogućnost da korisnik sam bira temu, što je savremeni standard u aplikacijama.

Sve ovo ne menja suštinsku logiku sistema, ali značajno utiče na utisak koji projekat ostavlja na korisnike i, u ovom slučaju, na komisiju/ profesora.

10.4. Integracije sa drugim sistemima

U realnom univerzitetskom okruženju, informacioni sistem redovno komunicira sa drugim servisima. Moguća buduća unapređenja uključuju:

- integraciju sa **e-mail servisom** radi automatskog slanja obaveštenja (npr. potvrda prijave ispita, obaveštenje o izmenjenom terminu),
- integraciju sa **LMS sistemima** (Moodle, Google Classroom) za prenos bodova, zadataka i materijala,
- eksport **PDF izveštaja** - generisanje uverenja o položenim ispitima, transkripta ocena, spiskova studenata po predmetima ili rokovima,
- integraciju sa **mobilnim aplikacijama** - npr. Android/iOS klijent koji koristi isti API sloj.

10.5. DevOps, backup i monitoring

Na nivou infrastrukture i održavanja moguće su dodatne mere:

- automatizovani backup baze podataka (dnevni/nedeljni),

- logovanje ključnih događaja (prijave, brisanja, promene privilegija),
- monitoring performansi i iskorišćenosti resursa,
- kontinuirana integracija i isporuka (CI/CD) - npr. automatsko prebacivanje nove verzije na server uz osnovne testove.

Iako ovakvi mehanizmi prevazilaze obim tipičnog studentskog projekta, njihovo planiranje pokazuje da je sistem osmišljen tako da može rasti u ozbiljan produkcionni servis.

11. Zaključak

Cilj projekta „Zmaj University“ bio je da se, polazeći od zadate projektne dokumentacije i zahteva, razvije funkcionalni informacioni sistem za upravljanje nastavnim procesom na visokoškolskoj ustanovi. Kroz rad na projektu obuhvaćeni su svi ključni koraci tipičnog životnog ciklusa softvera:

- analiza zahteva i identifikacija uloga (administrator, profesor, student),
- projektovanje baze podataka i definisanje relacija,
- projektovanje arhitekture aplikacije (trodeleni modul: admin / professor / student),
- implementacija backend logike u PHP-u i MySQL-u,
- dizajn korisničkog interfejsa uz pomoć Bootstrap-a,
- testiranje funkcionalnosti u realnom hosting okruženju,
- analiza mogućih unapređenja i bezbednosnih aspekata.

Implementirani sistem omogućava:

- administratorima da upravljaju korisnicima, predmetima, ispitnim rokovima i obaveštenjima,
- profesorima da pregledaju studente po predmetima, vide ispitne prijave i unose ocene,
- studentima da se prijave na predmete, prijave ispite u aktivnim rokovima, prate materijale i gledaju svoje ocene.

Kroz razvoj projekta primećeno je koliko je bitno:

- da model baze podataka bude dobro osmišljen od početka (zbog integriteta i jednostavnije logike),
- da se koristi dosledna kontrola pristupa po ulogama,
- da se koristi heširanje lozinki i pripremljeni SQL upiti,
- da se svaki problem otkriven testiranjem pravovremeno ispravi, umesto da se „zaobiđe“.

Pored tehničkog znanja (PHP, MySQL, HTML, CSS, Bootstrap), projekat je doprineo i razumevanju šire slike - kako izgleda realan informacioni sistem fakulteta iz perspektive programera: od definisanja tabela i veza, preko razmišljanja o korisničkom interfejsu, do bezbednosti i mogućih integracija sa drugim servisima.

Može se zaključiti da je postavljeni cilj ostvaren:

- razvijen je kompletan web sistem koji pokriva definisane funkcionalne zahteve,
- sistem je uspešno testiran na realnom serveru (Hostinger) i spreman je za demonstraciju,
- dokumentacija (seminarski rad) detaljno opisuje arhitekturu, implementaciju, bezbednost i potencijalna unapređenja.

Ovaj projekat može poslužiti kao osnova za dalji rad - bilo kao osnovu za ozbiljniji fakultetski informacioni sistem, bilo kao polazna tačka za nove projekte iz oblasti web programiranja i baza podataka.

12. Literatura

Tokom izrade projekta i dokumentacije korišćeni su sledeći izvori:

1. **PHP Manual - PHP Documentation**

PHP Group

Dostupno na: <https://www.php.net/manual/>

2. **MySQL 8.0 Reference Manual - Data Definition and SQL Language**

Oracle Corporation

Dostupno na: <https://dev.mysql.com/doc/>

3. **Bootstrap 5 - Official Documentation**

Bootstrap Authors

Dostupno na: <https://getbootstrap.com/docs/5.3/>

4. **MDN Web Docs - HTML, CSS, JavaScript Guides**

Mozilla Foundation

Dostupno na: <https://developer.mozilla.org/>

5. Materijali sa predmeta „**Serverske tehnologije**“ i/ili drugih predmeta na studijskom programu, prezentacije i skripte dobijene od predmetnog profesora.

6. Službena dokumentacija i tutorijali hosting provajdera **Hostinger** (PHP, MySQL, File Manager, podešavanje domena i poddomena).