

# Projekat: Web aplikacija za visokoškolsku ustanovu

## 1. Cilj projekta

Zadatak ovog projekta je da student, samostalno ili u grupi, razvije **web aplikaciju (dinamički sajt)** za imaginarni fakultet, akademiju ili visoku školu.

Aplikacija treba da simulira **realan informacijski sistem obrazovne ustanove** – poput internog portala gde studenti, profesori i administratori mogu da obavljaju osnovne radnje vezane za predmete i ispite.

Svrha projekta je da studenti primene znanja iz serverskih tehnologija i povežu:

- rad sa **baza podataka** (kreiranje tabela, relacija, primarnih i stranih ključeva),
- **backend programiranje** (Node.js, PHP, Laravel ili .NET Web Forms),
- i **frontend tehnologije** (HTML, CSS, JavaScript).

Aplikacija treba da podržava tri uloge korisnika:

- **Administrator (Admin)** – upravlja sistemom, korisnicima i rokovima,
- **Profesor** – postavlja nastavne materijale i unosi ocene,
- **Student** – prijavljuje ispite, pregleda ocene i materijale.

Svi podaci mogu biti **izmišljeni ili generisani uz pomoć AI alata** (npr. ChatGPT, Mockaroo, DataGenerator).

Bitno je da sistem sadrži **oko 10 profesora, 15–20 predmeta i 50–100 studenata**, kako bi baza bila dovoljno velika da pokaže strukturu realnog sistema.

## 2. Dozvoljene tehnologije i okruženje

Studenti sami biraju tehnologiju u kojoj će razvijati svoj projekat.

Svi projekti moraju imati **server–kljent arhitekturu**, gde server komunicira sa bazom podataka, a korisniku vraća dinamički sadržaj.

### Dozvoljene tehnologije:

Tehnologija	Okruženje / alat	Baza podataka
PHP	XAMPP / WAMP / Lagon	MySQL
Laravel (PHP framework)	Composer, Artisan CLI	MySQL
Node.js (Express.js)	VS Code, npm, nodemon	MySQL / MongoDB
.NET Web Forms / ASP.NET Core MVC	Visual Studio	MSSQL Server
Frontend	HTML, CSS, JavaScript	—

Možete koristiti i **Bootstrap** ili **Tailwind CSS** kako bi dizajn bio responzivan i moderan. Važno je da aplikacija ne bude samo statična, već da korisnik može da **dodaje, menja, briše i pretražuje podatke** (tzv. CRUD operacije).

### 3. Struktura baze podataka

Osnova svake aplikacije su podaci, zato je potrebno **kreirati relacione tabele** u bazi podataka. Tabele treba da budu povezane primarnim (PK) i stranim ključevima (FK), tako da sistem funkcioniše kao prava fakultetska baza.

#### Predložene tabele i veze:

##### 1. USERS

Ova tabela sadrži sve korisnike sistema, administratore, profesore i studente. Svi se čuvaju u jednoj tabeli jer imaju slične atribute, a razlikuju se po ulozi (uloga).

- **Ključni atributi:**
  - user\_id (primarni ključ)
  - ime
  - prezime
  - adresa
  - telefon
  - email
  - lozinka
  - uloga (admin / profesor / student)
  - indeks, smer, godina (*popunjava se samo za studente*)
- **Veze:**
  - 1–n veza sa tabelom courses (kao profesor)
  - 1–n sa grades, announcements i exam\_applications
  - koristi se kao referenca u većini ostalih tabela putem user\_id

##### 2. COURSES

Tabela courses sadrži sve predmete koji se predaju na fakultetu. Svaki predmet ima naziv, šifru, ESPB vrednost, opis i profesora koji ga predaje.

- **Ključni atributi:**
  - course\_id (primarni ključ)
  - naziv
  - šifra
  - ESPB
  - opis
  - profesor\_id (FK → users.user\_id)

- **Veze:**
  - 1–n veza sa materials (svaki predmet ima više materijala)
  - 1–n sa grades (više studenata može imati ocene iz jednog predmeta)
  - 1–n sa exam\_applications (studenti prijavljuju ispite iz predmeta)
  - 1–n sa enrollments (više studenata može biti upisano na predmet)

### 3. ENROLLMENTS

Tabela enrollments povezuje studente i predmete koje su upisali.

Ovo je *spojna tabela* koja ostvaruje relaciju “više prema više” između korisnika i predmeta.

- **Ključni atributi:**
  - enrollment\_id (primarni ključ)
  - student\_id (FK → users.user\_id)
  - course\_id (FK → courses.course\_id)
- **Veze:**
  - m–n relacija između users (studenti) i courses (predmeti)
  - koristi se za praćenje koji studenti pohađaju koji predmet

### 4. EXAM\_PERIODS

Tabela u kojoj se čuvaju svi ispitni rokovi. Svaki rok ima naziv i datume trajanja.

- **Ključni atributi:**
  - period\_id (primarni ključ)
  - naziv
  - datum\_od
  - datum\_do
- **Veze:** 1–n veza sa exam\_applications (više prijava može pripadati jednom roku)

### 5. EXAM\_APPLICATIONS

Sadrži prijave koje su studenti podneli za ispite u određenim rokovima.

Ova tabela povezuje tri entiteta — studenta, predmet i rok.

- **Ključni atributi:**
  - application\_id (primarni ključ)
  - student\_id (FK → users.user\_id)
  - course\_id (FK → courses.course\_id)
  - period\_id (FK → exam\_periods.period\_id)
  - status (npr. prijavljen, položio, nije položio)
- **Veze:**
  - povezuje studente sa predmetima i rokovima
  - koristi se kasnije kao osnov za unos ocene u grades

## 6. GRADES

Evidencija ocena koje profesori unose nakon ispita.

Svaka ocena pripada tačno jednom studentu i jednom predmetu.

- **Ključni atributi:**
  - grade\_id (primarni ključ)
  - student\_id (FK → users.user\_id)
  - course\_id (FK → courses.course\_id)
  - ocena
  - datum
- **Veze:**
  - 1–1 veza sa exam\_applications (svaka prijava može imati najviše jednu ocenu)
  - omogućava pregled položenih ispita i statistiku uspešnosti

## 7. MATERIALS

Tabela u kojoj se čuvaju svi nastavni materijali koje profesori postavljaju za svoje predmete.

- **Ključni atributi:**
  - material\_id (primarni ključ)
  - course\_id (FK → courses.course\_id)
  - naslov
  - opis
  - fajl\_putanja (lokacija fajla na serveru)
- **Veze:**
  - 1–n sa courses (svaki predmet može imati više materijala)

## 8. ANNOUNCEMENTS

Sadrži obaveštenja koja mogu postavljati profesori ili administratori.

Studenti ta obaveštenja mogu videti na svom panelu.

- **Ključni atributi:**
  - announcement\_id (primarni ključ)
  - naslov
  - tekst
  - datum
  - autor\_id (FK → users.user\_id)
- **Veze:**
  - 1–n veza sa users (jedan korisnik može objaviti više obaveštenja)

💡 *Napomena:*

- Svi strani ključevi moraju biti jasno definisani u SQL-u.
- Kod kreiranja tabela obavezno koristiti ON DELETE CASCADE gde ima smisla (npr. ako obrišemo predmet, brišu se i materijali).
- Preporučuje se da dodate i **polja za vreme unosa** (created\_at, updated\_at) radi lakšeg praćenja podataka.

#### 4. Funkcionalnosti sajta (po ulogama)

Sajt mora imati **početnu stranu za login**, a zatim tri različita panela (dashboarda) – za admina, profesora i studenta.

Nakon prijave, sistem proverava korisnikovu ulogu (role) i otvara odgovarajući panel.

##### ◆ ADMIN PANEL

Administrator ima puni pristup sistemu i kontroliše sve korisnike, predmete i rokove.

##### Operacije:

###### 1. Upravljanje korisnicima

- Pregled svih korisnika u tabeli (users).
- Dodavanje novih profesora i studenata.
- Brisanje korisnika.
- Izmena podataka (ime, email, uloga).

###### 2. Upravljanje predmetima

- Dodavanje novog predmeta sa nazivom, šifrom i ESPB.
- Dodeljivanje profesora predmetu.
- Izmena i brisanje predmeta.

###### 3. Upravljanje ispitnim rokovima

- Kreiranje rokova (npr. Januar, April, Jun, Septembar, Oktobar).
- Unos datuma početka i kraja.
- Aktivacija i deaktivacija roka.

###### 4. Obaveštenja

- Pisanje i objavljivanje važnih informacija (npr. “Ispitni rok počinje 12.01.”).
- Pregled starih obaveštenja i njihovo brisanje.

###### 5. Statistika

- Pregled broja korisnika, predmeta, studenata i prijavljenih ispita.
- Prikaz preko grafikona ili tabela.

## ◆ PROFESOR PANEL

Profesor koristi svoj deo sajta za rad sa studentima i materijalima predmeta.

### Operacije:

#### 1. Moji predmeti

- Lista predmeta koje profesor predaje.
- Pregled studenata koji su upisani na svaki predmet.
- Ažuriranje termina predavanja i konsultacija.

#### 2. Materijali

- Postavljanje novih nastavnih datoteka (PDF, PPT, DOC).
- Brisanje ili izmena postojećih materijala.
- Fajlovi se fizički čuvaju u server folderu, a link u bazi.

#### 3. Prijavljeni studenti

- Pregled svih studenata prijavljenih za ispit.
- Mogućnost sortiranja i filtriranja.

#### 4. Unos ocena

- Unos ocena u aktivnim rokovima (validacija datuma).
- Čuvanje ocena u tabeli grades.
- Ažuriranje statusa u exam\_applications.

#### 5. Obaveštenja studentima

- Slanje poruka studentima vezanih za predmet (npr. "Novi test je dostupan.").
- Brisanje zastarelih obaveštenja.

## ◆ STUDENT PANEL

Student vidi svoj profil, upisane predmete, materijale i ispite.

### Operacije:

#### 1. Moj profil

- Pregled osnovnih informacija (ime, indeks, email).
- Izmjena kontakt podataka i profilne slike.

#### 2. Predmeti

- Pregled svih predmeta i profesora.
- Klikom na predmet – prikaz opisa i dostupnih materijala.

#### 3. Upis na predmet

- Dodavanje predmeta u svoj plan (insert u tabelu enrollments).
- Sistem proverava da li je student već upisan.

#### 4. Prijava ispita

- Pregled aktivnih rokova.
- Prijava ispita za predmet u okviru roka.
- Status prijave (“prijavljen”, “položio”).

#### 5. Ocene i materijali

- Pregled položenih ispita i ocena.
- Preuzimanje nastavnih fajlova.

**Sprečiti upisivanje ocena ukoliko današnji datum ne pripada nijednom ispitnom roku!**

---

### 5. Vizuelni identitet i dizajn

#### Paleta boja i UI smernice

- Definirati paletu od 4–5 boja pomoću [Coolers.co](https://coolers.co).
- Koristiti dosledan dizajn na svim stranicama (header, footer, sidebar, tabele).
- Navigacija: horizontalni meni (navbar) ili sidebar sa ikonama i tekstom.

#### Logotipi i brendiranje

1. Glavni logotip (baner) – ime fakulteta + simbol, prikazan u zaglavlju sajta.
2. Sekundarni logotip (minimalni) – inicijal fakulteta (npr. “V”, “A”, “U”), prikazan u uglu sajta.
3. Tamna verzija – ako tim implementira *dark mode* bonus 10 poena

#### Alati: Canva, Figma, Photopea, Photoshop, AI generisani logo dizajni i sl.

#### Slike i ikonice

- Ikonice preuzeti sa [Flaticon.com](https://flaticon.com).
- Slike i ilustracije – [Pexels](https://pexels.com), [Pixabay](https://pixabay.com), ili AI generisane.
- Sve slike treba da budu optimizovane za web (manje od 500 KB).

Ako želite da budete dodatno kreativni, možete koristiti druge boje (na primer Admin prati crvene nijanse, Student zelene, Profesor plave), na primer generišete više paleta na [coolers.co](https://coolers.co), ali stil i struktura mora da ostanu isti i da sve to izgleda uniformno.

Pogledajte ove korisne linkove za optimizaciju slika, na časovima ćemo pričati o konceptima Green Web-a svakako:

<https://www.globaldizajn.net/optimizacija-slika-za-web-kompletan-vodic/>

<https://seoptimizacijijasajta.com/kako-optimizovati-slike-za-web/>

## 6. Navigacija i izgled sajta

Sajt treba da bude **pregledan, dosledan i funkcionalan**, sa jasno razdvojenim delovima.

### Preporučeni raspored:

- **Header:** naziv fakulteta i logo.
- **Sidebar ili gornji meni (navbar):** pet glavnih stavki po ulozi.
- **Footer:** autori projekta, godina i predmet.
- **Centralni deo stranice:** dinamički sadržaj (tabele, forme, liste, obaveštenja).

### Predlozi za menije:

#### Admin:

1. 👤 Korisnici
2. 📖 Predmeti
3. 📅 Rokovi
4. 📣 Obaveštenja
5. 📊 Statistika

#### Profesor:

1. 📖 Moji predmeti
2. 📄 Materijali
3. 👤 Studenti
4. 📝 Ocene
5. 📣 Obaveštenja

#### Student:

1. 👤 Profil
2. 📖 Predmeti
3. 📝 Upis
4. 📅 Ispiti
5. 📊 Ocene

#### 💡 Napomena:

Layout može biti horizontalni (navbar na vrhu) ili vertikalni (sidebar levo), ali svi ekrani moraju imati **dosledan dizajn, iste boje i fontove**.

Logo fakulteta može se prikazivati u gornjem desnom uglu kao **sekundarni logotip**.

## 7. Napomena za ocenjivanje

Kriterijum	Max poena	Opis
<b>Funkcionalnost</b>	25	Sve stranice rade, login i CRUD operacije funkcionalne
<b>Struktura baze</b>	15	Tabele povezane FK vezama, konzistentni podaci
<b>Dizajn i preglednost</b>	20	Jasna navigacija, konzistentne boje i logo
<b>Serverska logika</b>	15	Ispravno korišćenje PHP/Node/.NET za prikaz i unos podataka
<b>Kreativnost i realizacija</b>	5	Originalnost, dodatne opcije, timski rad, itd.