Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektronsko poslovanje

Laravel

Domaći zadatak 1

|  |  |  |
| --- | --- | --- |
| Ime | Prezime | Broj indeksa |
| Jovan | Vukovic | 2020/0122 |
| Ognjen | Gajic | 2020/0050 |
| Link ka Github-u |  | |

Sadržaj

[1 Korisnički zahtev 3](#_Toc184727824)

[1.1 Slučajevi korišćenja 3](#_Toc184727825)

[2 Implementacija backenda (laravel) 5](#_Toc184727826)

[2.1 Kratak uvod u laravel 5](#_Toc184727827)

[2.2 Model Podataka 6](#_Toc184727828)

[2.3 Migracije 6](#_Toc184727829)

[2.4 Kontroleri 7](#_Toc184727830)

[2.5 Lista api ruta 8](#_Toc184727831)

[2.6 Korisničko uputstvo (backend) 38](#_Toc184727832)

[3 Implementacija frontenda(React) 42](#_Toc184727833)

[3.1 Kratak uvod u React 42](#_Toc184727834)

[3.2 Karakteristicni delovi koda 42](#_Toc184727835)

[3.3 Korisničko uputstvo 57](#_Toc184727836)

[4 Opis arhitekture aplikacije 62](#_Toc184727837)

# Korisnički zahtev

Aplikacija za online ucenje je interaktivna aplikacija razvijena u **React-u**. koja omogućava korisnicima da se registruju, prijave, pregledaju različite kurseve i upravljaju svojim korpom proizvoda. Za stilizaciju komponenti korišćen je **Mini.css**, zbog jednostavnosti i brzine njegove implementacije u celokupan projekat.

Osim osnovnih funkcionalnosti, aplikacija integriše **PayPal API**, omogućavajući sigurno i jednostavno plaćanje prilikom kupovine kurseva.

Aplikacija takođe sadrži funkciju za **filtriranje kurseva**, što korisnicima omogućava da lako pretražuju i pronalaze kurseve prema njegovom nazivu. Ovo olakšava korisnicima da pronađu sadržaje koji najbolje odgovaraju njihovim potrebama i interesovanjima.

Aplikacija koristi više **reusable komponenti**, poput **Navbar**, što poboljšava organizaciju koda i omogućava lako održavanje i nadogradnju aplikacije. Ove komponente su dizajnirane da budu fleksibilne i prilagodljive, što dodatno unapređuje korisničko iskustvo.

**Rute** su pažljivo postavljene kako bi se omogućila jednostavna navigacija kroz sve delove sajta. Korisnici mogu lako pristupiti različitim stranicama aplikacije, uključujući stranice za pregled kurseva, registraciju, prijavu, korpu i plaćanje.

## Slučajevi korišćenja

1. **Registracija novog korisnika:**

* Korisnik popunjava formu sa osnovnim informacijama.
* Sistem vrsi validaciju podataka.
* Ako su podaci validni korisnik se registruje.

1. **Logovanje korisnika:**

* Korisnik unosi podatke za prijavu.
* Sistem vrši validaciju podataka.
* Ako su podaci tačni, korisnik se uspešno prijavljuje na sistem.
* Korisniku se prikazuje početna stranica sa relevantnim informacijama.

1. **Prikaz profila korisnika:**

* Korisnik može pregledati svoj profil, uključujući informacije poput imena, email adrese...

1. **Kupovina i pregled kurseva:**

* Korisniku je omogućena kupovina kurseva i različite opcije plaćanja (Paypal,Visa,Uplatnica...).
* Korisnik može pregledati listu kurseva koje je kupio.

**Slučajevi korišćenja Administratora:**

1. **Logovanje Administratora:**

* Administrator unosi podatke za prijavu.
* Sistem vrši validaciju podataka.
* Administratoru se pokazuje personalizovana početna stranica sa relevantnim informacijama.

1. **Dodavanje novih predavača:**

* Administrator unosi osnovne informacije o novom predavaču u predviđena polja (ime, prezime, korisničko ime, e-mail adresa).
* Sistem vrši validaciju unetih podataka kako bi osigurao tačnost informacija.

1. **Upravljanje predavačima:**

* Administrator može pregledati listu predavaca na platformi.
* Može ukloniti predavača ili mu dodeliti odredjene privilegije
* Administrator moze izmeniti informacije o predavaču

1. **Upravljanje korisnicima:**

* Administrator može pregledati listu korisnika na platformi.
* Može ukloniti korisnika ili mu dodeliti određene privilegije.
* Administrator može izmeniti informacije o korisniku

**Slučajevi korišćenja Predavača:**

1. **Logovanje Predavača:**

* Predavač unosi podatke za prijavu.
* Sistem vrši validaciju podataka.
* Predavaču se pokazuje personalizovana početna stranica sa relevantnim informacijama.

1. **Dodavanje novog kursa:**

* Predavač popunjava formu sa naslovom, opisom, cenom i drugim detaljima kursa.
* Podaci za novi kurs prolaze validaciju.
* Ako su podaci validni, kurs se kreira.

1. **Dodavanje nove lekcije:**

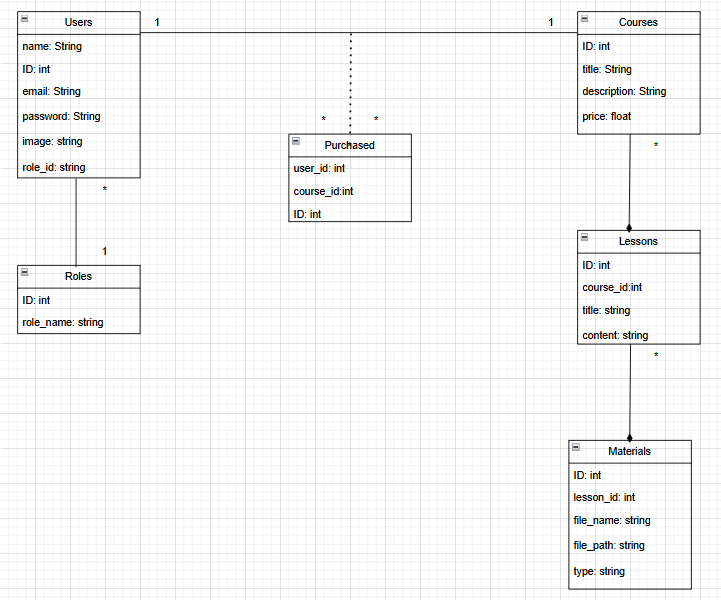
* Predavac odabira kurs kojem će dodati lekciju, Popunjava neophodne informacije, uključujući naslov, opis i link ka video materijalu.
* Podaci za novu lekciju prolaze kroz validaciju.
* Ako su podaci validni, lekcija se kreira.

# Implementacija backenda (laravel)

## Kratak uvod u laravel

Laravel je PHP **framework** zasnovan na **model-view-controller (MVC)** arhitekturi, koji omogućava brz razvoj aplikacija sa čitljivim i elegantnim kodom. Laravel nudi ugrađene funkcionalnosti poput autentikacije, migracija baza podataka, REST API podrške i rad sa bazom kroz **Eloquent ORM**. U ovom projektu, Laravel je korišćen za backend deo aplikacije, upravljanje bazom podataka i implementaciju **REST API** funkcionalnosti.

## Model Podataka



## Migracije

Pored generisanih migracija koje dobijamo instaliranjem laravela kreirali smo sledece migracije:

**Kreiranje tabela**

* Create\_courses\_table
* Create\_lessons\_table
* Create\_roles\_table
* Create\_purchases\_table

**Dodavanje spoljnog kljuca**

* Add\_foreign\_key\_for\_roles\_in\_user

**Promena imena kolone**

* Change\_name\_to\_rolename

**Dodavanje nove kolone**

* Add\_column\_picture\_to\_user (dodaje kolonu za cuvanje putanje od slike korisnika)

**Brisanje kolone**

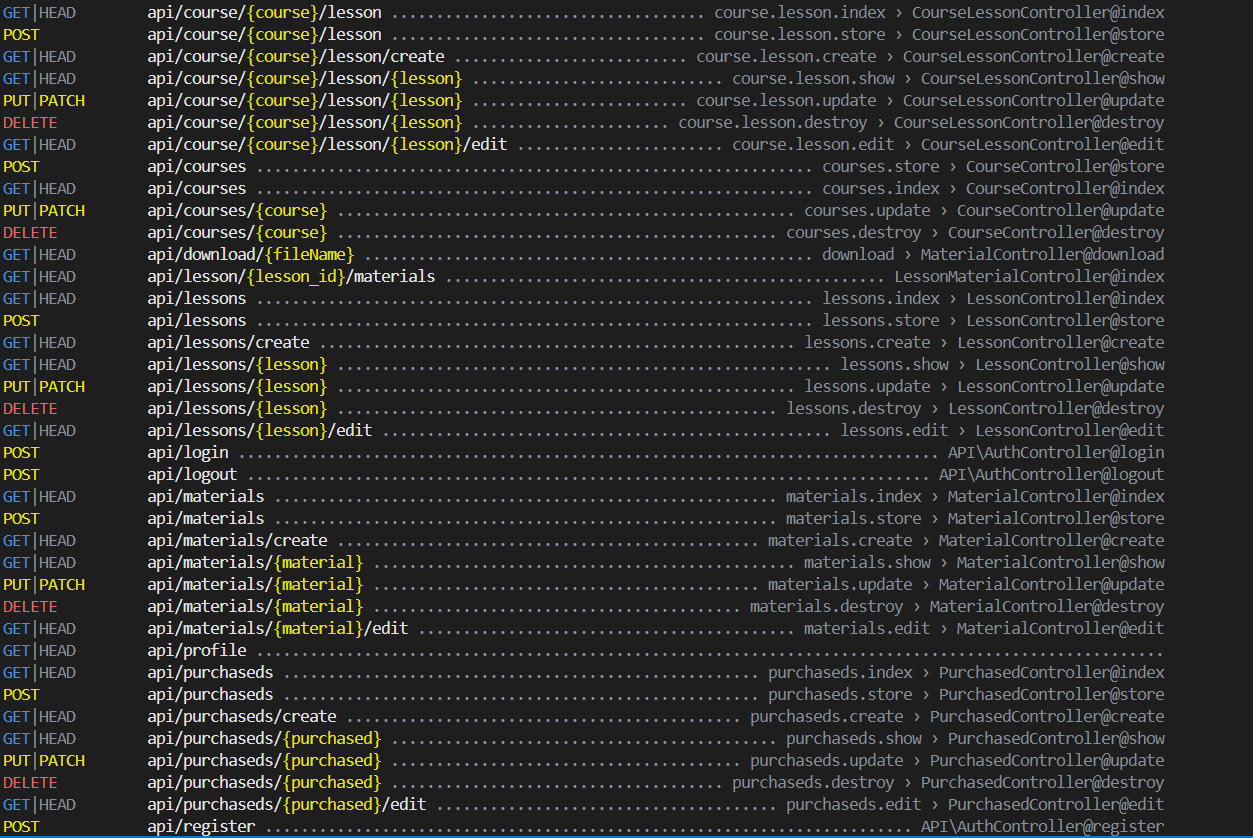
* Drop\_picture\_from\_users (uklanja kolonu za cuvanje putanje do slike korisnika)

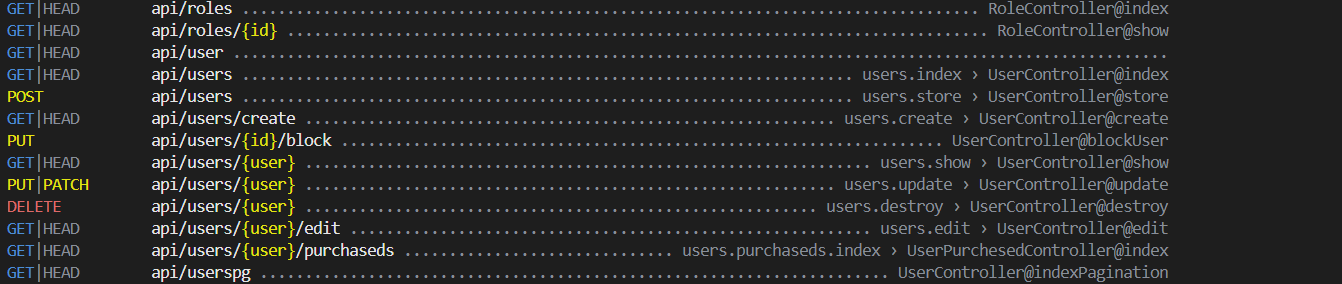
## Kontroleri

Spisak kreiranih kontrolera:

* **AuthController** – Autentikacija, implementirane login, register, logout operacije
* **CourseController** – Implementirane CRUD operacije,implementirani validatori, vraca odgovore u json formatu
* **LessonController** – Implementirane CRUD operacije,implementirani validatori, vraca odgovore u json formatu
* **RoleController** – Implementirane operacije prikazivanja uloga, vraca odgovore u json formatu
* **UserController** – Implementirane CRUD operacije,implementirani validatori, vraca odgovore u json formatu
* **UserPurchasedController** – Implementirana funkcija vracanja svih kurseva koje je kupio odabrani korisnik
* **MaterialController –** Implementirane funkcije vracanja video/tekst materijala kao i njihovo preuzimanje
* **LessonMaterialController** – vraca sve materijale za lekciju koja je prosledjena
* **CourseLessonController**  - Implementirane operacije kreiranja lekcije za prosledjen kurs, kao i izmena date lekcije

## Lista api ruta





|  |  |
| --- | --- |
| Opis funkcije | Prikaz uloga |
| HTTP metoda | GET |
| URL | /api/roles |
| URL parametri | (nema) |
| HTTP body parametri |  |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [      {          "id": 1,          "role\_name": "administrator",          "created\_at": "2024-01-30T21:59:38.000000Z",          "updated\_at": "2024-01-30T21:59:38.000000Z"      },      {          "id": 2,          "role\_name": "predavac",          "created\_at": "2024-01-30T21:59:38.000000Z",          "updated\_at": "2024-01-30T21:59:38.000000Z"      },      {          "id": 3,          "role\_name": "korisnik",          "created\_at": "2024-01-30T21:59:38.000000Z",          "updated\_at": "2024-01-30T21:59:38.000000Z"      }  ] |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz jedne uloge |
| HTTP metoda | GET |
| URL | /api/roles/{id} |
| URL parametri | (nema) |
| HTTP body parametri |  |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [      {          "id": 1,          "role\_name": "administrator",          "created\_at": "2024-01-30T21:59:38.000000Z",          "updated\_at": "2024-01-30T21:59:38.000000Z"      }  ] |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz svih kupovina |
| HTTP metoda | GET |
| URL | /api/purchaseds |
| URL parametri | (nema) |
| HTTP body parametri |  |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "Purchases": [          {              "id": 1,              "user": {                  "id": 1,                  "name": "jdsakdjas",                  "email": "9asdadas@gdgdga.com",                  "image": null,                  "role": {                      "id": 2,                      "role\_name": "predavac",                      "created\_at": "2024-01-30T21:59:38.000000Z",                      "updated\_at": "2024-01-30T21:59:38.000000Z"                  }              },              "course": {                  "id": 4,                  "title": "nasloooov",                  "description": "opiiiiiiis",                  "price": 50.25              },              "payment\_method": "Bank Transfer"          },          {              "id": 2,              "user": {                  "id": 3,                  "name": "Prof. Khalil Schowalter Sr.",                  "email": "gorn@example.org",                  "image": null,                  "role": {                      "id": 1,                      "role\_name": "administrator",                      "created\_at": "2024-01-30T21:59:38.000000Z",                      "updated\_at": "2024-01-30T21:59:38.000000Z"                  }              },              "course": {                  "id": 10,                  "title": "Dr.",                  "description": "Nobis ea modi velit quibusdam.",                  "price": 72.3              },              "payment\_method": "Bank Transfer"          },          {              "id": 3,              "user": {                  "id": 10,                  "name": "Davin Wisozk",                  "email": "carolyne.willms@example.org",                  "image": null,                  "role": {                      "id": 1,                      "role\_name": "administrator",                      "created\_at": "2024-01-30T21:59:38.000000Z",                      "updated\_at": "2024-01-30T21:59:38.000000Z"                  }              },              "course": {                  "id": 8,                  "title": "Mr.",                  "description": "Ipsa illum qui neque impedit officia sed.",                  "price": 15.88              },              "payment\_method": "Bank Transfer"          },          {              "id": 4,              "user": {                  "id": 6,                  "name": "Floyd Kautzer",                  "email": "nelle74@example.org",                  "image": null,                  "role": {                      "id": 3,                      "role\_name": "korisnik",                      "created\_at": "2024-01-30T21:59:38.000000Z",                      "updated\_at": "2024-01-30T21:59:38.000000Z"                  }              },              "course": {                  "id": 2,                  "title": "Mr.",                  "description": "Mollitia et iusto rerum sint voluptatem architecto alias.",                  "price": 32.89              },              "payment\_method": "Bank Transfer"          },          {              "id": 5,              "user": {                  "id": 9,                  "name": "Beaulah Crist",                  "email": "goodwin.victoria@example.com",                  "image": null,                  "role": {                      "id": 1,                      "role\_name": "administrator",                      "created\_at": "2024-01-30T21:59:38.000000Z",                      "updated\_at": "2024-01-30T21:59:38.000000Z"                  }              },              "course": {                  "id": 3,                  "title": "Dr.",                  "description": "Dignissimos repudiandae eos iure modi.",                  "price": 72.65              },              "payment\_method": "Bank Transfer"          },          {              "id": 6,              "user": {                  "id": 4,                  "name": "Xavier Doyle",                  "email": "ghauck@example.net",                  "image": null,                  "role": {                      "id": 2,                      "role\_name": "predavac",                      "created\_at": "2024-01-30T21:59:38.000000Z",                      "updated\_at": "2024-01-30T21:59:38.000000Z"                  }              },              "course": {                  "id": 2,                  "title": "Mr.",                  "description": "Mollitia et iusto rerum sint voluptatem architecto alias.",                  "price": 32.89              },              "payment\_method": "Bank Transfer"          },          {              "id": 7,              "user": {                  "id": 4,                  "name": "Xavier Doyle",                  "email": "ghauck@example.net",                  "image": null,                  "role": {                      "id": 2,                      "role\_name": "predavac",                      "created\_at": "2024-01-30T21:59:38.000000Z",                      "updated\_at": "2024-01-30T21:59:38.000000Z"                  }              },              "course": {                  "id": 3,                  "title": "Dr.",                  "description": "Dignissimos repudiandae eos iure modi.",                  "price": 72.65              },              "payment\_method": "Credit Card"          },          {              "id": 8,              "user": {                  "id": 9,                  "name": "Beaulah Crist",                  "email": "goodwin.victoria@example.com",                  "image": null,                  "role": {                      "id": 1,                      "role\_name": "administrator",                      "created\_at": "2024-01-30T21:59:38.000000Z",                      "updated\_at": "2024-01-30T21:59:38.000000Z"                  }              },              "course": {                  "id": 3,                  "title": "Dr.",                  "description": "Dignissimos repudiandae eos iure modi.",                  "price": 72.65              },              "payment\_method": "credit card"          }      ]  } |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz jedne kupovine |
| HTTP metoda | GET |
| URL | api/purchaseds/1 |
| URL parametri | (nema) |
| HTTP body parametri |  |
| Format HTTP body parametara | JSON |
| Izlazni parametri | "Purchased": {          "id": 1,          "user": {              "id": 1,              "name": "jdsakdjas",              "email": "9asdadas@gdgdga.com",              "image": null,              "role": {                  "id": 2,                  "role\_name": "predavac",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          },          "course": {              "id": 4,              "title": "nasloooov",              "description": "opiiiiiiis",              "price": 50.25          },          "payment\_method": "Bank Transfer"      }  } |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje kupovine |
| HTTP metoda | DELETE |
| URL | /api/purchaseds/{id} |
| URL parametri | (nema) |
| HTTP body parametri |  |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [      "Purchase was deleted successfully"  ] |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Dodavanje kupovine |
| HTTP metoda | POST |
| URL | /api/purchaseds/ |
| URL parametri | (nema) |
| HTTP body parametri |  |
| Format HTTP body parametara | course\_id:10  user\_id:4  payment\_method:credit card |
| Izlazni parametri | [      "Lesson was created successfully",      {          "id": 12,          "user": {              "id": 4,              "name": "Xavier Doyle",              "email": "ghauck@example.net",              "image": null,              "role": {                  "id": 2,                  "role\_name": "predavac",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          },          "course": {              "id": 10,              "title": "Dr.",              "description": "Nobis ea modi velit quibusdam.",              "price": 72.3          },          "payment\_method": "credit card"      }  ] |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Izmena kupovine |
| HTTP metoda | PUT |
| URL | /api/purchaseds/{id} |
| URL parametri | (nema) |
| HTTP body parametri | {  course\_id:9  user\_id:4  payment\_method:credit card  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [      "Lesson was created successfully",      {          "id": 12,          "user": {              "id": 4,              "name": "Xavier Doyle",              "email": "ghauck@example.net",              "image": null,              "role": {                  "id": 2,                  "role\_name": "predavac",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          },          "course": {              "id": 10,              "title": "Dr.",              "description": "Nobis ea modi velit quibusdam.",              "price": 72.3          },          "payment\_method": "credit card"      }  ] |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz Lekcija |
| HTTP metoda | GET |
| URL | /api/lessons |
| URL parametri | (nema) |
| HTTP body parametri |  |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "lessons": [          {              "id": 1,              "title": "nasloooov",              "content": "blablabla",              "course": {                  "id": 1,                  "title": "Prof.",                  "description": "Sequi aut corporis porro sequi ea expedita repudiandae.",                  "price": 58.04              }          },          {              "id": 3,              "title": "Magni ad quia magnam reprehenderit.",              "content": "Asperiores magni ut omnis et. Aliquid cupiditate est sint. Reprehenderit molestiae repellat dignissimos nemo voluptatem porro.",              "course": {                  "id": 9,                  "title": "Dr.",                  "description": "Est sint eos quod quidem.",                  "price": 97.01              }          },          {              "id": 4,              "title": "Sint voluptas atque error.",              "content": "At ratione rerum molestiae. Delectus placeat cumque saepe dolore nesciunt reiciendis. Cum voluptas error neque delectus. Eveniet aut optio est et quia.",              "course": {                  "id": 9,                  "title": "Dr.",                  "description": "Est sint eos quod quidem.",                  "price": 97.01              }          },          {              "id": 5,              "title": "Sit labore occaecati doloribus eligendi quae.",              "content": "Ea est repellat nostrum ratione sit possimus quaerat. Dolores culpa dolorum consequuntur in. Qui illo nobis aut at ut vel. Et adipisci voluptas aspernatur rerum et.",              "course": {                  "id": 3,                  "title": "Dr.",                  "description": "Dignissimos repudiandae eos iure modi.",                  "price": 72.65              }          },          {              "id": 6,              "title": "Aut id eveniet fugit eum culpa praesentium beatae.",              "content": "Ut et vel recusandae iste cupiditate ducimus nesciunt. Quas sapiente aliquam voluptatem. Assumenda necessitatibus nulla numquam illum eveniet. Et voluptatem exercitationem qui aperiam explicabo maiores voluptatem soluta.",              "course": {                  "id": 4,                  "title": "nasloooov",                  "description": "opiiiiiiis",                  "price": 50.25              }          },          {              "id": 7,              "title": "Non qui pariatur reprehenderit facere eveniet nam earum.",              "content": "Mollitia libero sit nulla tempore dolor. Enim ex expedita porro quia placeat labore. Illum ipsa dolores sed reiciendis. Qui voluptatem totam id quasi.",              "course": {                  "id": 5,                  "title": "Mr.",                  "description": "Consequatur nam et minus.",                  "price": 58.6              }          },          {              "id": 8,              "title": "Illo sed et dolor quia.",              "content": "Occaecati dolore nostrum fugit eum excepturi est facilis. Molestiae nam vero maxime. Et eum aliquam at est quia fuga ullam. Ut blanditiis quod a ut dolorum. Quia molestiae id dolorem temporibus doloremque aut.",              "course": {                  "id": 5,                  "title": "Mr.",                  "description": "Consequatur nam et minus.",                  "price": 58.6              }          },          {              "id": 9,              "title": "Dicta ducimus voluptatem adipisci tenetur explicabo quam.",              "content": "Accusantium necessitatibus iste atque aliquam porro ipsum. Alias laboriosam qui ut nihil. Dolorem aut possimus assumenda. Ut hic deserunt cupiditate suscipit voluptates veritatis vitae.",              "course": {                  "id": 9,                  "title": "Dr.",                  "description": "Est sint eos quod quidem.",                  "price": 97.01              }          },          {              "id": 10,              "title": "Sunt debitis aut quo fuga.",              "content": "Cum sed quia temporibus sed ut voluptatum ut. Omnis nulla dolore suscipit repellat inventore autem velit. Consequuntur cum voluptas consequatur. Autem dolor dolores inventore aut aut quaerat.",              "course": {                  "id": 1,                  "title": "Prof.",                  "description": "Sequi aut corporis porro sequi ea expedita repudiandae.",                  "price": 58.04              }          },          {              "id": 11,              "title": "naslov2",              "content": "sadrzaj",              "course": {                  "id": 1,                  "title": "Prof.",                  "description": "Sequi aut corporis porro sequi ea expedita repudiandae.",                  "price": 58.04              }          },          {              "id": 12,              "title": "naslov2",              "content": "sadrzaj",              "course": {                  "id": 1,                  "title": "Prof.",                  "description": "Sequi aut corporis porro sequi ea expedita repudiandae.",                  "price": 58.04              }          },          {              "id": 13,              "title": "naslov23",              "content": "sadrzaj",              "course": {                  "id": 1,                  "title": "Prof.",                  "description": "Sequi aut corporis porro sequi ea expedita repudiandae.",                  "price": 58.04              }          },          {              "id": 14,              "title": "nasloooov",              "content": "blablabla",              "course": {                  "id": 1,                  "title": "Prof.",                  "description": "Sequi aut corporis porro sequi ea expedita repudiandae.",                  "price": 58.04              }          }      ]  } |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz jedne lekcije |
| HTTP metoda | GET |
| URL | /api/lessons /{id} |
| URL parametri | (nema) |
| HTTP body parametri |  |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "lesson": {          "id": 1,          "title": "nasloooov",          "content": "blablabla",          "course": {              "id": 1,              "title": "Prof.",              "description": "Sequi aut corporis porro sequi ea expedita repudiandae.",              "price": 58.04          }      }  } |
| Format izlaznih parametara | json |
| Opis funkcije | Brisanje Lekcije |
| HTTP metoda | DELETE |
| URL | /api/lessons /{id} |
| URL parametri | (nema) |
| HTTP body parametri |  |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [      "Lesson was deleted successfully"  ] |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Dodavanje lekcije |
| HTTP metoda | POST |
| URL | /api/lessons |
| URL parametri | (nema) |
| HTTP body parametri | {  course\_id:9  content:djsakldjaskldjkalsjdkl  title:NAAslov  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [      "Lesson is created successfully",      {          "id": 15,          "title": "NAAslov",          "content": "djsakldjaskldjkalsjdkl",          "course": {              "id": 9,              "title": "Dr.",              "description": "Est sint eos quod quidem.",              "price": 97.01          }      }  ] |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Izmena lekcije |
| HTTP metoda | PUT |
| URL | /api/lessons/{id} |
| URL parametri | (nema) |
| HTTP body parametri | {  course\_id:9  content:djsakldjaskldjkalsjdkl  title:NAAslovvvv  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [      "Lesson is updated successfully",      {          "id": 15,          "title": "NAAslovvvv",          "content": "djsakldjaskldjkalsjdkl",          "course": {              "id": 9,              "title": "Dr.",              "description": "Est sint eos quod quidem.",              "price": 97.01          }      }  ] |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz kurseva |
| HTTP metoda | GET |
| URL | /api/courses |
| URL parametri | (nema) |
| HTTP body parametri |  |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "course": [          {              "id": 1,              "title": "Prof.",              "description": "Sequi aut corporis porro sequi ea expedita repudiandae.",              "price": 58.04          },          {              "id": 2,              "title": "Mr.",              "description": "Mollitia et iusto rerum sint voluptatem architecto alias.",              "price": 32.89          },          {              "id": 3,              "title": "Dr.",              "description": "Dignissimos repudiandae eos iure modi.",              "price": 72.65          },          {              "id": 4,              "title": "nasloooov",              "description": "opiiiiiiis",              "price": 50.25          },          {              "id": 5,              "title": "Mr.",              "description": "Consequatur nam et minus.",              "price": 58.6          },          {              "id": 6,              "title": "Mr.",              "description": "Sed sit aut odit est quae.",              "price": 45.9          },          {              "id": 7,              "title": "Ms.",              "description": "Animi voluptatibus consequatur ea dignissimos veniam quisquam.",              "price": 81.38          },          {              "id": 8,              "title": "Mr.",              "description": "Ipsa illum qui neque impedit officia sed.",              "price": 15.88          },          {              "id": 9,              "title": "Dr.",              "description": "Est sint eos quod quidem.",              "price": 97.01          },          {              "id": 10,              "title": "Dr.",              "description": "Nobis ea modi velit quibusdam.",              "price": 72.3          },          {              "id": 11,              "title": "naslov",              "description": "opisopisopsi",              "price": 27.56          },          {              "id": 13,              "title": "\"nasloooov\"",              "description": "\"opiiiiiiis\"",              "price": 50.25          },          {              "id": 14,              "title": "nasloooov",              "description": "opiiiiiiis",              "price": 50.25          },          {              "id": 15,              "title": "nasloooov",              "description": "opiiiiiiis",              "price": 50.25          }      ]  } |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz jednog kursa |
| HTTP metoda | GET |
| URL | /api/courses/{id} |
| URL parametri | (nema) |
| HTTP body parametri |  |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "course": {          "id": 5,          "title": "Mr.",          "description": "Consequatur nam et minus.",          "price": 58.6      }  } |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje kursa |
| HTTP metoda | DELETE |
| URL | /api/courses/{id} |
| URL parametri | (nema) |
| HTTP body parametri |  |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {     "Course was deleted successfully"  } |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Kreiranje kursa |
| HTTP metoda | POST |
| URL | /api/courses |
| URL parametri | (nema) |
| HTTP body parametri | {  title:nassslo  description:desc  price:99.7  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [      "Course is created successfully",      {          "id": 16,          "title": "nassslo",          "description": "desc",          "price": "99.7"      }  ] |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Izmena kursa |
| HTTP metoda | PUT |
| URL | /api/courses/{id} |
| URL parametri | (nema) |
| HTTP body parametri | {  title:nassslov  description:desc  price:99.7  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [      "Course was updated successfully.",      {          "id": 16,          "title": "nassslov",          "description": "desc",          "price": "99.7"      }  ] |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz Svih Korisnika |
| HTTP metoda | GET |
| URL | /api/users |
| URL parametri | (nema) |
| HTTP body parametri |  |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "users": [          {              "id": 1,              "name": "jdsakdjas",              "email": "9asdadas@gdgdga.com",              "image": null,              "role": {                  "id": 2,                  "role\_name": "predavac",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          },          {              "id": 2,              "name": "Ms. Noelia Block IV",              "email": "goodwin.chance@example.com",              "image": null,              "role": {                  "id": 1,                  "role\_name": "administrator",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          },          {              "id": 3,              "name": "Prof. Khalil Schowalter Sr.",              "email": "gorn@example.org",              "image": null,              "role": {                  "id": 1,                  "role\_name": "administrator",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          },          {              "id": 4,              "name": "Xavier Doyle",              "email": "ghauck@example.net",              "image": null,              "role": {                  "id": 2,                  "role\_name": "predavac",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          },          {              "id": 5,              "name": "Mrs. Leora Auer",              "email": "cletus04@example.net",              "image": null,              "role": {                  "id": 2,                  "role\_name": "predavac",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          },          {              "id": 6,              "name": "Floyd Kautzer",              "email": "nelle74@example.org",              "image": null,              "role": {                  "id": 3,                  "role\_name": "korisnik",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          },          {              "id": 7,              "name": "Carmine Armstrong",              "email": "schuppe.elena@example.com",              "image": null,              "role": {                  "id": 3,                  "role\_name": "korisnik",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          },          {              "id": 8,              "name": "Akeem Walter",              "email": "mikel.medhurst@example.com",              "image": null,              "role": {                  "id": 2,                  "role\_name": "predavac",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          },          {              "id": 9,              "name": "Beaulah Crist",              "email": "goodwin.victoria@example.com",              "image": null,              "role": {                  "id": 1,                  "role\_name": "administrator",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          },          {              "id": 10,              "name": "Davin Wisozk",              "email": "carolyne.willms@example.org",              "image": null,              "role": {                  "id": 1,                  "role\_name": "administrator",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          },          {              "id": 11,              "name": "jovan",              "email": "jovanvukovic2001@gmail.com",              "image": null,              "role": {                  "id": 1,                  "role\_name": "administrator",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          },          {              "id": 12,              "name": "jovan",              "email": "jovanvukovic20011@gmail.com",              "image": null,              "role": {                  "id": 1,                  "role\_name": "administrator",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          }      ]  } |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz jednog korsinika |
| HTTP metoda | GET |
| URL | /api/users /{id} |
| URL parametri | (nema) |
| HTTP body parametri |  |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "user": {          "id": 5,          "name": "Mrs. Leora Auer",          "email": "cletus04@example.net",          "image": null,          "role": {              "id": 2,              "role\_name": "predavac",              "created\_at": "2024-01-30T21:59:38.000000Z",              "updated\_at": "2024-01-30T21:59:38.000000Z"          }      }  } |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje korisnika |
| HTTP metoda | DELETE |
| URL | /api/users /{id} |
| URL parametri | (nema) |
| HTTP body parametri |  |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [      "User was deleted successfully"  ] |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Kreiranje korisnika |
| HTTP metoda | POST |
| URL | /api/users |
| URL parametri | (nema) |
| HTTP body parametri | {  name:Nikola  role\_id:2  email:jdaskldja@gmail.com  password:12345678  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [      "User was created successfully",      {          "id": 13,          "name": "Nikola",          "email": "jdaskldja@gmail.com",          "image": null,          "role": 1      }  ] |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Izmena korisnika |
| HTTP metoda | POST |
| URL | /api/users/{id} |
| URL parametri | (nema) |
| HTTP body parametri | {  name:Nikolaa  role\_id:2  email:jdaskldja@gmail.com  password:12345678  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [      "User was updated successfully",      {          "id": 13,          "name": "Nikola",          "email": "jdaskldjaa@gmail.com",          "image": null,          "role": {              "id": 1,              "role\_name": "administrator",              "created\_at": "2024-01-30T21:59:38.000000Z",              "updated\_at": "2024-01-30T21:59:38.000000Z"          }      }  ] |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Kreiranje korisnika |
| HTTP metoda | POST |
| URL | /api/users/{id} |
| URL parametri | (nema) |
| HTTP body parametri | {  name:Nikola  role\_id:2  email:jdaskldja@gmail.com  password:12345678  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | [      "User was created successfully",      {          "id": 13,          "name": "Nikola",          "email": "jdaskldjaa@gmail.com",          "image": null,          "role": {              "id": 1,              "role\_name": "administrator",              "created\_at": "2024-01-30T21:59:38.000000Z",              "updated\_at": "2024-01-30T21:59:38.000000Z"          }      }  ] |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Lista svih kupljenih kurseva od strane jednog korisnika |
| HTTP metoda | GET |
| URL | /api/users/{id}/purcheseds |
| URL parametri | (nema) |
| HTTP body parametri |  |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "1": {          "id": 3,          "user\_id": 10,          "course\_id": 8,          "payment\_method": "Bank Transfer",          "created\_at": "2024-01-30T21:59:38.000000Z",          "updated\_at": "2024-01-30T21:59:38.000000Z"      }  } |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Lista korisnika sa paginacijom |
| HTTP metoda | GET |
| URL | /api/userspg |
| URL parametri | (nema) |
| HTTP body parametri |  |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "users": [          {              "id": 1,              "name": "jdsakdjas",              "email": "9asdadas@gdgdga.com",              "image": null,              "role": {                  "id": 2,                  "role\_name": "predavac",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          },          {              "id": 2,              "name": "Ms. Noelia Block IV",              "email": "goodwin.chance@example.com",              "image": null,              "role": {                  "id": 1,                  "role\_name": "administrator",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          },          {              "id": 3,              "name": "Prof. Khalil Schowalter Sr.",              "email": "gorn@example.org",              "image": null,              "role": {                  "id": 1,                  "role\_name": "administrator",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          },          {              "id": 4,              "name": "Xavier Doyle",              "email": "ghauck@example.net",              "image": null,              "role": {                  "id": 2,                  "role\_name": "predavac",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          },          {              "id": 6,              "name": "Floyd Kautzer",              "email": "nelle74@example.org",              "image": null,              "role": {                  "id": 3,                  "role\_name": "korisnik",                  "created\_at": "2024-01-30T21:59:38.000000Z",                  "updated\_at": "2024-01-30T21:59:38.000000Z"              }          }      ],      "links": {          "first": "http://127.0.0.1:8000/api/userspg?page=1",          "last": "http://127.0.0.1:8000/api/userspg?page=3",          "prev": null,          "next": "http://127.0.0.1:8000/api/userspg?page=2"      },      "meta": {          "current\_page": 1,          "from": 1,          "last\_page": 3,          "links": [              {                  "url": null,                  "label": "&laquo; Previous",                  "active": false              },              {                  "url": "http://127.0.0.1:8000/api/userspg?page=1",                  "label": "1",                  "active": true              },              {                  "url": "http://127.0.0.1:8000/api/userspg?page=2",                  "label": "2",                  "active": false              },              {                  "url": "http://127.0.0.1:8000/api/userspg?page=3",                  "label": "3",                  "active": false              },              {                  "url": "http://127.0.0.1:8000/api/userspg?page=2",                  "label": "Next &raquo;",                  "active": false              }          ],          "path": "http://127.0.0.1:8000/api/userspg",          "per\_page": 5,          "to": 5,          "total": 12      }  } |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | Logout |
| HTTP metoda | POST |
| URL | /api/logout |
| URL parametri | (nema) |
| HTTP body parametri |  |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "message": "You have successfully logged out."  } |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | login |
| HTTP metoda | POST |
| URL | /api/login |
| URL parametri | (nema) |
| HTTP body parametri | {  email:jdaskldjaa@gmail.coms  password:12345678  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "message": "Hi Nikola Welcome to home",      "access\_token": "6|U9tBobiRghRoJFigeKN5NxMACC0zLs0Vu5gOC6it9d227c8c",      "token\_type": "Bearer"  } |
| Format izlaznih parametara | json |

|  |  |
| --- | --- |
| Opis funkcije | register |
| HTTP metoda | POST |
| URL | /api/register |
| URL parametri | (nema) |
| HTTP body parametri | {  name:Nikola  role\_id:2  email:jdaskldjaa@gmail.coms  password:12345678  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "data": {          "name": "Nikola",          "email": "jdaskldjaa@gmail.coms",          "updated\_at": "2024-02-01T21:57:38.000000Z",          "created\_at": "2024-02-01T21:57:38.000000Z",          "id": 14      },      "access\_token": "7|j2IKv4yn29UBbfkawssC6P5jEqIcngeClgqs12uic844c58c",      "token\_type": "Bearer"  } |
| Format izlaznih parametara | json |

## Korisničko uputstvo (backend)

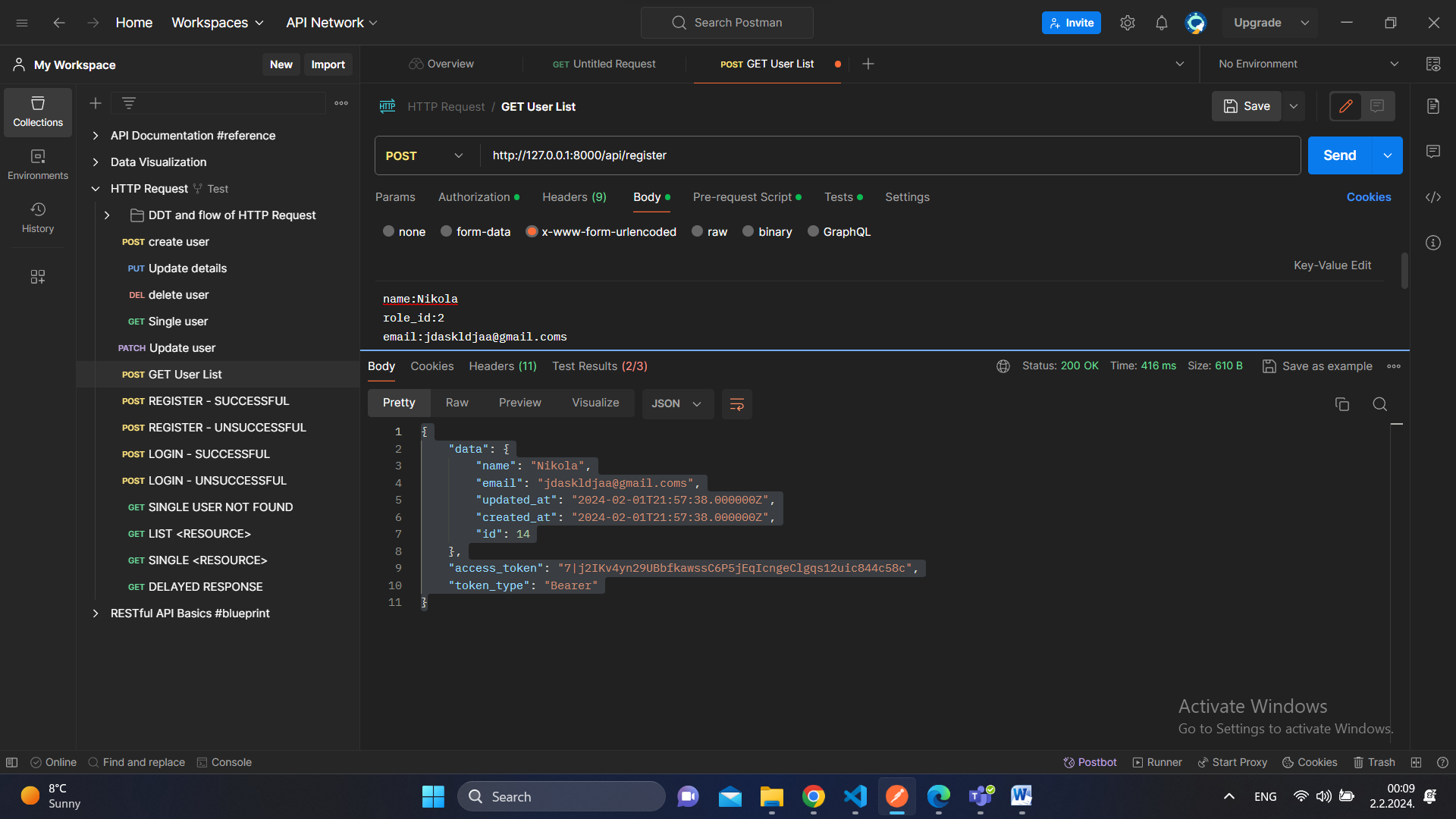
Naša aplikacija je RESTful API koji omogućava korisnicima da upravljaju kursevima, lekcijama, korisnicima i kupovinama kurseva. Ovaj API pruža platformu za korisnike da istražuju, kupuju, prate napredak i upravljaju svojim učenjem putem različitih kurseva i lekcija. Takođe omogućava administratorima da efikasno upravljaju sadržajem, korisnicima i transakcijama u sistemu.

Ovde su neke od ključnih funkcionalnosti API-ja:

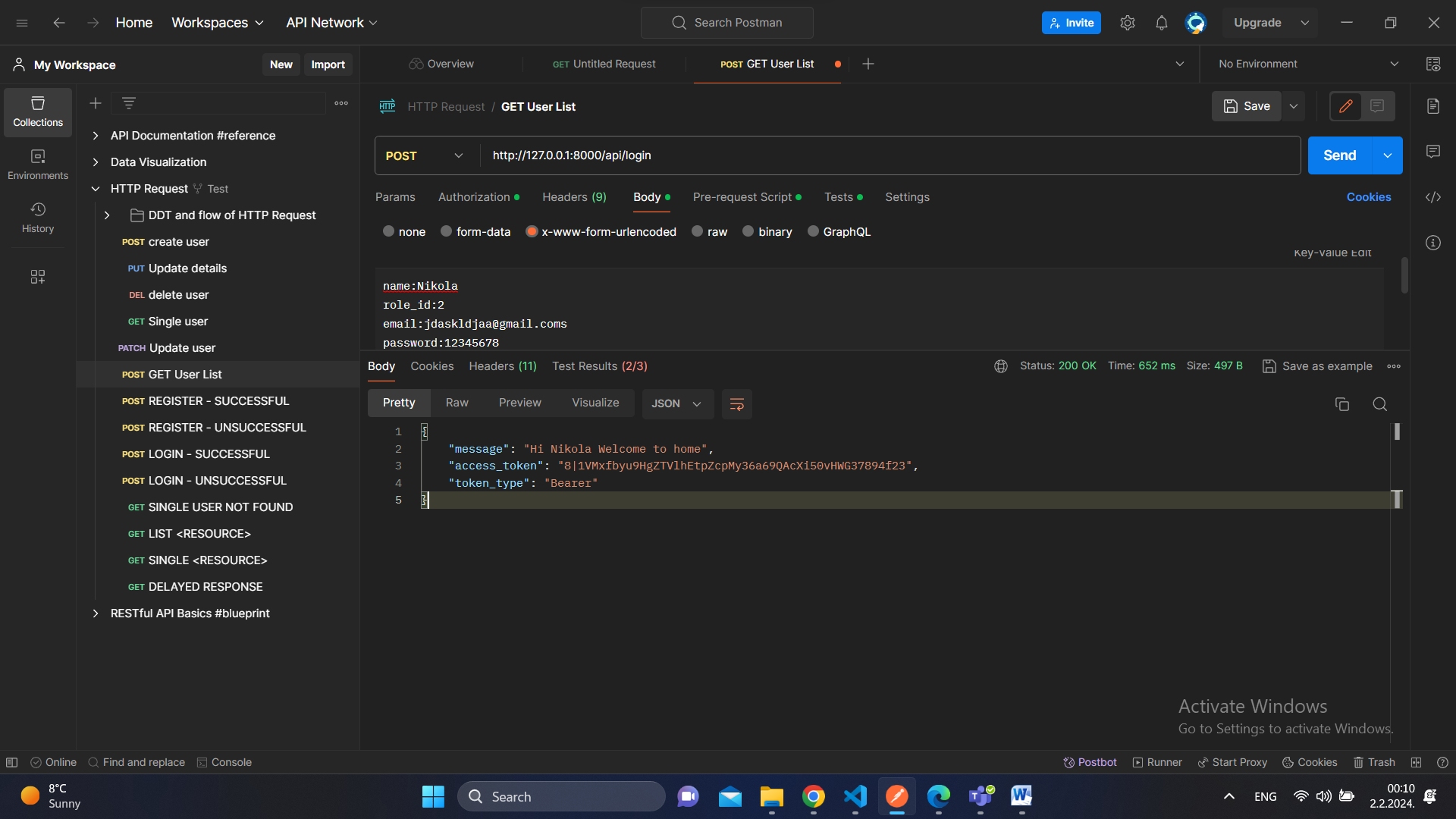
1. **Autentikacija i autorizacija**: Korisnici se mogu prijaviti i dobiti autentifikacioni token koji koriste za pristup zaštićenim rutama. Pristup određenim rutama može biti ograničen na osnovu uloge korisnika.
2. **Upravljanje korisnicima**: Korisnici mogu da se registruju, prijavljuju i ažuriraju svoje profile. Administratori mogu da pregledaju, ažuriraju i brišu korisničke naloge.
3. **Upravljanje kursevima i lekcijama**: Korisnici mogu da pregledaju dostupne kurseve i lekcije. Administratori mogu da dodaju, ažuriraju i brišu kurseve i lekcije.
4. **Kupovina kurseva**: Korisnici mogu da kupuju kurseve i pristupaju lekcijama u okviru tih kurseva.

Ovde su neki od mogućih endpoint-ova koje naš API podržava:

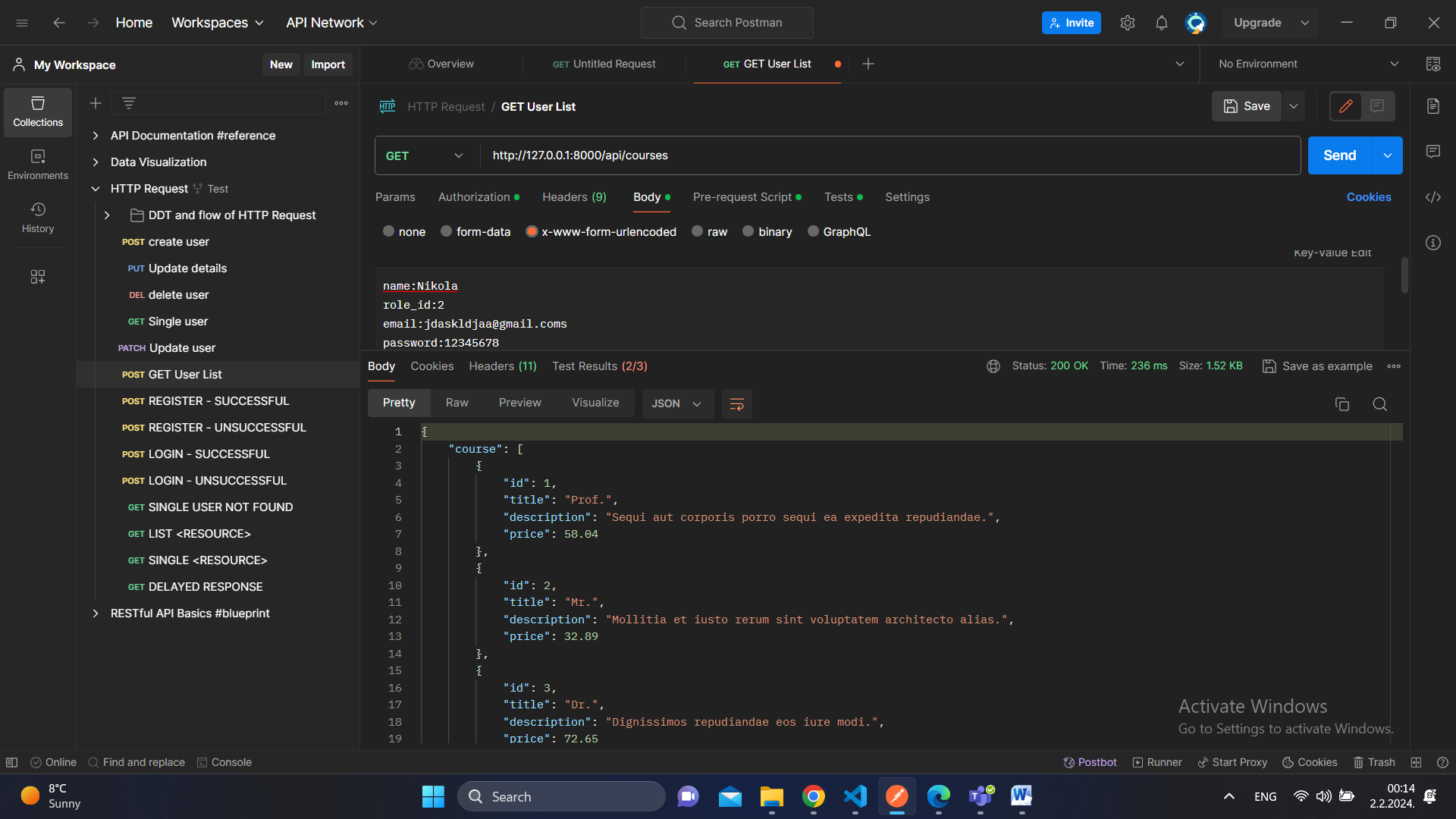
* **POST /api/register**: Registracija novog korisnika.
* **POST /api/login**: Prijava korisnika.
* **GET /api/courses**: Pregled dostupnih kurseva.
* **GET /api/courses/{id}**: Prikaz određenog kursa.
* **PUT /api/user/profile**: Ažuriranje korisničkog profila.



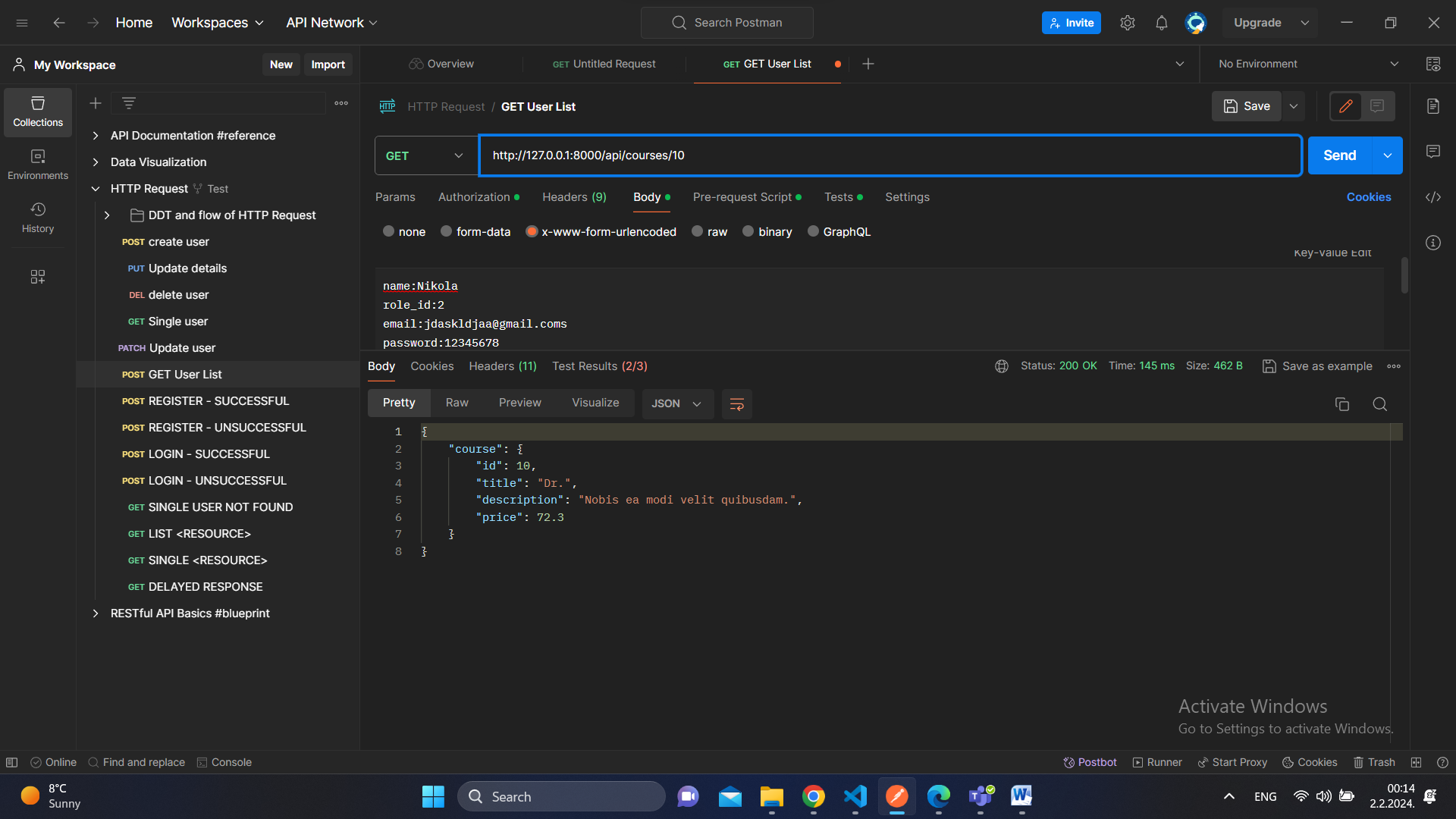
Slika - Registracija korisnika



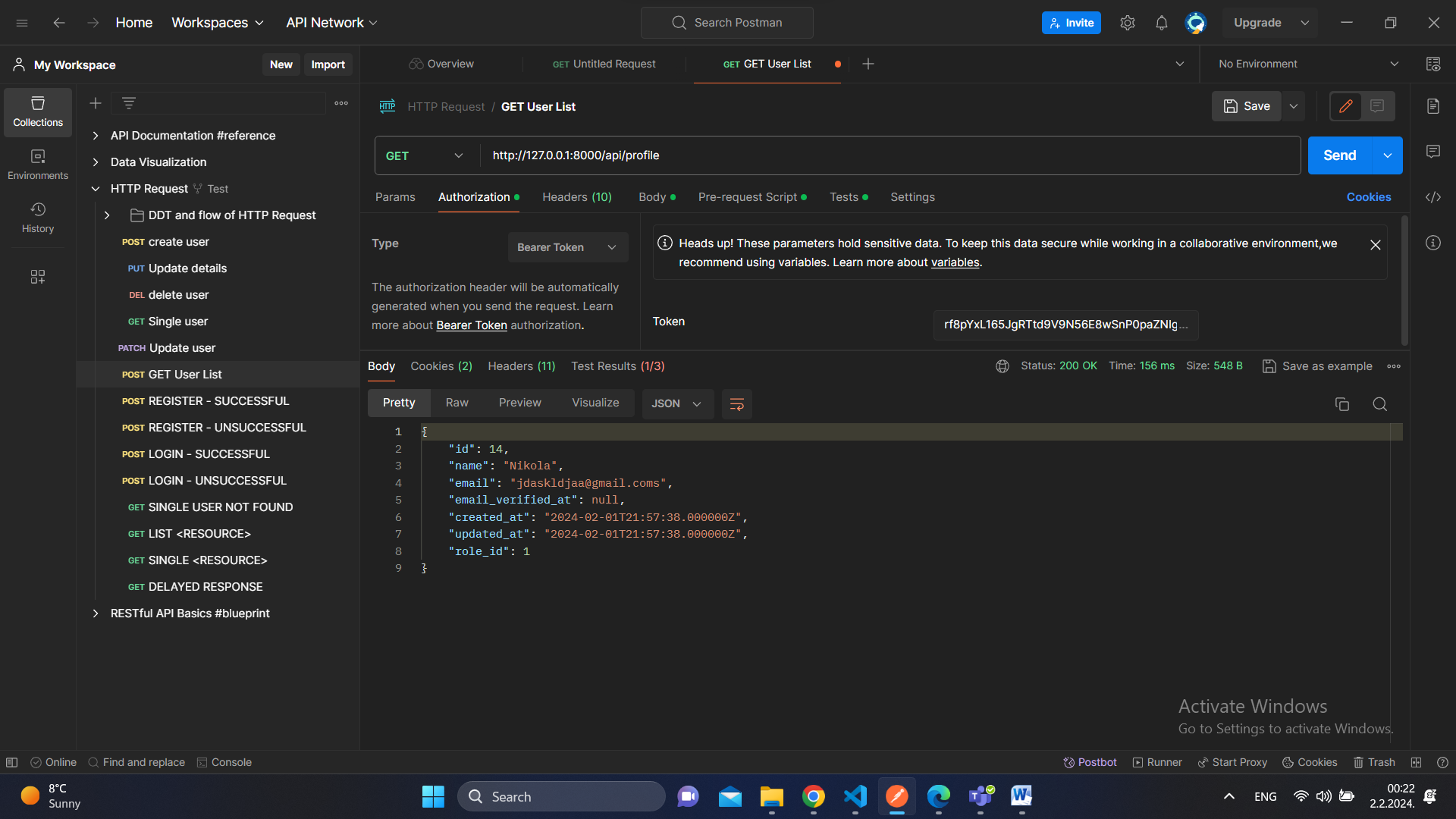
Slika - Login



Slika - Prikaz kurseva



Slika - Prikaz jednog kursa



Slika - Prikaz profil korisnika

# Implementacija frontenda(React)

## Kratak uvod u React

React.js je popularna JavaScript biblioteka za izradu korisničkih interfejsa. Omogućava razvoj dinamičkih i interaktivnih web aplikacija kroz komponentni pristup, gde se UI deli na male, samostalne i ponovno iskoristive komponente. React koristi virtualni DOM, čime povećava brzinu renderovanja i performanse aplikacije. U ovom projektu React.js je korišćen za frontend aplikaciju, pružajući intuitivno i responzivno korisničko iskustvo.

## Karakteristicni delovi koda

Izvrsavanje svake React aplikacije pocinje iz **index.js** fajla

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

import "./index.css";

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

    <App />

);

Root uzima dokument iz korenskog index.html fajla, i zatim kreira komponentu App unutar tog korenskog čvora čime se pokreće cela aplikacija

**App.js**

App.js je centralna tacka aplikacije. Ona obezbeđuje osnovne funkcionalnosti kao sto su upravljanje kursevima, korisničkim interfejsom i navigacijom. Sadrzi sve bitne komponente kao i logiku za upravljanjem stanja aplikacije. Importuje sve kljucne komponente kao sto **NavBar, Footer, Login** kao i fotografije iz foldera **assets.**

import Navbar from "./components/NavBar/Navbar.jsx";

import Footer from "./components/Footer/Footer.jsx";

import Login from "./pages/LoginPage/Login.jsx";

import Register from "./pages/RegisterPage/Register.jsx";

import { useState, useEffect } from "react";

import { BrowserRouter, Routes, Route } from "react-router-dom";

import UserProfile from "./pages/UserPages/UserProfilePage/UserProfile.jsx";

import axios from "axios";

import Cart from "./pages/Cart/Cart.jsx";

import Course from "./pages/CoursesPage/Course.jsx";

import HomePage from "./pages/HomePage/HomePage.jsx";

import CheckoutPage from "./pages/CheckoutPage/CheckoutPage.jsx";

import UserCourses from "./pages/UserPages/UserCoursePage/UserCourses.jsx";

import AdminLogin from "./pages/AdminPages/AdminLogin/AdminLogin.jsx";

import Users from "./pages/AdminPages/Users/Users.jsx";

import UserDetails from "./pages/AdminPages/UserDetails/UserDetails.jsx";

import AdminRegister from "./pages/AdminPages/Register admin/AdminRegister.jsx";

import Predavaci from "./pages/AdminPages/Predavaci/Predavaci.jsx";

import TeacherDetails from "./pages/AdminPages/TeacherDetails/TeacherDetails.jsx";

import AddCoursePage from "./pages/TeacherPages/AddCoursePage.jsx";

import TeacherCoursesPage from "./pages/TeacherPages/TeacherCourses/TeacherCoursesPage.jsx";

import TeacherCourseDetailsPage from "./pages/TeacherPages/TeacherCourseDetailsPage/TeacherCourseDetailsPage.jsx";

import AddLessonPage from "./pages/TeacherPages/AddLessonPage/AddLessonPage.jsx";

import LessonMaterialsPage from "./pages/TeacherPages/LessonMaterialsPage/LessonMaterialsPage.jsx";

function App() {

  const[courses, setCourses] = useState();

  useEffect(()=>{

    if(courses == null){

      axios.get("api/courses").then((res)=>{

        const coursesWithAmount = res.data.course.map(course => ({

          ...course,

          amount: 0,

        }));

        setCourses(coursesWithAmount);

  });

    }

  },[courses]);

  const updateCoursesAfterAdding = () => {

    axios.get("api/courses").then((res) => {

      const coursesWithAmount = res.data.course.map(course => ({

        ...course,

        amount: 0,

      }));

      setCourses(coursesWithAmount);

    }).catch((e) => {

      console.error("Error updating courses:", e);

    });

  };

  const [cartNum, setCartNum] = useState(0);

  const [cartCourses, setCartCourses] = useState([]);

  const [token, setToken] = useState();

  const [user, setUser] = useState();

  const[loggedUserCourses, setLoggedUserCourses] = useState();

  useEffect(()=>{

      if( user != null){

          axios.get('/api/users/'+ user.id +'/purchaseds').then((res) => {

              setLoggedUserCourses(res.data.data);

          });

      }

    },[user]);

    function updateCourses() {

      if (user != null) {

          axios.get('/api/users/' + user.id + '/purchaseds').then((res) => {

              setLoggedUserCourses(res.data.data);

          });

      }

  }

  const addToCart = (id) => {

    const isInCart = cartCourses.some((course) => course.id === id);

    if (isInCart) {

      alert("Ne mozete kupiti 2 puta isti kurs");

      return;

    }

    const updatedCourses = courses.map((course) => {

      if (course.id === id) {

        course.amount = 1;

      }

      return course;

    });

    setCartCourses(updatedCourses.filter((course) => course.amount > 0));

    setCartNum(cartNum + 1);

  };

  const remFromCart = (id) => {

    courses.map((course) => {

      if (course.id === id) {

        if (course.amount > 0) {

          course.amount = 0;

          setCartNum(cartNum - 1);

          refreshCart();

          console.log("product id=", course.id, "amount=", course.amount);

        } else {

          alert("Amount of product is already 0.");

        }

      }

      return course;

    });

  };

  function addCourseToUserCourses(newCourse) {

    console.log('Adding course:', newCourse);

    setLoggedUserCourses((prevCourses) => [...prevCourses, newCourse]);

}

  const refreshCart = () => {

    const newCourses = cartCourses.filter((course) => course.amount > 0);

    setCartCourses(newCourses);

  };

  function calculateTotal() {

    return cartCourses.reduce((total, course) => total + course.price \* course.amount, 0);

  }

  function addToken(auth\_token){

    setToken(auth\_token);

  }

  function addUser(user){

    setUser(user);

  }

  return (

    <BrowserRouter>

       <><Navbar cartNum={cartNum} token = {token} loggedUser = {user} />

      <Routes>

      <Route path="/" element={<HomePage/>} />

        <Route path="/home" element={<HomePage/>} />

        <Route path="/courses" element={<Course courses={courses} onAdd={addToCart} onRemove={remFromCart} />}/>

        <Route path="teacher/courses" element={<TeacherCoursesPage courses={courses} />}/>

        <Route path="teacher/courses/course\_details" element={<TeacherCourseDetailsPage loggedUser ={user}/>}/>

        <Route path="/add\_lesson" element={<AddLessonPage/>}/>

        <Route path="/lesson/materials" element={<LessonMaterialsPage loggedUser = {user}/>}/>

        <Route path="user/courses/course\_details" element={<TeacherCourseDetailsPage  />}/>

        <Route path="/login" element={<Login addToken={addToken} addUser = {addUser} />} />

        <Route path="/register" element={<Register />} />

        <Route path="/cart" element={<Cart cartCourses={cartCourses} calculateTotal={calculateTotal} />} />

        <Route path="/checkout" element={<CheckoutPage cartCourses={cartCourses} loggedUser={user} userCourses = {loggedUserCourses} updateUserCourses={updateCourses} />} />

        <Route path="/admin/login" element={<AdminLogin addToken={addToken} addUser={addUser} />} />

        <Route path="/admin/users" element={<Users  />}/>

        <Route path="/admin/teachers" element={<Predavaci  />}/>

        <Route path="/admin/teachers/teacher\_details" element={<TeacherDetails  />}/>

        <Route path="/admin/register" element={<AdminRegister  />}/>

        <Route path="/admin/users/user\_details" element={<UserDetails  />}/>

        <Route path="/teacher/add\_course" element={<AddCoursePage updateCourses={updateCoursesAfterAdding} />}/>

        <Route path="/user/profile" element={<UserProfile loggedUser={user}/>} />

        <Route path="/user/courses" element={<UserCourses loggedUser={user} courses={loggedUserCourses} />} />

      </Routes>

      <Footer /></>

    </BrowserRouter>

  );

}

export default App;

**Komponenta App**:

Ovo je glavna komponenta aplikacije. Koristi useState za upravljanje stanjem:

* **cartNum**: broj kurseva u korpi.
* **cartCourses**: niz kurseva koji su dodati u korpu.

Navedena stanja koriste funkcije kao sto su **addToCart**, **remFromCart, refreshCart**, I **calculateTotal** da bi upravljali korpom I menjali vrednosti promenljivih.

U return delu svake komponente se nalazi kod koji se renderuje. U **App.js** return-u nalazi se **BrowserRouter** (uvezen iz **react-router-doma**) za definisanje ruta u aplikaciji. Unutar BrowserRoutera su komponente **NavBar** i **Footer**, koje se renderuju na svakoj putanji. Komponente koje su okruzene **Routes** tagom predstavljaju putanju do svake stranice u aplikaciji(**HomePage, Course, Login, Register,Cart,CheckoutPage**).

U ovoj komponenti su definisane funkcije i zahtevi koji se prosledjuju drugim komponentama.

**Components**

U React-u komponente su dinamičke i imaju svoje stanje. Stanje je ustvari jedan objekat koji određuje kako se komponenta renderuje i ponaša. To je samostalna, ponovo upotrebljiva jedinica koda koja predstavlja deo korisničkog interfejsa.

Komponente koje su definisane u našoj aplikaciji su:

* NavBar
* Footer
* CourseCard
* CartCourse

**NavBar**

import React from 'react'

import { Link } from "react-router-dom";

import "../../index.css";

import { FaCartShopping } from "react-icons/fa6";

import "./navbar.css";

const Navbar = ({cartNum}) => {

  return (

    <div className = 'container'>

    <nav>

        <ul className='row'>

            <p className='col-sm-offset-1' ><Link to ="home">Pocetna</Link></p>

            <p className='col-sm-offset-1'><Link to ="courses">Kursevi</Link></p>

            <p className='col-sm-offset-3'><Link to ="login">Login</Link></p>

            <p className='col-sm-offset-1'><Link to ="register">Registruj se</Link></p>

            <p className='col-sm-offset-1'><Link to = "cart"><FaCartShopping />{cartNum}</Link></p>

        </ul>

    </nav>

</div>

  )

}

export default Navbar

Komponenta **Navbar** je funkcionalna komponenta koja prihvata props, u ovom slučaju **cartNum**, koji predstavlja broj kurseva ili proizvoda u korisničkoj korpi. cartNum je prosledjen iz **App.js.**

Navigacioni Linkovi: Svaki **p** element sadrži **Link**(uvezen iz **react-router-doma**) koji vodi do različitih ruta aplikacije:

* **Pocetna**: vodi na početnu stranicu.
* **Kursevi**: vodi na stranicu sa kursevima.
* **Login**: vodi na stranicu za prijavu.
* **Registruj** se: vodi na stranicu za registraciju.
* **Korpa**: poslednji link sadrži ikonu korpe i dinamički prikazuje cartNum, što je broj proizvoda u korpi.

**Footer**

import React from 'react';

import { Link } from "react-router-dom";

import "../../index.css";

import { FaInfoCircle, FaPhone, FaLock, FaFileAlt } from "react-icons/fa";

const Footer = () => {

  return (

    <footer className="container footer">

      <div className="row">

        <ul className="row">

          <li className="col-sm"><Link to="/o-nama"><FaInfoCircle /> O nama</Link></li>

          <li className="col-sm"><Link to="/kontakt"><FaPhone /> Kontakt</Link></li>

          <li className="col-sm"><Link to="/politika-privatnosti"><FaLock /> Politika privatnosti</Link></li>

          <li className="col-sm"><Link to="/uslovi-koriscenja"><FaFileAlt /> Uslovi korišćenja</Link></li>

        </ul>

      </div>

      <div className="row">

        <p className="col-sm text-center">© Projektni ITEH</p>

      </div>

    </footer>

  );

};

export default Footer;

Komponenta **Footer** služi kao donji deo stranice koji sadrži navigacione linkove ka važnim informacijama i dodatne podatke o aplikaciji. Ikone iz react-icons pomažu korisnicima u bržem prepoznavanju svakog linka.

Navigacioni linkovi:

* **O nama**: vodi korisnike na stranicu sa informacijama o aplikaciji ili timu.
* **Kontakt**: vodi na stranicu sa kontakt informacijama.
* **Politika** **privatnosti**: vodi na stranicu sa politikom privatnosti.
* **Uslovi** **korišćenja**: vodi na stranicu sa uslovima korišćenja.

*(napomena: navigacioni linkovi u footeru nisu implementirani radi jednostavnosti aplikacije i stoje samo kao place-holderi)*

**CourseCard**

import React from 'react';

import "./coursecard.css";

import { FaPlus } from "react-icons/fa";

import { FaMinus } from "react-icons/fa6";

const Coursecard = ({ course, onAdd, onRemove }) => {

  return (

    <div className="card small">

      <div className='section'>

        <h2>{course.title}</h2>

      </div>

      <div className='section'>

        <img src={course.picture} className="section media" alt='picture' />

      </div>

      <div className='section'>

        <p>{course.description}</p>

      </div>

      <div className='section row'>

        <p>Dodaj u korpu</p>

        <div className="button-group">

          <button className="small" onClick={() => onAdd(course.id)}><FaPlus /></button>

          <button className="small" onClick={() => onRemove(course.id)}><FaMinus /></button>

        </div>

      </div>

    </div>

  );

}

export default Coursecard;

**Coursecard** je funkcionalna komponenta koja prima props(prosledjene iz **App.js**):

* **course**: objekat koji sadrži informacije o pojedinačnom kursu (naslov, opis, slika).
* **onAdd** i **onRemove**: funkcije koje omogućavaju dodavanje i uklanjanje kursa iz korpe.

Struktura **Coursecard** komponente prikazuje detalje o kursu i funkcionalnosti za upravljanje dodavanjem i uklanjanjem kurseva iz korpe.

Dodavanje i Uklanjanje kurseva:

* **Dodaj u korpu**: Dugme sa **FaPlus** ikonom koje poziva **onAdd** funkciju sa **course.id** kada se pritisne, čime se kurs dodaje u korpu.
* **Ukloni iz korpe**: Dugme sa **FaMinus** ikonom koje poziva **onRemove** funkciju sa **course.id**, čime se kurs uklanja iz korpe.

**cartCourse**

import React from 'react';

const cartCourse = ({ course }) => {

  return (

    <div className="card small">

      <div className='section'>

        <h2>{course.title}</h2>

      </div>

      <div className='section'>

        <img src={course.picture} className="section media" alt='picture' />

      </div>

      <div className='section'>

        <p>{course.description}</p>

      </div>

      <div className='section row'>

      <h3>Amount: {course.amount}</h3>

      </div>

    </div>

  );

}

export default cartCourse;

Komponenta slicna kao CourseCard, samo modifikovana za prikazivanje unutar trenutne korpe. Izbacen je deo u kome moze da se dodaje i izbacuje kurs iz korpe, i na tom mestu stavljena kolicina kursa(ammount) u korpi.

cartCourse je funkcionalna komponenta koja prima props:

* **course**: objekat koji sadrži informacije o kursu (naslov, opis, slika, količina), koji je dodat u korpu.

**Pages**

Stranice (pages) u React aplikacijama su komponente višeg nivoa koje predstavljaju glavne segmente aplikacije – poput početne strane, stranice sa kursevima, stranice za prijavu ili korpe.

**Homepage**

import React from 'react';

import { Link } from 'react-router-dom';

import './homePage.css';

const Homepage = () => {

  return (

    <div className="hero">

      <div className="hero-content">

        <h1>Dobrodošli na našu platformu!</h1>

        <p>Otvorite vrata novim znanjima i veštinama kroz širok spektar kurseva i programa.</p>

        <div className="button-group">

          <Link to="/courses" className="button primary">Pogledajte Kurseve</Link>

          <Link to="/register" className="button secondary">Registrujte se</Link>

        </div>

      </div>

    </div>

  );

};

export default Homepage;

Ova stranica služi kao dobrodošlica i nudi korisnicima jednostavne opcije za pregled kurseva ili registraciju.

button-group: Sadrži **Link** elemente koji vode korisnika ka stranici **kurseva** (/courses) i stranici za **registraciju** (/register).

LoginPage

import React from 'react'

import './login.css'

const Login = () => {

  return (

    <div className='login\_\_container'>

      <h2 className='col-sm-offset-5'>Login</h2>

      <form className='row'>

        <div className='col-sm-offset-4'>

          <label htmlFor="username">Username:</label>

          <input type="text" id="username" name="username" />

        </div>

        <div className='col-sm-offset-4'>

          <label htmlFor="password">Password:</label>

          <input type="password" id="password" name="password" />

        </div>

        <div className='col-sm-offset-5'>

        <button className='primary' type="submit">Login</button>

        </div>

      </form>

    </div>

  )

}

export default Login

RegisterPage

import React from 'react';

import "./register.css";

const RegisterForm = () => {

  const handleSubmit = (event) => {

    event.preventDefault();

    const formData = new FormData(event.target);

    const firstName = formData.get('firstName');

    const lastName = formData.get('lastName');

    const password = formData.get('password');

    const email = formData.get('email');

    const role = formData.get('role');

    console.log('First Name:', firstName);

    console.log('Last Name:', lastName);

    console.log('Password:', password);

    console.log('Email:', email);

    console.log('Role:', role);

  };

  return (

  <div className='register\_\_container'>

    <div className='container'>

      <h2 className='col-sm-offset-4'>Register</h2>

      <form onSubmit={handleSubmit}>

        <div className='col-sm-offset-2' >

          <label htmlFor="firstName">First Name:</label>

          <input type="text" id="firstName" name="firstName" required />

        </div>

        <div className='col-sm-offset-2'>

          <label htmlFor="lastName">Last Name:</label>

          <input type="text" id="lastName" name="lastName" required />

        </div>

        <div className='col-sm-offset-2'>

          <label htmlFor="password">Password:</label>

          <input type="password" id="password" name="password" required />

        </div>

        <div className='col-sm-offset-2'>

          <label htmlFor="email">Email:</label>

          <input type="email" id="email" name="email" required />

        </div>

        <div className='col-sm-offset-2'>

          <label htmlFor="role">Role:</label>

          <select id="role" name="role" required>

            <option value="">Select Role</option>

            <option value="Administrator">Administrator</option>

            <option value="Predavač">Predavač</option>

            <option value="Korisnik">Korisnik</option>

          </select>

        </div>

        <div className='col-sm-offset-3'>

        <button className='primary' type="submit">Register</button>

        </div>

      </form>

    </div>

  </div>

  );

};

export default RegisterForm;

**RegisterPage** omogućava korisnicima da unesu svoje podatke i izaberu ulogu na platformi

**CoursesPage**

import React, { useState } from 'react';

import Coursecard from '../../components/CourseCard/Coursecard';

const Course = ({ products, onAdd, onRemove }) => {

  const [searchTerm, setSearchTerm] = useState("");

  // Funkcija za filtriranje kurseva po imenu

  const filteredCourses = products.filter((product) =>

    product.title.toLowerCase().includes(searchTerm.toLowerCase())

  );

  return (

    <div className="container">

      <div className="row" style={{ justifyContent: 'flex-end', marginBottom: '20px' }}>

        <input

          type="text"

          placeholder="Search courses"

          value={searchTerm}

          onChange={(e) => setSearchTerm(e.target.value)}

          className="input"

        />

      </div>

      <div className="row">

        {filteredCourses.map((product) => (

          <Coursecard

            key={product.id}

            course={product}

            onAdd={onAdd}

            onRemove={onRemove}

          />

        ))}

      </div>

    </div>

  );

};

export default Course;

**CoursePage** stranica omogućava korisnicima da pregledaju kurseve sa opcijom pretrage i dodavanja ili uklanjanja kurseva iz korpe.

Prop-ovi **products, onAdd, i onRemove** (primljeni iz **App.js**):

* **products**: Lista kurseva koja dolazi kao prop i koristi se za prikaz dostupnih kurseva.
* **onAdd** i **onRemove**: Funkcije koje se prenose kroz prop-ove i omogućavaju dodavanje i uklanjanje kurseva.

Stanje **searchTerm**:

* **searchTerm** je stanje za praćenje unosa u polju za pretragu, koje inicijalno ima vrednost praznog stringa.
* **setSearchTerm** ažurira stanje na osnovu unosa korisnika u polju za pretragu.

Prikazuje se polje za pretragu gde korisnici mogu uneti ključne reči za filtriranje kurseva.

Prikaz kurseva funkcioniše tako što se prolazi kroz listu filtriranih kurseva i generiše se komponenta CourseCard za svaki kurs u listi. CourseCard se prosledjuju svi potrebni parametri

**Cart**

import React from 'react';

import CartCourse from '../../components/cartCourse/cartCourse.jsx';

import "./cart.css";

import { Link } from 'react-router-dom';

const Cart = ({ cartCourses, calculateTotal }) => {

  return (

    <div className="container cart-container">

      <h1>Your Cart</h1>

      {cartCourses.length > 0 ? (

        <div className="row cards">

          {cartCourses.map((course) => (

            <CartCourse key={course.id} course={course} />

          ))}

        </div>

      ) : (

        <p className="empty-cart">Your cart is empty.</p>

      )}

       <Link to="/checkout" className="button primary checkout-button" state={{ total: calculateTotal() }}>

            Checkout

          </Link>

    </div>

  );

};

export default Cart;

Ova Cart stranica prikazuje kurseve koje je korisnik dodao u svoju korpu i omogućava mu da pređe na checkout.

Prop-ovi **cartCourses** i **calculateTotal**:

* **cartCourses**: Niz kurseva koji su dodati u korpu, prosleđen kao prop.
* **calculateTotal**: Funkcija za proračun ukupne cene svih kurseva u korpi.

Proverava se da li korpa sadrži kurseve. Ako **cartCourses** ima elemente, prikazuju se kursevi pomoću **CartCourse** komponente. Svaki kurs u korpi dobija svoj jedinstveni ključ (key).

Dugme za prelazak na checkout stranicu koristi **Link** iz **react-router-dom**. state prop prenosi izračunatu ukupnu cenu (total: **calculateTotal()**) na **checkout** stranicu.

**CheckoutPage**

import React from "react";

import { PayPalScriptProvider, PayPalButtons } from "@paypal/react-paypal-js";

import { useLocation } from "react-router-dom";

const CheckoutPage = () => {

    const location = useLocation();

  const { total } = location.state || {};

  const clientId = "AXGedky7tPB3jDTjJtikmCviDA2wzXWC1lAjwOELO-oKBgsFD5hChc7tdFUnSGAuhFFtP6xVzcDtTSnL";

  return (

    <div className="checkout-container">

      <h1>Checkout Page</h1>

      <PayPalScriptProvider options={{ "client-id": clientId }}>

        <PayPalButtons

          style={{ layout: "vertical" }}

          createOrder={(data, actions) => {

            return actions.order.create({

              purchase\_units: [

                {

                  amount: {

                    value: total,

                  },

                },

              ],

            });

          }}

          onApprove={(data, actions) => {

            return actions.order.capture().then((details) => {

              alert(`Transaction completed by ${details.payer.name.given\_name}`);

            });

          }}

        />

      </PayPalScriptProvider>

    </div>

  );

};

export default CheckoutPage;

Ova **CheckoutPage** komponenta omogućava korisnicima da obave plaćanje putem **PayPal-a** koristeći PayPal-ov **react-paypal-js** SDK.

**useLocation** se koristi za pristup state-u prenetom putem **Link** komponente na **Cart** stranici. **total** predstavlja ukupnu cenu kurseva u korpi.

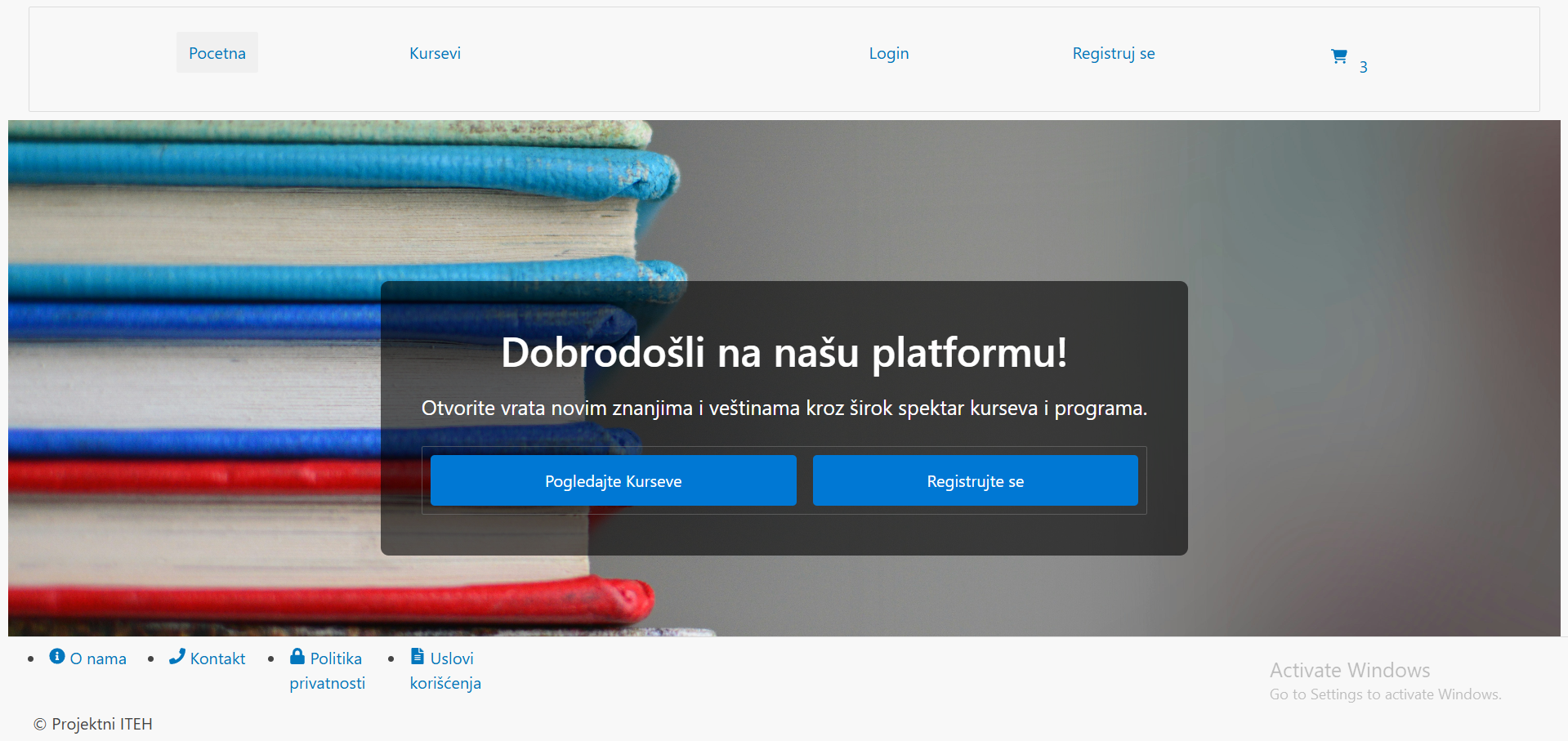
**clientId** je jedinstveni identifikator aplikacije na PayPal platformi koji omogućava autentifikaciju i povezivanje sa PayPal računom za plaćanje.Identifikator je preuzet sa <https://developer.paypal.com>.

**createOrder**: Kreira PayPal porudžbinu sa ukupnim iznosom (**total**), specificiranim u **purchase\_units**.

**onApprove**: Nakon odobrenja transakcije, koristi capture metodu za završetak plaćanja i prikazuje alert sa porukom o uspešnoj transakciji.

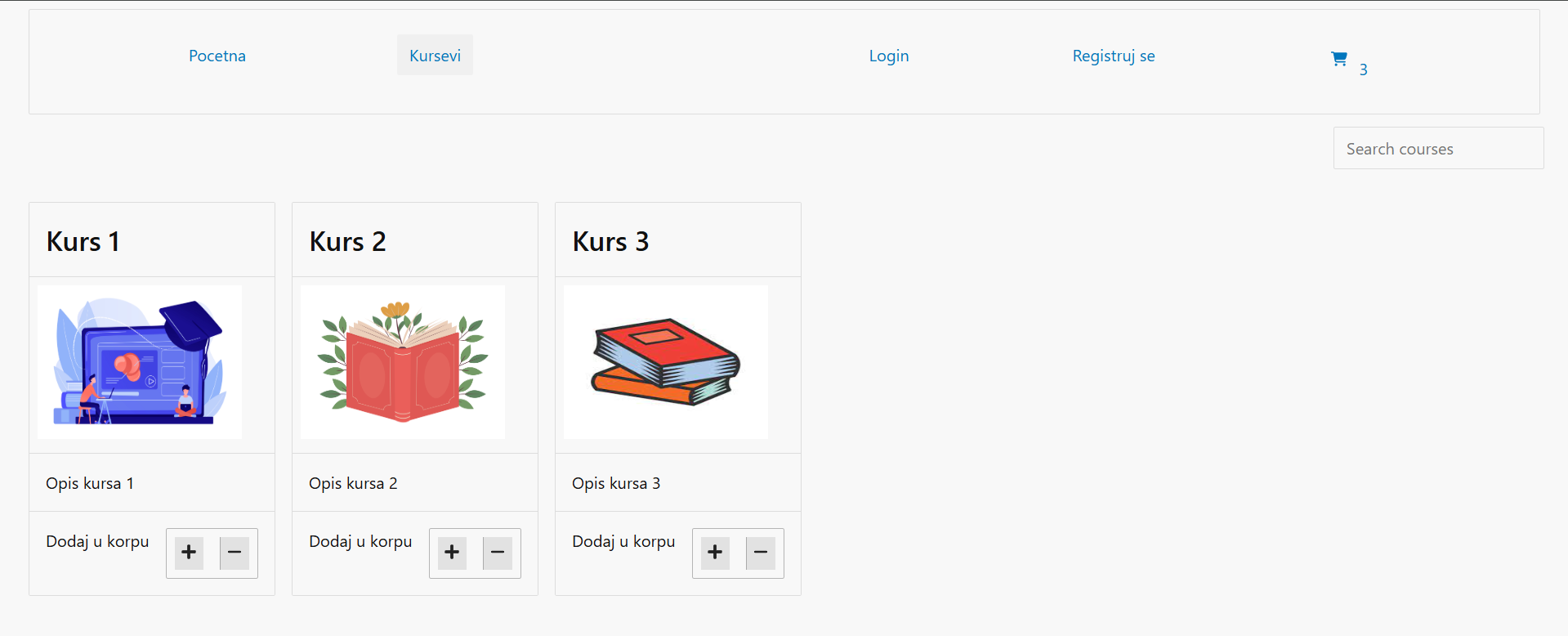
Svaka od gorenavedenih komponenti I stranica poseduje I svoj custom .css fajl za dodatno stilizonvanje ukoliko je potrebno.

## Korisničko uputstvo



1. /home

Početna stranica je centralna tačka aplikacije gde korisnici dobijaju uvodni pregled platforme i mogućnost da se registruju ili istraže kurseve

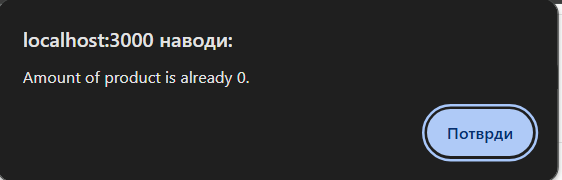


. /courses

Stranica sa kursevima omogućava korisnicima da pretražuju i pregledaju sve dostupne kurseve. Glavne funkcionalnosti uključuju:

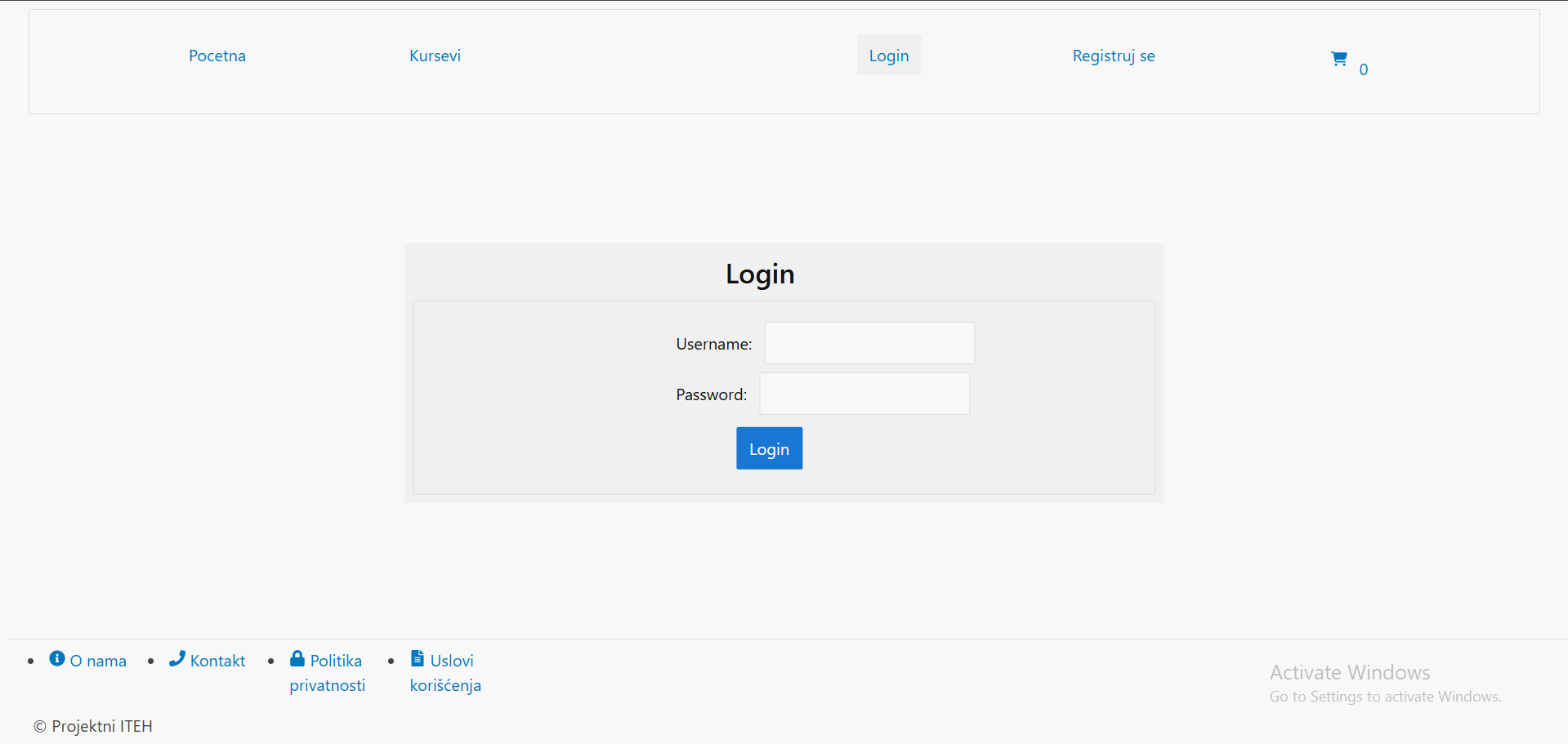
* Polje za pretragu kurseva prema nazivu.
* Prikaz svake kartice kursa sa informacijama poput naziva, slike, opisa i dugmićima za dodavanje ili uklanjanje iz korpe.

Ova stranica daje korisnicima mogućnost da pretražuju i upravljaju kursima koje žele da pohađaju ili dodaju u svoju korpu.



. Alert

Ukoliko je promenljiva **amount** već 0, a korisnik pokuša da izbaci taj kurs iz korpe pojavljuje se alert koji obaveštava korisnika da je količina u korpi već 0.

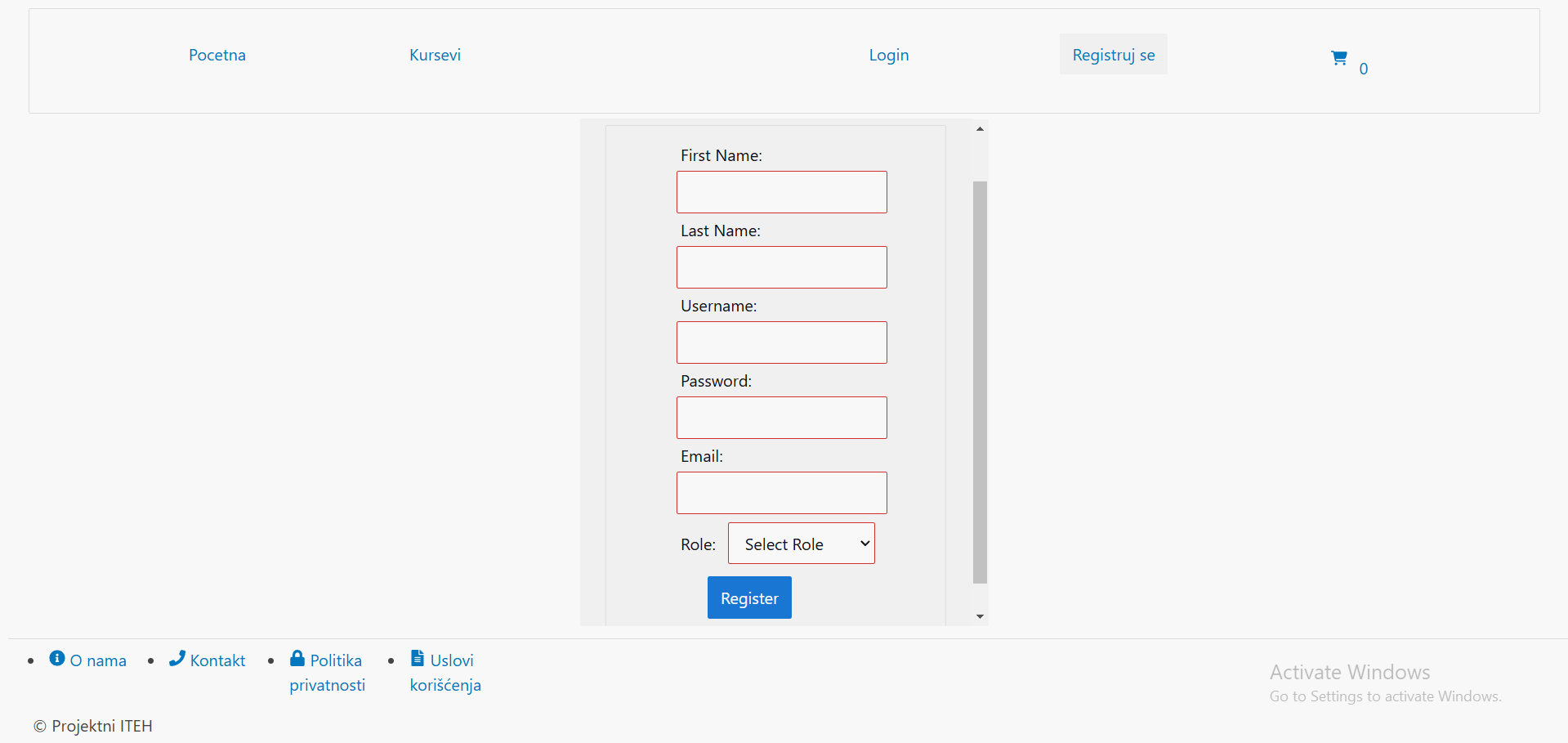
****

4. /login

Stranica za prijavu omogućava korisnicima da unesu svoje korisničko ime i lozinku za pristup svojim profilima. Sadrži osnovnu formu sa dva input polja:

* Username (korisničko ime)
* Password (lozinka)

Nakon prosledjivanja username-a I passworda u konzoli se ispisuju prikupljeni podaci.

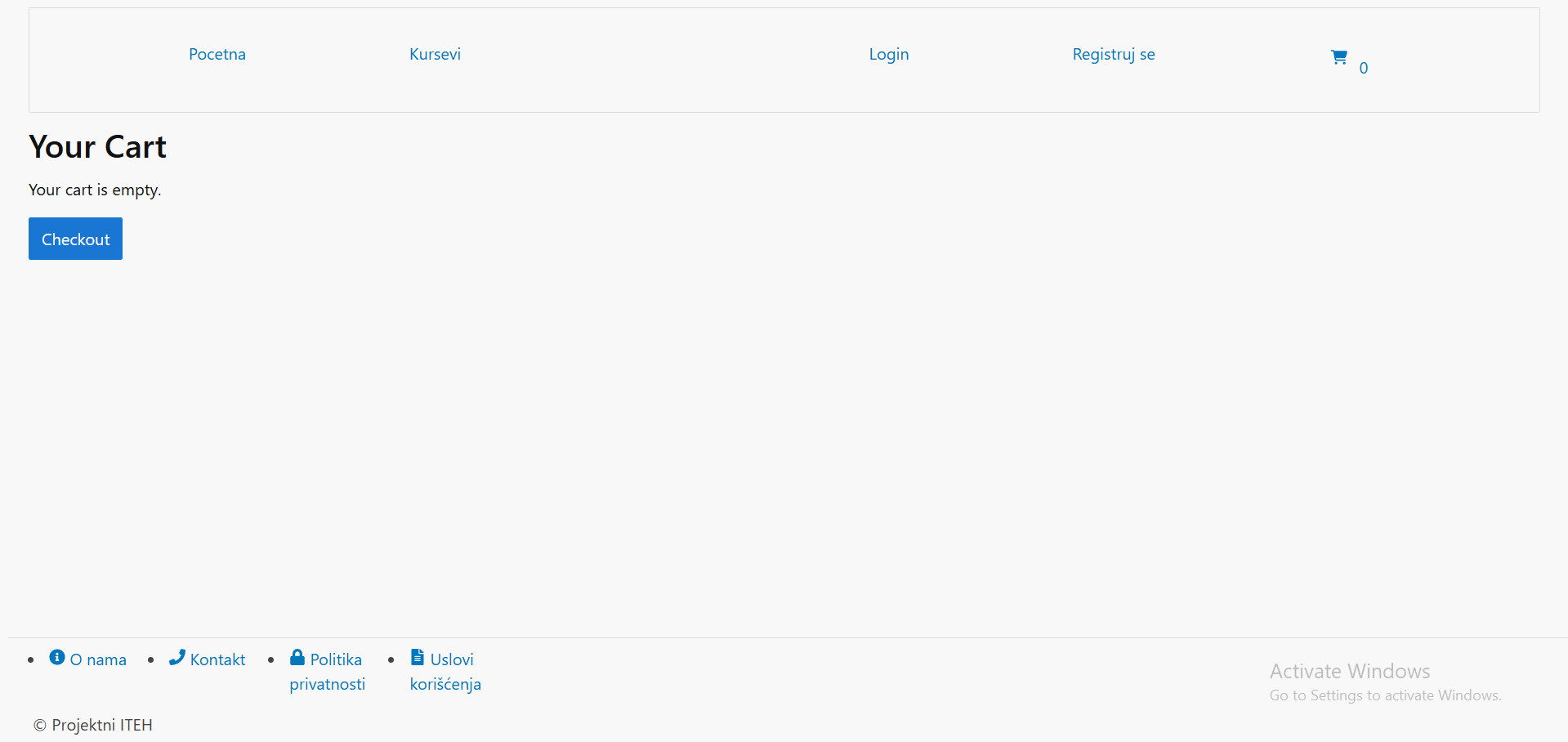


. /register

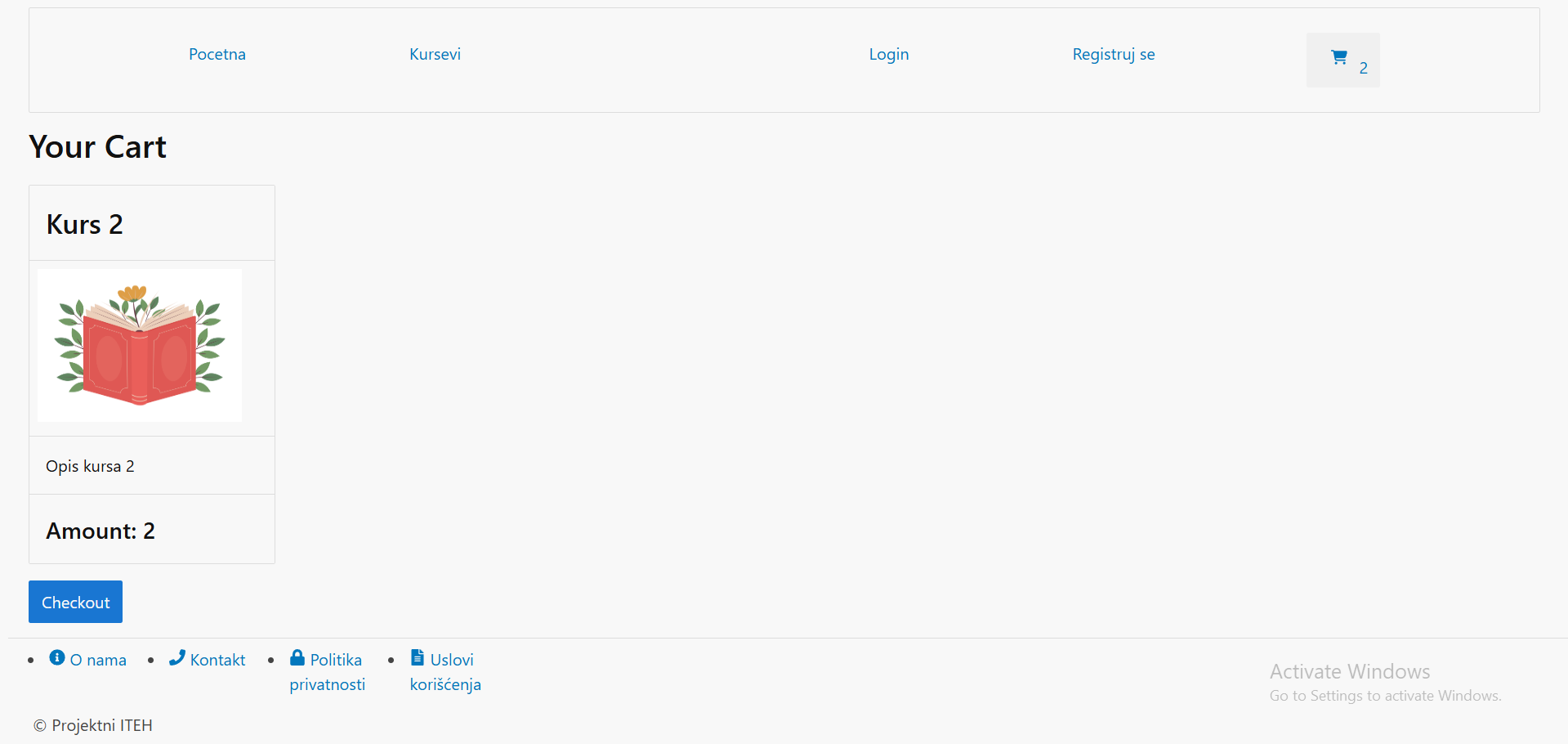
Stranica za registraciju omogućava korisnicima da kreiraju novi nalog. Sadrži formu koja zahteva sledeće podatke:

* First Name (ime)
* Last Name (prezime)
* Password (lozinka)
* Email (email adresa)
* Role (uloga) – izborna opcija gde korisnik može da odabere jednu od uloga: Administrator, Predavač ili Korisnik.

Klikom na dugme "**Register**" korisnici završavaju registraciju, a aplikacija prihvata i prikazuje unete podatke (u konzoli).



. /cart (prazna korpa)

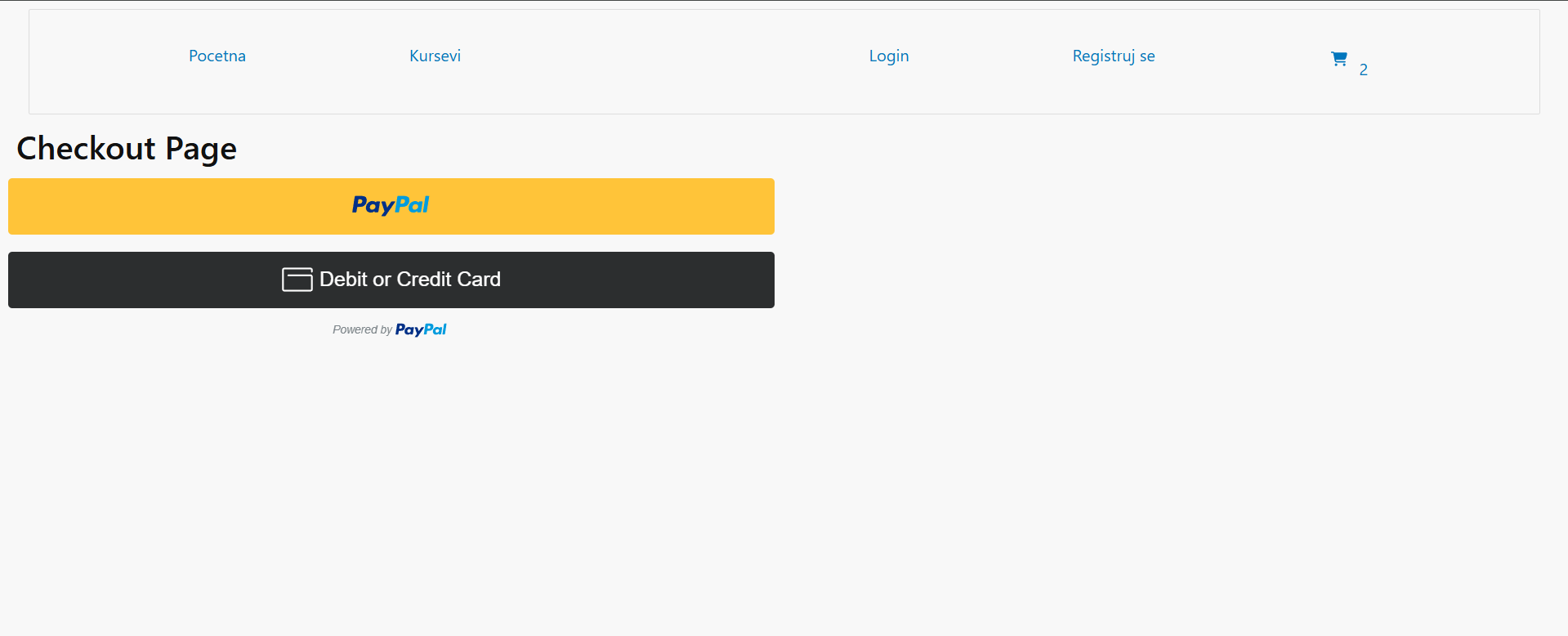


. /cart

Stranica sa korpom prikazuje sve kurseve koje je korisnik dodao u korpu. Pruža sledeće opcije:

* Prikaz svakog kursa sa nazivom, slikom, opisom i količinom.
* Dugme za Checkout koje vodi na Checkout stranicu sa ukupnim iznosom (računatim pomoću **calculateTotal()** funkcije).

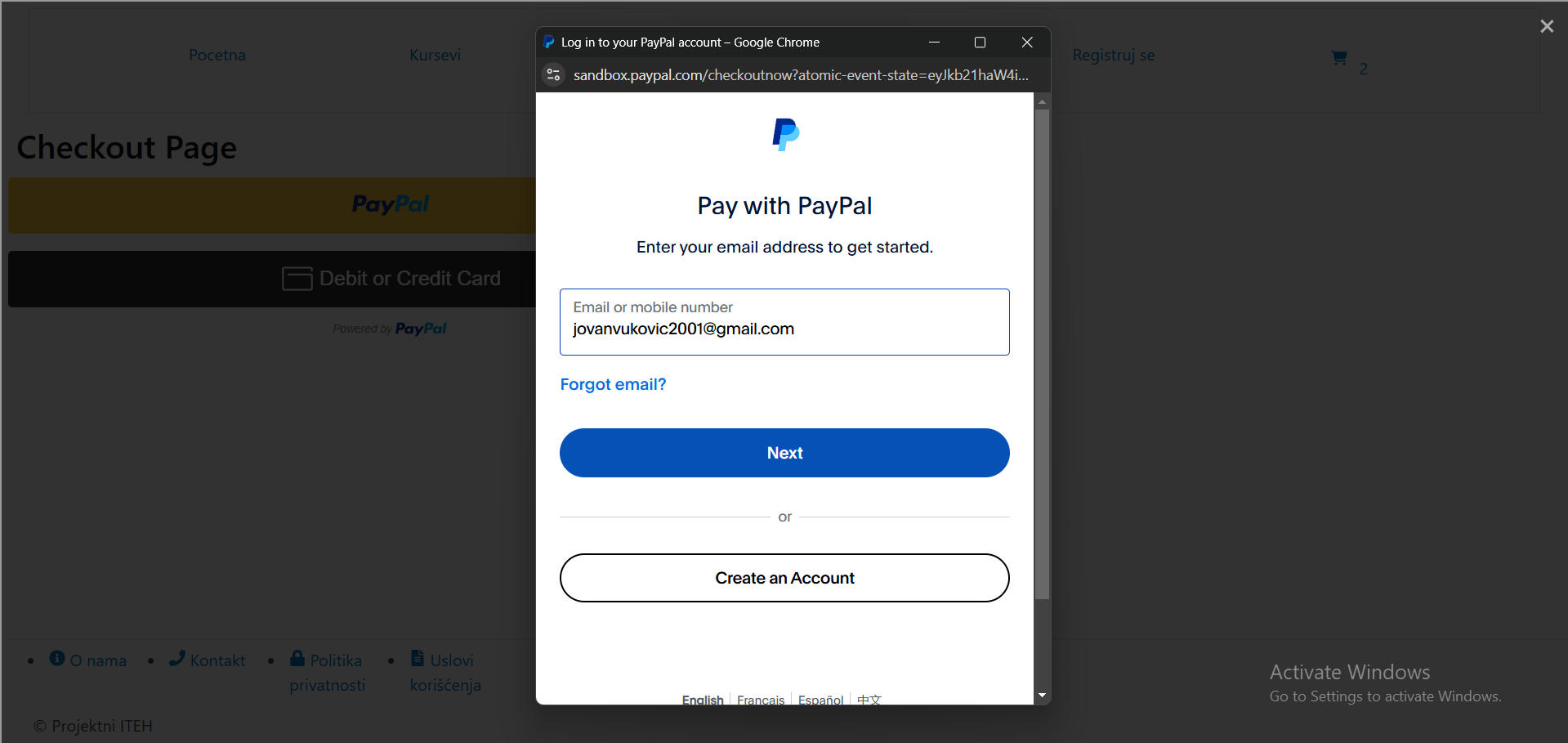
Stranica je ključna za pregled odabranih kurseva i omogućava korisnicima pregled i završnu proveru pre nego što pređu na plaćanje.



. /checkout

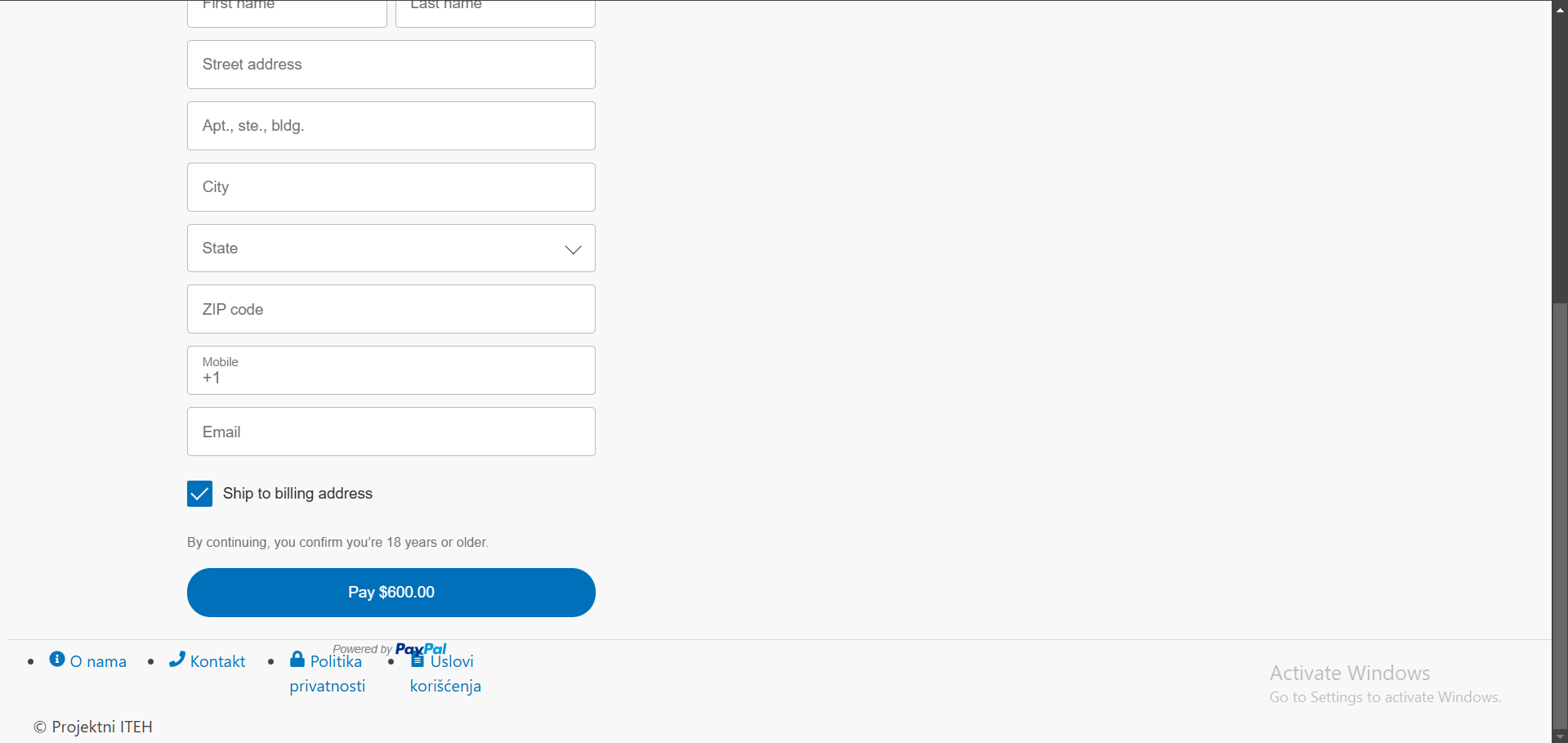
Checkout stranica koristi PayPal za obavljanje transakcija i plaćanje odabranih kurseva. Glavne karakteristike uključuju:

* PayPalButtons komponentu koja omogućava korisnicima jednostavan PayPal checkout.
* Automatizovano prikazivanje ukupnog iznosa iz korpe.



. paypal payment

Klikom na dugme “**PayPal”** otvara se login forma za paypal account, cime se zavrsava proces placanja.



. debit card payment

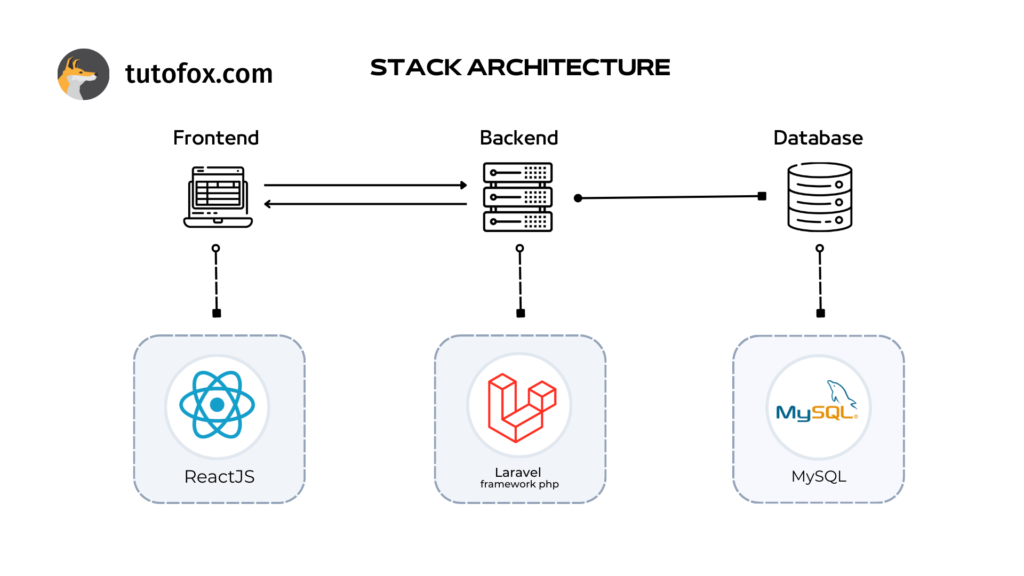
# Opis arhitekture aplikacije

Aplikacija je podeljena na nekoliko osnovnih segmenata: frontend, backend i baza podataka, koje zajedno formiraju arhitekturu sistema. Frontend je razvijen koristeći React.js i pruža korisnički interfejs putem dinamičkih komponenti koje obuhvataju funkcionalnosti poput registracije korisnika, pregleda materijala za lekcije, kao i dodavanja i preuzimanja datoteka. Za komunikaciju sa backendom koristi se Axios biblioteka, koja omogućava slanje HTTP zahteva različitih tipova, uključujući GET, POST, PUT i DELETE.

Backend aplikacije razvijen je u Laravel frameworku, koji omogućava kreiranje REST API-ja za upravljanje poslovnom logikom i rukovanje zahtevima od strane frontenda. Backend funkcioniše kao posrednik između frontenda i baze podataka, obrađuje zahteve, validira podatke i vraća odgovarajuće odgovore, obično u JSON formatu. Ključni elementi backend logike su Laravel kontroleri, modeli i rute.

Baza podataka koristi MySQL za čuvanje podataka. Struktura baze je organizovana pomoću Eloquent ORM modela, a podaci su smešteni u tabele koje prate međusobne veze korisnika, lekcija, materijala... Laravel obezbeđuje lakše upravljanje podacima kroz ORM, dok migracije definišu strukturu baze.

Komunikacija između komponenti je centralna za rad aplikacije. Frontend šalje HTTP zahteve ka backendu pomoću Axios-a za operacije poput autentikacije i manipulacije materijalima. Backend zatim koristi ORM za komunikaciju s bazom podataka, preuzima ili ažurira podatke i odgovara frontendu sa potrebnim informacijama. Na ovaj način, frontend koristi vraćene JSON podatke za dinamičko prikazivanje i ažuriranje korisničkog interfejsa.



Slika 11 - Arhitektura aplikacije