Развој апликације за руковођење научним семинаром

*Дипломски рад*

Студент Ментор

Јована Мика проф. др Драган Матић

Бања Лука, 2023.

Садржај:

[1. Увод 3](#_Toc153821049)

[2. Постојеће апликације за управљање семинарима 4](#_Toc153821050)

[3. Развој веб апликације 8](#_Toc153821051)

[3.1. Веб апликација 8](#_Toc153821052)

[3.2. Динамичке веб странице 8](#_Toc153821053)

[3.3. Front-end и Back-end развој 9](#_Toc153821054)

[3.3.1. Креирање корисничког интерфејса 9](#_Toc153821055)

[3.3.2. Развој функционалности на серверској страни 9](#_Toc153821056)

[3.4. Кориштене техологије и алати 10](#_Toc153821057)

[3.4.1. HTML 10](#_Toc153821058)

[3.4.2. CSS 10](#_Toc153821059)

[3.4.4. React JS 11](#_Toc153821060)

[3.4.4. Visual Studio Code 12](#_Toc153821061)

[3.4.5. Java 13](#_Toc153821062)

[3.4.6. Spring Boot 14](#_Toc153821063)

[3.4.7. Maven 14](#_Toc153821064)

[3.4.8. Eclipse IDE 15](#_Toc153821065)

[3.4.9. PostgreSQL 16](#_Toc153821066)

[4. Спецификација апликације 17](#_Toc153821067)

[4.1. Дефиниције улога у апликацији 18](#_Toc153821068)

[4.1.1. Администратор 18](#_Toc153821069)

[4.1.2. Гост 18](#_Toc153821070)

[4.1.3. Корисник 18](#_Toc153821071)

[4.1.4. Предавач 18](#_Toc153821072)

[4.1.5. Регистрација 18](#_Toc153821073)

[4.1.6. Пријава 19](#_Toc153821074)

[4.1.7. Приступни подаци 19](#_Toc153821075)

[4.2. Кориснички захтјеви 19](#_Toc153821076)

[5. Развој апликације 21](#_Toc153821077)

[5.1. Развој серверског дијела апликације 21](#_Toc153821078)

[5.1.1. Структура апликације 22](#_Toc153821079)

[5.1.2. Модел базе података 29](#_Toc153821080)

[5.2. Развој корисничког интерфејса 30](#_Toc153821081)

[5.2.1. Почетна страна 34](#_Toc153821082)

[5.2.2. Модул за регистрацију 36](#_Toc153821083)

[5.2.3. Модул за пријаву 37](#_Toc153821084)

[5.2.4. Модул за приказ корисничког профила 38](#_Toc153821085)

[5.2.5. Модул за додавање новог поста 40](#_Toc153821086)

[5.2.6. Модул за приказ семинара 43](#_Toc153821087)

[5.2.7. Модул за приказ новости 44](#_Toc153821088)

[5.2.8. Модул за приказ појединачног поста 45](#_Toc153821089)

[5.2.9. Администраторски панел 46](#_Toc153821090)

[6. Закључак 57](#_Toc153821091)

[7. Литература 58](#_Toc153821092)

# 1. Увод

Друштво све више препознаје значај науке и образовања као кључних елемената напретка и развоја. Научни семинари постају важан дио академске сфере, а напредак технологије омогућава нове приступе организацији и праћењу научних семинара путем апликација које пружају ефикасније алате за управљање садржајем.

Учествовање у научном семинару не само да доприноси академском развоју студената, већ такође омогућава истраживачима и професорима да дијеле своје знање и искуствo. Ипак, организација и вођење семинара могу представљати изазов, посебно у окружењима гдје се технолошки алати не користе довољно или нису прилагођени одређеним потребама ове врсте образовне активности.

Циљ овог дипломског рада је развој апликације која ће омогућити лако и ефикасно руковођење научним семинаром. Веб апликација за руковођење научним семинаром креирана је тако да корисницима омогући преглед информација везаних за семинаре који су се одржали или тек треба да се одрже, корисне објаве у области науке као и обавјештења о другим догађајима научног карактера. Поред информативног карактера, апликација омогућава и резервацију мјеста, односно пријаву на семинар. Апликација за руковођење научним семинаром, која је развијана у оквиру практичног дијела дипломског рада, представља иновативно рјешење које се истиче по својој јасноћи и ефикасности у приказу информација.

# 2. Постојеће апликације за управљање семинарима

Проучавање већ постојећих апликација са сличном или истом тематиком помаже нам да боље разумијемо основне функционалности које су важне за кориштење апликација. Анализом њихових предности и недостатака можемо доћи до идеје о томе како унаприједити постојеће апликације и створити нову која ће задовољити одређене потребе циљне групе.

За разлику од постојећих апликација, апликација која је развијана у оквиру овог рада, ограничена је само на информације које се тичу семинара и објава научног и истраживачког карактера. Ограничавање функционалности на специфичан домен, у случају ове апликације, на семинаре и научне објаве, има кључну предност у пружању јасног фокуса и олакшавању корисницима проналажење и управљање информацијама. Употреба овог приступа спречава непотребно претраживање сувишних информација.

У наставку овог поглавља ће бити приказани кориснички интерфејси двије апликације које садрже неки вид управљања научним семинаром. Оно што је заједничко овим апликацијама јесте што садрже све релевантне информације везане за рад факултета. С обзиром на количину информација које пружају и сложене структуре, проналажење одређених секција посвећених вођењу семинара може представљати изазов.

Прва апликација представљена у оквиру овог поглавља, апликација Економског факултета Универзитета у Нишу, има значајан потенцијал, али су примјетни и одређени изазови који би могли утицати на корисничко искуство (*Слика 2.1*) [1]. Посебно је изражен проблем велике количине информација које су представљене у оквиру апликације, што може довести до непрегледности приликом претраге одређених информација.

Навигациона трака се налази у заглављу апликације и садржи десет опција. Поред тога, свака од опција отвара споредни мени. Тако, на примјер, секција „Научни рад“, отвара споредни мени са многобројним опцијама, укључујући пројекте, детаље о одржавању семинара и информације о међународној сарадњи.

Оваква обимна структура може да утиче на корисникову потребу за брзим приступом конкретним информацијама, што може утицати на цјелокупни доживљај корисника.

Функционалност и употреба апликациије могли би значајно да се унаприједе када би се садржај подијелио у неколико цјелина, чиме би се и комплексност навигације смањила.

Избором одређеног научног скупа, отвара се страница са информацијама везаним за изабрани догађај (*Слика 2.2.*). На овој страници јасно су приказани сви потребни подаци као што су наслов, вријеме и мјесто одржавања семинара.

На *Слици 2.3.* приказан је кориснички интерфејс апликације Института за математику и информатику у Крагујевцу [2]. Иако је ова апликација концептуално слична претходној, примјетно је да је садржај друге апликације боље организован, што омогућава кориснику да брже приступи информацијама које га интересују. У циљу побољшања корисничког искуства, потребно је отклонити одређене недостатке, међу којима се истиче ограничење које онемогућава кориснику да прегледа садржај семинара без преузимања одређеног документа на свој уређај.



**Слика 2.1.** *Приказ одјељка који садржи информције о научним скуповима, семинарима и исраживачким радовима.*



**Слика 2.2.** *Приказ објаве везане за мјесто и вријеме одржавање семинара.*

# ScIMI.png

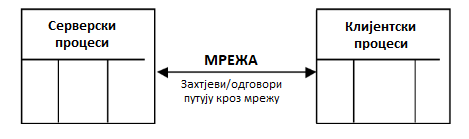
**Слика 2.3.** *Примјер апликације Института за математику и информатику.*

# 3. Развој веб апликације

У овој секцији ћемо размотрити основне појмове везане за развој апликација.

## 3.1. Веб апликација

Веб апликација је врста софтверске апликације којој се приступа и са којом се остварује интеракција путем веб претраживача [3]. Ове апликације се базирају на клијент-сервер моделу. Клијентски дио извршава се у веб прегледачу и представља особу или организацију која користи одређену услугу, док је сервер удаљени рачунар који пружа информације или омогућава приступ одређеним услугама.



***Слика 3.1.1.*** Приказ основног клијент- сервер модела

На почетку развоја интернета, већина веб страница служила је за приказ статичких података написаних уз помоћ *HTML*- а (енгл. *HyperText Markup Language*). Клијент-сервер модел у овом случају функционисао је тако што би корисници слали захтјев серверу за одређеном страницом, а сервер би им као одговор враћао предефинисану страницу која се налази у фајл систему сервера.

## 3.2. Динамичке веб странице

Развојем технологије, јавила се потреба за већом интеракцијом између корисника и сервера. Како би се ово постигло, развијене су динамичке веб репрезентације чије су странице генерисане динамички од стране сервера приликом обраде корисничког захтјева.

За разлику од статичких *HTML* страница, које представљају ресурс на серверу који је фиксан и генерисан унапријед, динамичка страница се генерише у реалном времену приликом сваког корисничког захтјева.

Динамичке странице често користе апликативне програме за различите врсте услуга и захтјевају ресурсе на серверској страни, укључујући и приступ базама података.

## 3.3. Front-end и Back-end развој

Развој веб апликације обухвата два важна аспекта: креирање корисничког интерфејса (енгл. *front-end*) и развоја функционалности на серверској страни (енгл. *back-end*).

### 3.3.1. Креирање корисничког интерфејса

Развој корисничког интерфејса обухвата креирање и имплементацију видљивих елемената сајта, са циљем да се кориснику пружи пријатно и функционално искуство. Важно је да кориснички интерфејс буде прилагођен различитим типовима уређаја и претраживача. Поред основих технологија као што су *HTML*, *CSS* и *JavaScript*, за потребе овог дипломског рада, за развој корисничког интерфејса кориштена је *React* библиотека, која ће бити детаљније описана у наставку.

### 3.3.2. Развој функционалности на серверској страни

Развој функционалности на серверској страни односи се на дио апликације који се бави логиком и обрадом података, а којем корисник не може директно да приступи. Овај дио укључује серверску страну апликације, базу података као и све процесе који се одвијају како би се омогућиле одређене функционалности корисницима и осигурао интерфејс са корисничком страном.

## 3.4. Кориштене техологије и алати

### 3.4.1. HTML

*HTML* (*HyperText Markup Language*) је мета језик за означавање који се користи за креирање структуралног оквира веб страница [6]. Састоји се од низа елемената, односно тагова, на основу којих веб прегледач приказује садржај одређеног веб документа. Неки од примјера кориштења тагова су: форматирање садржаја, приказивање табеле, уметање слике и многи други.

На *Слици 3.4.1.1.* приказана је осовна структура *HTML* документа.



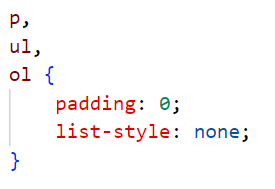
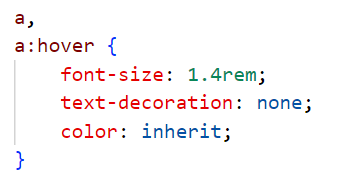
**Слика 3.4.1.1.** *Основна структура HTML документа.*

### 3.4.2. CSS

*CSS* (*Cascading Style Sheets*) је језик који се користи за дефинисање правила приказа елемената у документима као што су, на примјер, *HTML* и *XML*. *CSS* правила обично се дефинишу унутар датотеке чија је екстензија .css, .scss, или могу бити интегрисана унутар *HTML* или *XML* докумената [7].

Помоћу *CSS*-а можемо прецизно управљати изгледом елемената у прегледачу. Да бисмо стилизовали одређени елемент, наводимо назив елемента, а затим у витичастим заградама додајемо *CSS* кôд како бисмо постигли жељени изглед. Такође, правила се могу примјењивати на одређене класе додавањем тачке испред имена класе, или на одређене идентификаторе кориштењем знака „#“ испред имена идентификатора елемента.

Могуће je стилизовати више елемената истовремено тако што се називи класа, идентификатора или елемената дефинишу један за другим, одвојених зарезом (*Слика 3.4.2.1.*).

**Слика 3.4.2.2.** *Примјер постављања „hover“ догађаја на елемент.*

**Слика 3.4.2.1.** *Примјер стилизовања више елемената истовремено.*

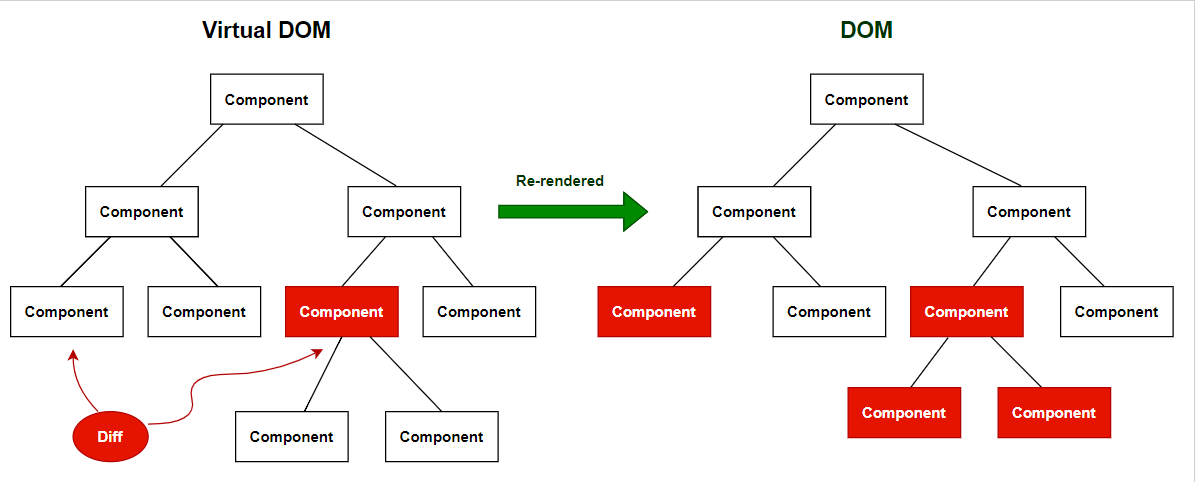
За свако правило можемо додати догађаје, као што су *„hover“*, „*focus“*, „*submit“* и други, који дефинишу када се правило примјењује на одређени елемент (*Слика 3.4.2.2.*). Уколико желимо одређени стил примјенити на све елементе то можемо урадити кориштењем ознаке „\*“ умјесто назива елемента.

### 3.4.4. React JS

*ReactJS* је *JavaScript* библиотека, која је креирана како би се изградили кориснички интерфејси који ефикасно реагују на догађаје корисничког уноса и управљање стањем апликације [10]. Главни циљ у креирању ове библиотеке био је превазилажење изазова који су повезани са развојем сложених апликација које садрже велику количину података. За разлику од других популарних окружења за развој веб апликација *React* представља самосталну библиотеку чија је основна намјена дефинисање изгледа и приказа података на екрану, док остале функционалности програмер мора да напише самостално, или уз помоћ библиотека које су компатибилне са *React* библиотеком.

*React* подржава кориштење више екстензија као што су *. js* или *.jsx*. *JSX* (*JavaScript XML*) је синтаксна екстензија која омогућава писање структура корисничког интерфејса на начин сличан *HTML*-у. Кориштење *JSX* синтаксе олакшава визуелизацију и писање *React* компоненти комбиновањем ознака сличних *HTML*-у са *JavaScript* логиком [12].

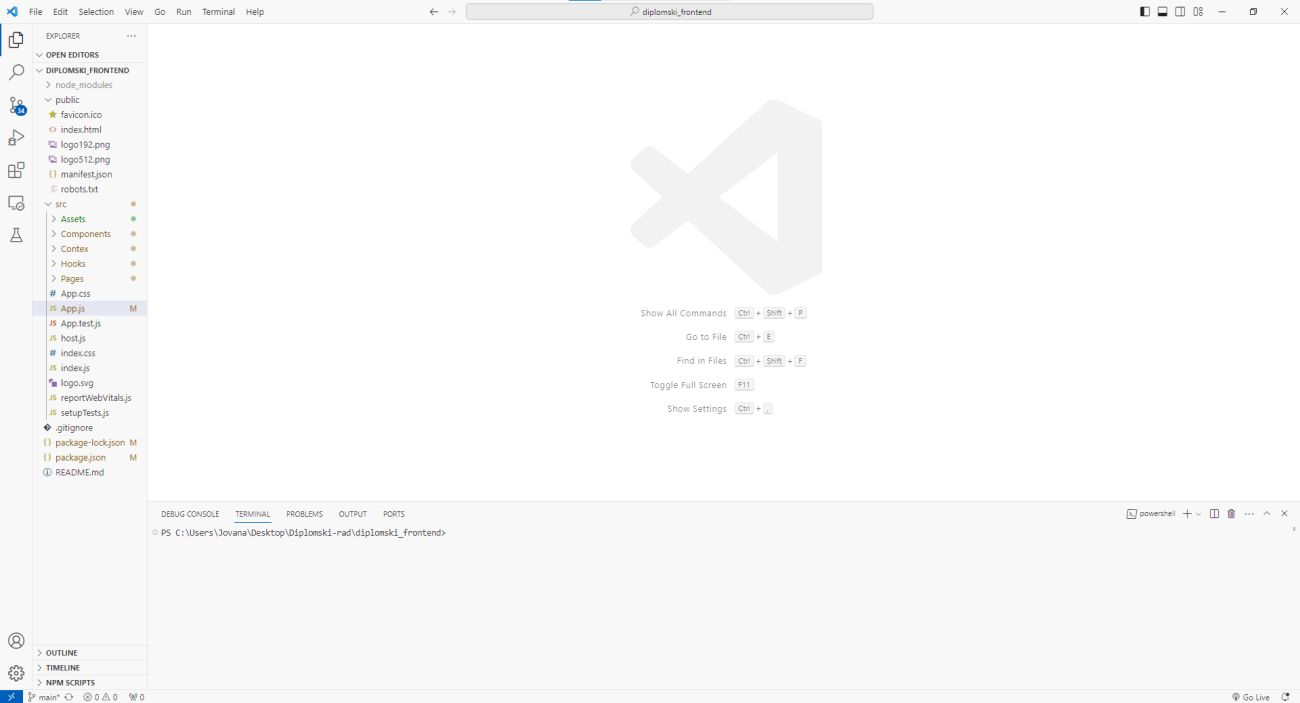
Основна карактеристика *React*-a је употреба виртуалног DOM-а ( *Слика 3.4.3.1.*).  
DOM ( *енгл. Document Object Model*) представља интерфејс за програмске апликације и омогућава програмима да читају и мијењају садржај било које веб странице. За разлику од *DOM*-a, код кога се промјене приказују тек након поновног учитавања странице, виртуални *DOM* омогућава тренутно приказивање промјена у стварном времену без потребе за поновним учитавањем.



**Слика 3.4.3.1.** *React Virtual DOM*

### 3.4.4. Visual Studio Code

*Visual studio Code*, развијен од стране Microsofta, представља уређивач изворног кода који је доступан за *Windows*, *MacOS* и *Linux* (*Слика3.4.4.1*). Долази са уграђеном подршком за *JavaScript*, *TypeScript* и *Node.js* али има богат екосистем екстензија за друге језике као што су: *C++*, *Java*, *Python* и други [14].



**Слика 3.4.4.1.** *Изглед Visual Studio Code-a.*

### 3.4.5. Java

Java је објектно оријентисан, платформски независан програмски језик [15]. Карактеристике овог програмског језика су једноставна употреба, сигурност извршавања и могућност преносивости кода. Развијен је од стране компаније *Sun Microsystems* 1995. године [15].

*Java* програмски језик користи специфичан начин компилације. Након писања изворног кода, свака класа чува се у одвојеној датотеци која има *.java* екстензију. Кориштењем *Java* компајлера, изворне датотеке се преводе у бајткод и на тај начин се генеришу *.class* датотеке. *Class* датотеке садрже инструкције за извршавање које виртуелнa машинa (*JVM*) користи приликом покретања програма.

Програмски језик Java може се користити за развој различитих врста софтвера, укључујући веб апликације, десктоп апликације, мобилне апликације, уграђене системе ( енгл. *embedded systems*) и игрице.

### 3.4.6. Spring Boot

*Spring Boot* представља оквир отвореног типа за развој апликација у програмском језику *Javа*. Уграђени веб сервери, у нашем случају *Tomcat*, омогућавају једноставно покретање *Java* апликације.

При иницијализацији пројекта, *Spring Boot* поставља иницијалну *pom.xml* датотеку која представља конфигурацију *Maven* пројекта и омогућава *Maven* алату да аутоматски управља зависностима, изградњом и експортовањем *Spring Boot* апликације. Кључни елементи *pom.xml* датотеке су [16] :

1. ***<modelVersion>*:** Означава верзију *POM* датотеке.
2. ***<parent>* :** Дефинише родитељски *POM*.
3. ***<groupId>***, ***<artifactId>***, ***<version>*:** Идентификатори пројекта.
4. ***<packaging>* :**Дефинише врсту архивирања пројекта, на примјер .*war*.
5. ***<properties>*:** Дефинише својства пројекта.
6. ***<dependencies>*:** Листа зависности које пројекат користи. У зависности спадају *Spring Boot* стартери, *Tomcat* и други, о којима ће бити ријечи у наставку рада.
7. ***<build>*** ,***<plugins>*:** Конфигурација за Maven процесе.
8. ***<exclude>*:**Искључује *Lombok* библиотеку из *Spring Boot Maven Plugin-a* како би се избјегао конфликт са генерисаним кодом.

### 3.4.7. Maven

*Apache Maven* је алат за управљање пројектима који се користи за изградњу, управљање зависностима и аутоматизацију процеса развоја софтвера [17].

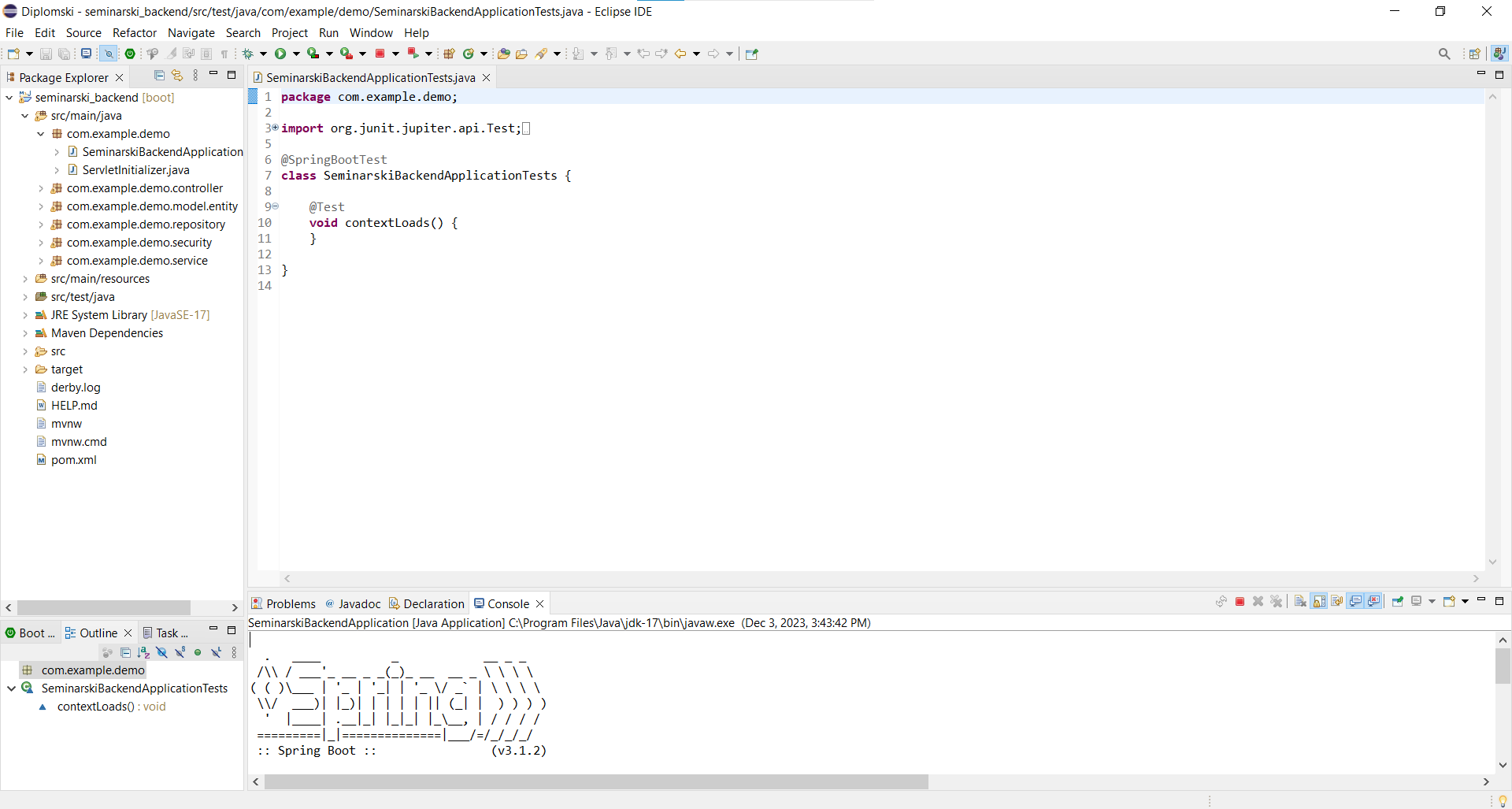
За креирање датотека, користи декларативни приступ који се ослања на конфигурацијске датотеке како би омогућио програмерима ефикасно управљање свим аспектима животног циклуса програма. Једна од основних карактеристика је употреба Објектног модела пројекта (енгл. *Project Object Model*), који дефинише све потребне зависности.

*Maven* аутоматски обавља компилацију, тестирање и друге активности које су саставни дио развоја софтвера.

### 3.4.8. Eclipse IDE

*Eclipse Integrated Development Enviroment* је популаран алат, отвореног кода, за развој софтвера (*Слика 3.4.8.1*). Развијен је у програмском језику Java, али подржава рад у различитим програмским језицима, укључујући C++, PHP, Python и друге [18].

Доступан је на различитим оперативним системима (*Windows*, *macOS*, *Linux*), што омогућава кориштење истих алата на различитим платформама. Са подршком за развој мобилних апликација, прилагодљивости додатака (енгл. *plugins*), алатима за дебаговање и разним другим могућностима*, Eclipse* пружа ефикасно окружење за развој софтвера.



**Слика 3.4.8.1.** *Изглед Eclipse развојног окружења.*

### 3.4.9. PostgreSQL

*PostgreSQL* је систем за управљање базама података објектно-релационог типа [19]. Својим објектно-релационим приступом омогућава дефинисање властитих типова података и функција.

Неке од карактеристика су висока поузданост и подршка трансакцијама, проширивост која омогућава да се база података прошири према потребама пројекта. Поред наведених карактеристика *PostgreSQL* подржава стандардни *SQL* уз могућност проширења и додавања одређених функционалности. Нуди подршку за *JSON* податке и на тај начин омогућава рад са веб апликацијама и сервисима који користе *JSON* формат за размјену података. Припада групи „open source“ софтвера, односно слободно је доступан за преузимање, кориштење и прилагођавање потребама корисника.

Примјену је нашао између осталог за развој веб апликација, пословних система, геопросторних информативних система, аналитичких база података и других.

# 4. Спецификација апликације

Веб апликација за руковођење научним семинаром намијењена је свим корисницима који желе да се пријаве на семинар, прате дешавања у свијету науке или активно учествују у креирању истих.

Апликацију можемо подијелити на три дијела. Први дио апликације, који не захтијева пријаву у систем, омогућава корисницима да истраже актуелности у свијету науке кроз преглед новости и да добију увид у предстојеће или претходно одржане семинаре.

Други дио, који захтијева регистрацију корисника, представља интерактивни дио апликације. Овдје корисници могу да изврше пријаву на семинар као и да пошаљу захтјев за објављивање нових постова, у случају да су пријављени као предавачи. Пријављени корисници такође имају могућност прегледа и ажурирања својих приступних података на личном профилу, што обезбјеђује персонализовано корисничко искуство.

Трећи дио, односно администраторски панел, намијењен је искључиво администраторима апликације, који имају посебна овлаштења за управљањем апликације. Администратори имају право прегледа свих захтјева за објављивање новог поста, могућност да исти одобре или избришу. Поред тога, имају увид у све објаве, семинаре и кориснике, као могућност брисања, измјене и додавање нових. Систем омогућава и постојање више од једног администратора, што обезбјеђује колаборативни приступ.

Апликација садржи три типа регистрованих корисника: администратор, предавач и корисник. Поред регистрованих корисника, апликацију може да користи и „гост“, односно, нерегистровани корисник. Апликација усмјерава сваку групу корисника кроз персонализоване функционалности и на тај начин се прилагођава различитим потребама корисника.

У наредном поглављу, размотрићемо кључне улоге и активности унутар веб апликације за управљање научним семинаром. Свака од дефинисаних улога има специфичне одговорности и привилегије које доприносе функционалности и динамици цјелокупног система.

## 4.1. Дефиниције улога у апликацији

4.1.1. Администратор

Администратор је особа која има кључну улогу у управљању апликацијом. Има овлаштења за додавање нових корисника, као и брисање и измјену већ постојећих. Такође, администратор управља објављивањем постова, семинара, њиховим брисањем или измјеном. У бази података може бити регистровано више од једног администратора.

### 4.1.2. Гост

Гост је особа која може да прегледа садржај веб апликације иако није пријављена, што му обезбјеђује преглед информација о постовима и семинарима.

### 4.1.3. Корисник

Корисник представља особу која се пријављује у апликацију, чиме се отварају додатне функционалности, од којих је најважнија могућност пријаве на семинар.

4.1.4. Предавач

Предавачи су регистровани корисници који могу да подносе захтјеве за објављивање постова. Ова функционалност омогућава предавачима да активно учествују у стварању садржаја који се приказује у апликацији.

4.1.5. Регистрација

Процес регистрације представља активност коју корисник извршава уношењем скупа личних података који се просљеђују систему. По успјешном регистровању, корисник може да изврши пријаву на апликацију. Поред тога, предавачима су омогућене и функционалности обичног корисника, односно, предавачи такође могу да се пријављују на семинаре других предавача.

4.1.6. Пријава

Процес пријаве представља активност коју корисник извршава попуњавањем одговарајуће форме како би потврдио свој идентитет и приступио садржају веб апликације намијењеном регистрованим корисницима.

4.1.7. Приступни подаци

Приступни подаци, који укључују електронску адресу и лозинку, представљају кључне информације помоћу којих се корисник пријављује на апликацију. Ови подаци осигуравају безбједан приступ личном корисничком профилу на апликацији.

## 4.2. Кориснички захтјеви

У наставку је приказана табела која систематизује корисничке захтјеве за апликацију укључујући улоге администратора, предавача, корисника и госта.

Кориснички захтјеви представљају формално дефинисане спецификације функционалности које корисници очекују од система.

## 

***Табела 4.2.1.*** *Приказ учесника и њихових захтјева који се извршавају унутар система.*

|  |  |
| --- | --- |
| **Учесник** | **Захтјев** |
| Администратор | * Пријава * Преглед свих корисника   + Унос новог корисика   + Брисање корисика   + Измјена корисника * Преглед свих семинара   + Унос новог семинара   + Преглед семинара   + Измјена семинара   + Брисање семинара * Преглед свих захтјева за додавање објаве   + Одобравање и објава   + Одбацивање захтјева * Преглед свих објава   + Унос нове објаве   + Измјена објаве   + Брисање објаве * Одјава |
| Предавач | * Пријава * Преглед свих семинара   + Избор појединачног семинара     - Пријава на семинар     - Поништавање пријаве на семинар * Преглед свих објава   + Преглед појединачне објаве * Слање захтјева администратору за објављивање новог поста * Преглед корисничких података на профилу * Измјена корисничких података * Oдјава |
| Корисник | * Пријава   + Преглед свих семинара     - Избор појединачног семинара       * Пријава на семинар       * Поништавање пријаве на семинар   + Преглед свих објава     - Преглед појединачне објаве * Преглед корисничких података * Измјена корисничких података * Одјава |
| Гост | * Преглед садржаја веб странице |

# 5. Развој апликације

У оквиру практичног дијела дипломског рада развијана је апликација за руковођење научним семинаром. Развој апликације може да се подијели на два дијела:

1. Развој серверског дијела апликације (енг. Back-end)
2. Развој корисничког интерфејса (енрг. Front-end)

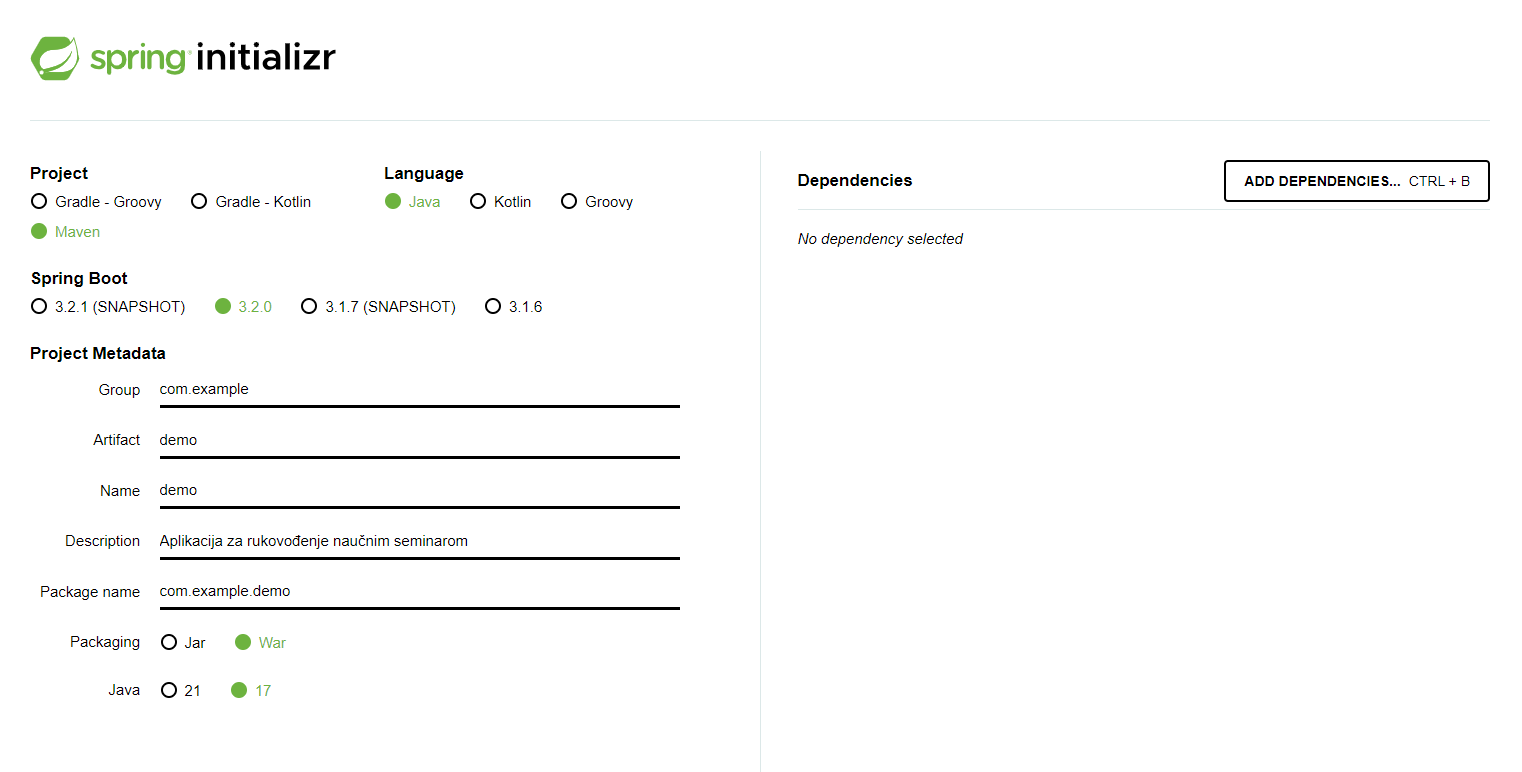
## 5.1. Развој серверског дијела апликације

Серверски дио апликације писан је у *Јava* програмском језику, кориштењем *Spring* развојног оквира.

Први корак у креирању *Spring Boot* апликације је инсталирање следећих алата:

* *Java SE Development Kit (JDK) 17*
* *Maven 3.2+*
* *Eclipse IDE развојно окружење*

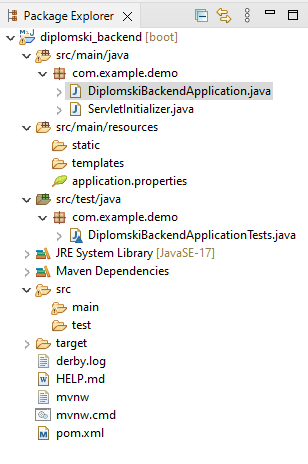
Помоћу *Spring Initializr*-a креиран је *Maven* пројекат (*Слика 5.1.1.*).



**Слика 5.1.1.** *Spring Initializr кориснички интерфејс.*

Генерисани пројекат садржи неколико директоријума и датотека (*Слика 5.1.2.)*:

* ***pom.xml*** - Садржи *Maven build* спецификацију.
* ***application.properties*** - Конфигурацијска датотека.
* ***DiplomskiBackendApplication.java*** - Класа која садржи *main()* методy и служи за покретање апликације.
* ***DiplomskiBackendApplicationTests.java*** – Тест класа која учитава контекст *Spring* апликације помоћу аутоматске конфигурације.

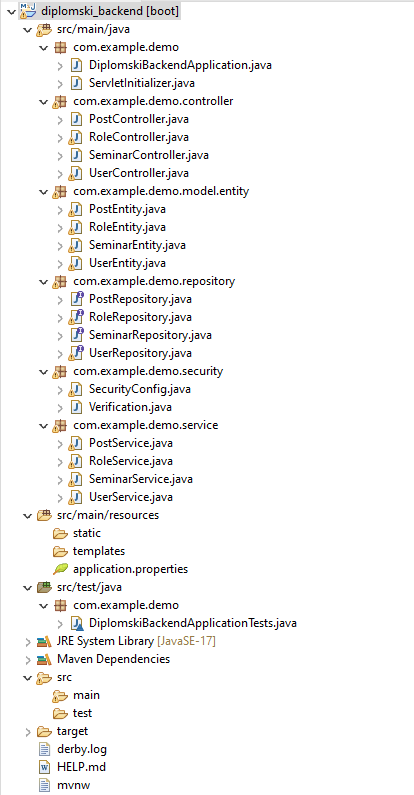


**Слика 5.1.2.** *Структура Maven генерисаног пројекта*.

### 5.1.1. Структура апликације

Структура апликације прати *Model-View-Controller* (MVC) архитектонски образac, при чему поглед није експлицитно дефинисан унутар *Spring* оквира. Кориштењем *MVC* образца апликација се дијели на три основне компоненте [20]:

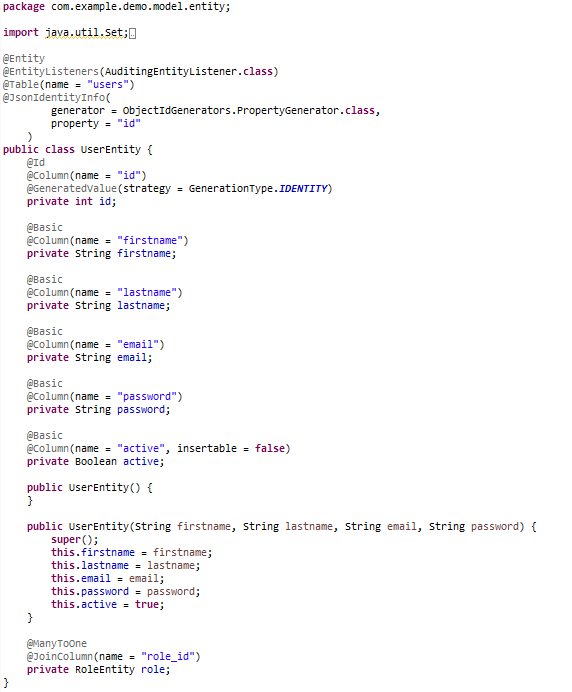
1. ***Модел*** (енгл. *Model*) – Представља податке и пословну логику, обезбјеђује читање и ажурирање података.
2. ***Поглед*** (енгл. *View*) – Обезбјеђује кориснички интерфејс за приказ података и може да прикаже информације из модела.
3. ***Контролер*** (енгл. *Controller*) – Представља повезницу између модела и погледа. Реагује на корисичке захтјеве и управља интеракцијом између модела и погледа.



**Слика 5.1.1.1.** *Структура пројекта након креирања свих потребних пакета*.

Модели апликације креирани су унутар пакета *com.example.demo.model.entity*. Модели су *PostEntity*, *UserEntity*, *RoleEntity*, *SeminarEntity* и они одговарају табелама у бази података.

На *Слици 5.1.1.2*. приказан је програмски кôд којим је дефинисана класа *UserEntity*.



**Слика 5.1.1.2.** *UserEntity класа*.

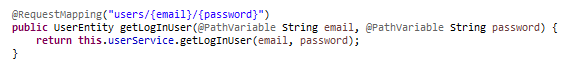
Анотацијe које су кориштене у класама које се налазе у овом пакету су следеће[16]:

* *@Entity* – Означава да класа *UserEntity* представља *JPA* ентитет који користи ову анотацију како би препознао класу као ентитет и мапирао је на базу података.
* *@EntityListeners(AuditingEntityListener.class)* - Указује на кориштење *AuditingEntityListener*-a који аутоматски поставља вриједности за временске ознаке, као што су датум креирања и ажурирања ентитета.
* *@Table(name = 'users')* – Користи се за прилагођавање детаља у бази података. У овом случају мапира ентитет на табелу са именом „users“.
* *@JsonIndentityInfo* - Управља идентитетом приликом серијализације у *JSON* формат.
* *@Id* - Означава да је поље 'id' кључ ентитета.
* *@GeneratedValue(strategy = GenerationType.IDENTITY)* –Користи се за аутоматско генерисање вриједности за поље које се наведе испод анотације, у овом примјеру то је поље 'id'.
* *@Column(name = 'id')* - Дефинише име колоне у табели базе података која се мапира на одговарајуће атрибуте у класи.
* *@ManyToOne*, *@JoinColumn* - Користе се за мапирање асоцијација. Поред ове, постоје и *@OneToMany*, *@OneToOne*, *@ManyToMany* анотације.

Унутар класе *com.example.demo.controller* налазе се контролери за сваки од креираних ентитета (*Слика 5.1.1.1*) . Примјер контролера за *UserEntity* класу приказан је на *Слици 5.1.1.4.*

Поред наведених кориштене су и следеће додатне анотације [16]:

* *@RestController* - Помоћу ове ознаке класа се означава као контролер, а сама анотација комбинује двије анотације *@Controller* и *@ResponseBody*, што значи да се резултати метода директно конвертују у *HTTP* одговор као *JSON* или *XML*.
* *@CrossOrigin(origins = '\*')* - Омогућава захтјевима са свих извора да приступају ресурсима на серверу.
* *@RequestMapping* - Користи се за мапирање *HTTP* захтјева на одређени метод (*Слика 5.1.1.3.*) .
* *@PathVariable* - Користи се за екстракцију вриједности из *URL* путање (*Слика 5.1.1.3.).*
* *@RequestBody* – Мапира тијело *HTTP* захтјева на објекат.



**Слика 5.1.1.3.** *Примјер кориштења @RequestMappig @PathVariable анотација.*



**Слика 5.1.1.4.** *UserController класа*.

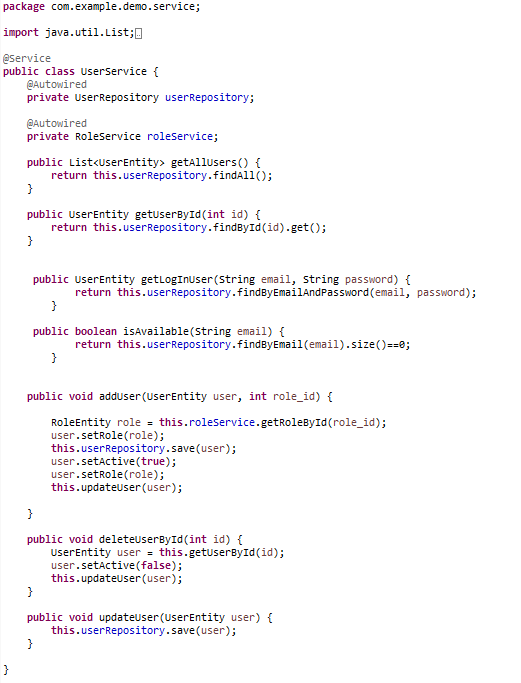
У пакету *com.example.demo.repository* налазе се интерфејси који се користе за приступ и манипулацију подацима. Користи се у комбинацији са *Spring Data JPA* модулом. Да би се користио овај модул потребно је додати зависност у *pom.xml* датотеку (*Слика 5.1.1.5.*).



**Слика 5.1.1.5.** *Дио pom.xml датотеке у којој се дефинише зависност Spring Data JPA модула*.

*Service* класе налазе се у пакету *com.example.demo.service* и представљају слој пословне логике у апликацији. Све *Service* класе означене су анотацијом *@Service*.

На *Слици 5.1.1.6.* је приказана *UserService* класа која управља пословном логиком везаном за кориснике. Користи *UserRepository* интерфејс како би приступила подацима о корисницима из базе података и *RoleService* за добијање података о улогама. Користећи методе ове класе можемо добити листу свих корисника, провјерити да ли је могућа регистрација новог корисника, додавање новог корисника и друго.



**Слика 5.1.1.6.** *UserService класа.*

### 5.1.2. Модел базе података

У овој секцији приказан је модел базе података.

База се састоји од четири основне табеле које представљају четири кључна ентитена у систему (*Слика 5.1.2.1):*

1. ***User*:** Представља све кориснике апликације.
2. ***Role*:** Представља улоге које корисник може да има (корисник, предавач или администратор).
3. ***Seminar*:** Представља семинар који се одржавао или се тек треба одржати, садржи колоне у којима се чувају информације о пријављеним учесницима, вријеме одржавања.
4. ***Post*:** Користи се за чување информација о објавама које нису семинари, а могу да буду објавјештења, научни чланци и слично.

Поред четири основне табеле постоје и двије међутабеле које се користе при креирању m:n веза.



***Слика 5.1.2.1.*** *Модел базе података*

## 5.2. Развој корисничког интерфејса

Клијентски интерфејс је развијен помоћу *React* библиотеке. Поред *React* библиотеке кориштене су и неке додатне библиотеке које омогућавају лакши и бржи развој апликације, односно:

* *React Router*
* *React Date Picker*
* *React Icons*
* *Days*
* *Axios*
* *Material UI*
* *Chakra UI*

Апликација је креирана преко терминала кориштењем *create-react-app* команде (*Слика 5.2.1*).

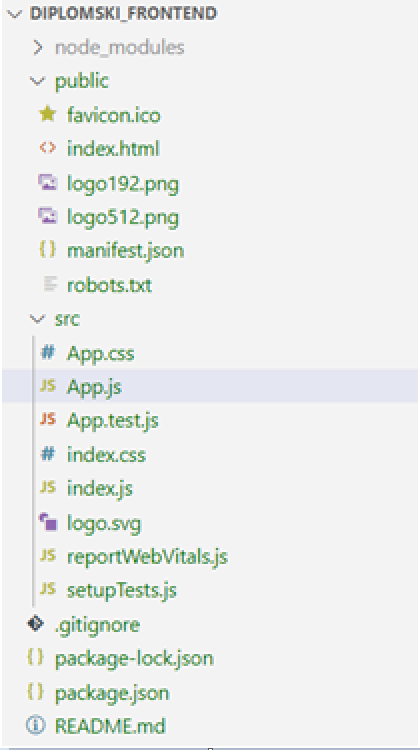
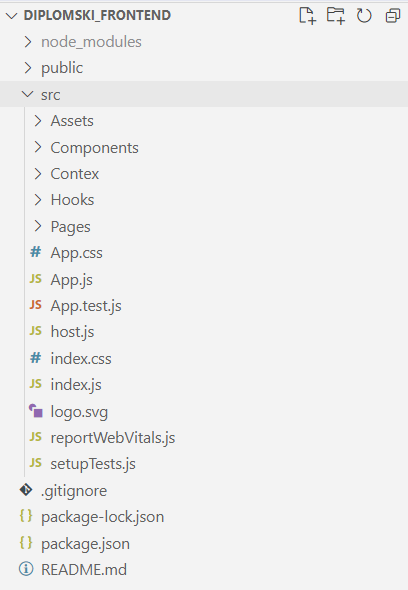


**Слика 5.2.1.** *Приказ терминала и наредбе за креирање React пројекта*.

*Npx* ( *Node Package eXecute*) се користи за извршавање пакета у оквиру алата *npm* и омогућава извођење пакета без потребе за њиховим глобалним инсталирањем. Када се покрене одређени пакет помоћу *Npx*-a, аутоматски се ствара директоријум са називом који смо навели у наставку наредбе [21]. У нашем случају то је diplomski-frontend (*Слика 5.2.2.).* Овај директоријум садржи све потребне ресурсе за покретање пакета:

* ***node\_modules*** – Садржи све инсталиране *npm* пакете. Пакете можемо инсталирати преко терминала било када у процесу развоја апликације, тако што након команде *npm install* наведемо име пакета.
* ***public*** – Представља јавне датотеке попут иконица и *index.html* документа који се користи за рендеровање *React* апликације у веб прегледачу.
* ***src*** – Конвенција која се користи за организацију изворног кода.
* ***.gitignore*** – Садржи списак датотека које се игноришу у *Git*-y при извршавању *commit* наредбе.
* ***package.json*** – Садржи све информације о инсталираним пакетима и скриптама за покретање апликације.

Након успјешног креирања пројекта, можемо покренути скрипту за отварање сервера кориштењем наредбе *npm start* коју уносимо у терминал. На локалном порту 3000 приказује се кориснички интерфејс апликације.

**Слика 5.2.3.** *Структура апликације након дефинисања потребних фолдера*.

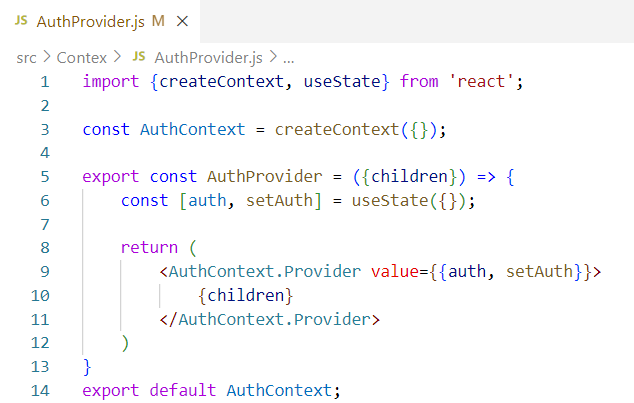
**Слика 5.2.2.** *Почетна структура апликације.*

На *Слици 5.2.3.* можемо видјети структуру апликације након креирања свих фолдера који ће нам бити потребни за развој апликације.

Унутар фолдера *Component* налазе се компоненте које се могу употребљавати на више мјеста, као што су, на примјер, *header* и *footer*. Кориштењем компоненти кориснички интерфејс се дијели у засебне цјелине које се могу поново користити, што у великој мјери смањује вријеме израде и комплексност апликације.

*Hooks* и *Contex* користе се за управљање стањем и дијељењем података између компоненти.

*АuthProvider.js* налази се унутар Contex датотеке и служи за праћење стања аутентификације (*Слика 5.2.4*). *AuthContext* садржи *AuthProvider* компоненту која користи *useState* да би пратила стање аутентификације и *AuthContex.Provider* да би другим компонентама обезбиједио приступ и могућност измјене аутентификацијских података.



**Слика 5.2.4.** *AuthProvides.js документ*.

Функционалности *AuthProvider*-a налазе се у *useAuth.js* документу, у *Hooks* датотеци (*Слика 5.2.5.*).

Двије кључне функције које су дефинисане унутар овог документа су:

1. ***saveAuth*** – Ова функција омогућава постављање стања аутентификације, односно ажурирање локалног складишта са новим аутентификацијским подацима.
2. ***clearAuth*** – Поставља стање аутентификације на *null* и уклања аутентификацијске податке из локалног складишта. Ова функција користи се приликом одјаве корисника.



**Слика 5.2.5.** useAuth.js документ.

У наставку ће бити представљени и детаљно објашњени сљедећи кориснички модули:

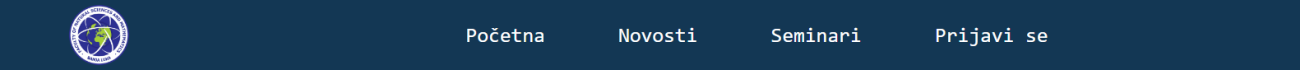
1. *Почетна страна*
2. *Модул за регистрацију корисника*
3. *Модул за пријаву корисника*
4. *Модул за приказ корисничког профила*
5. *Модул за додавање новог поста*
6. *Модул за приказ свих семинара*
7. *Модул за приказ свих објава*
8. *Модул за приказивање појединачне објаве*
9. *Администраторски панел*

### 5.2.1. Почетна страна

Почетна страна представља почетни модул који се приказује кориснику када се кроз веб претраживач отвори веб локација апликације за потребе управљања научним семинаром (*Слика 5.2.1.3*).

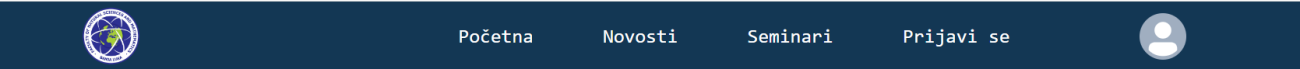
На почетној страни налазе се линковане форме ка модулима за регистрацију корисника, приказ свих семинара и *slider* у коме се приказују објаве. Кликом на дугме које се налази унутар картице у *slider-*у кориснику се отвара нови прозор односно модул за приказ појединачне објаве.

На врху странице налази се заглавље које се користи за навигацију. Уколико корисник није пријављен нуде му се четири опције: *Почетна*, *Новости*, *Семинари* и *Пријави се* (*Слика 5.2.1.1.).* Кликом на једну од понуђених опција отвара се одговарајући модул.

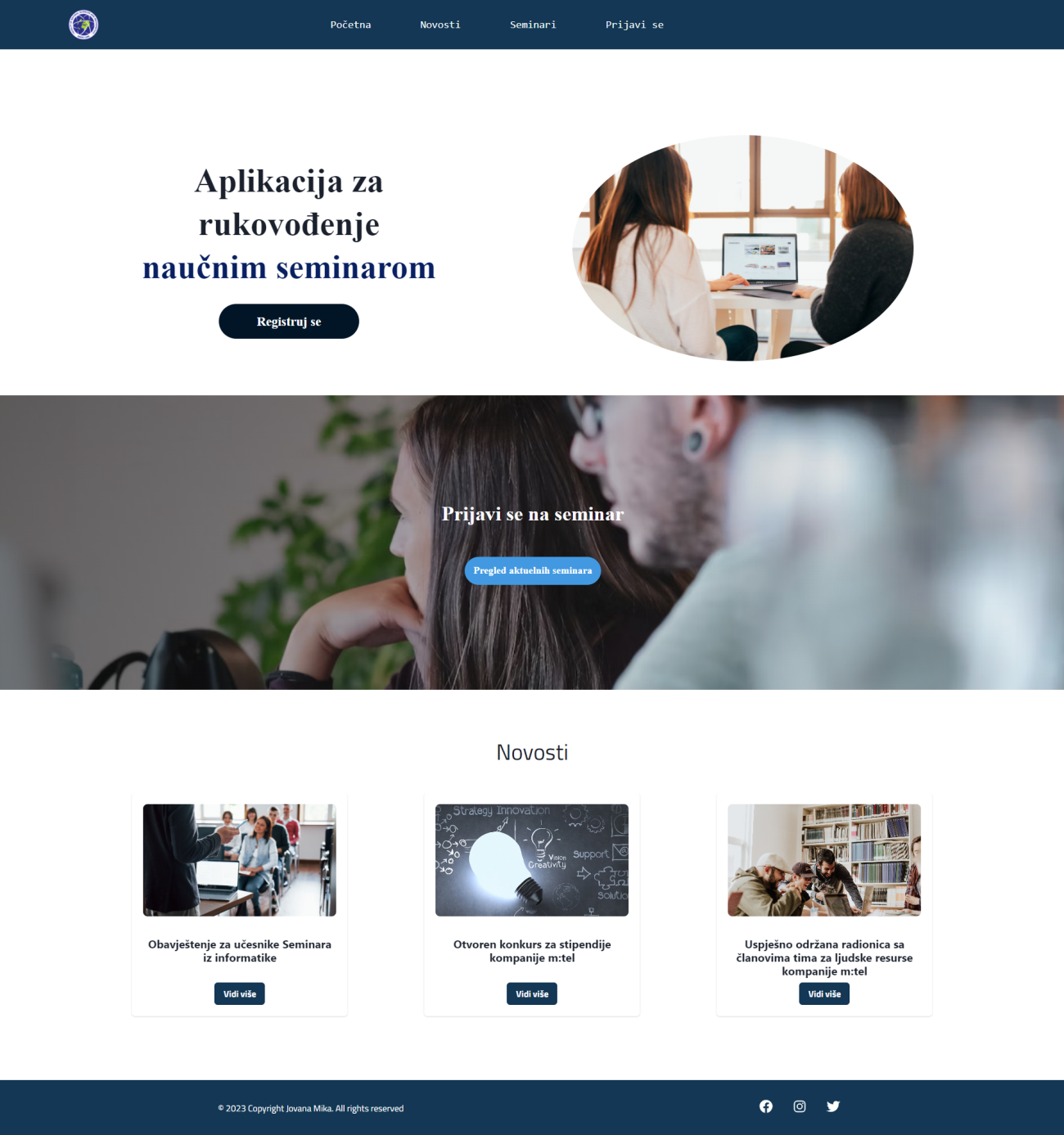


**Слика 5.2.1.1.** *Заглавње апликације у случају да корисник није пријавњен.*

У случају да је корисник пријављен , у десном углу заглавља се налази иконица која води до модула за приказ профила (*Слика 5.2.1.2).*



**Слика 5.2.1.2.** *Заглавље странице у случају да је корисник успјешно пријављен.*



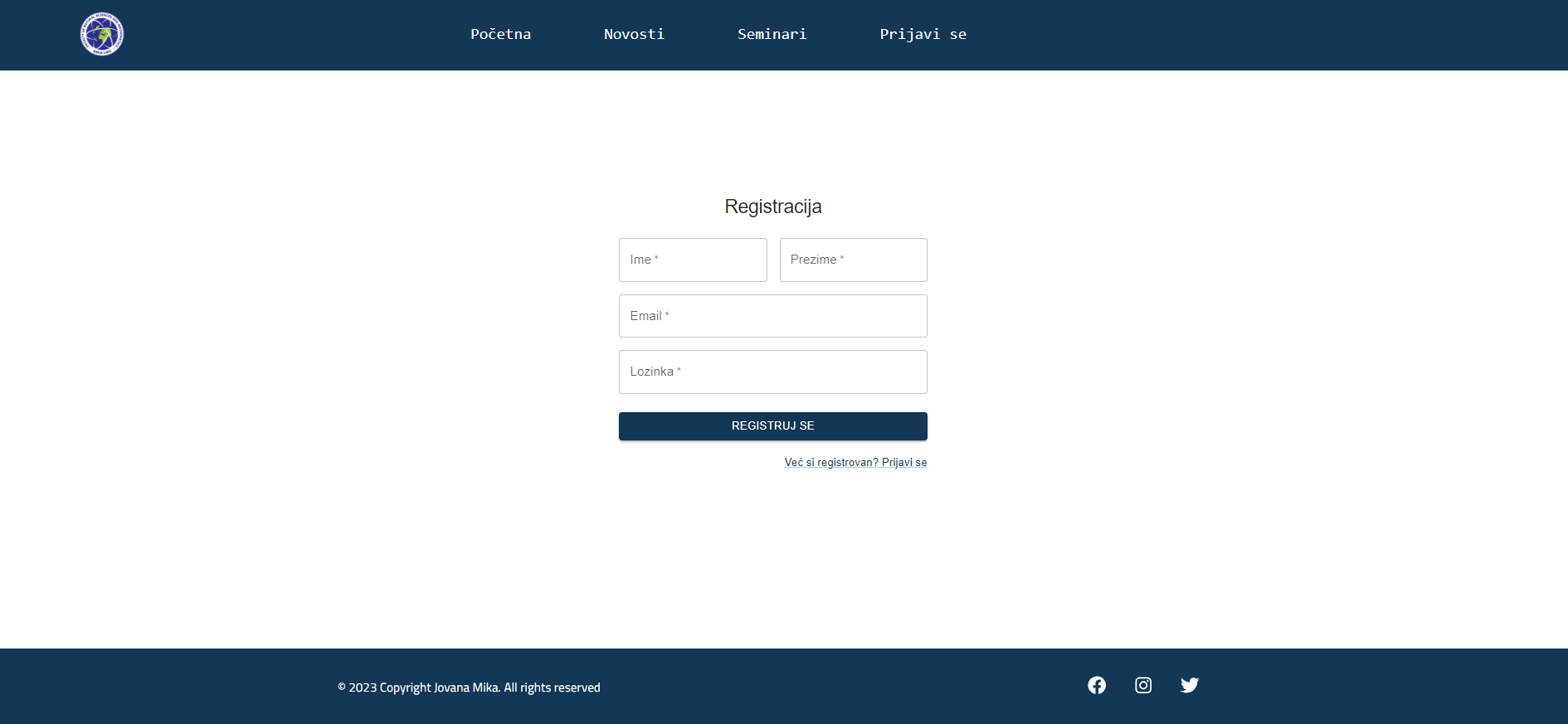
***Slika 5.2.1.3.***  *Korisnički prikaz početne strane у случају да корисник није пријављен.*

На дну странице налази се подножје (енгл. *footer*), и садржи информацију о креатору апликације и три линковане иконице које воде ка друштвеним мрежама институције којој је апликација намијењена.

### 5.2.2. Модул за регистрацију

Регистрација корисника представља једну од функционалности система и омогућава корисницима да стварају и одржавају своје корисничке профиле, као и да остваре одређене интеракције које нису могуће уколико корисник није регистрован.

Модул за регистрацију садржи форму унутар које корисник уноси своје личне податке: име, презиме, email и лозинку (*Слика 5.2.2.1*).



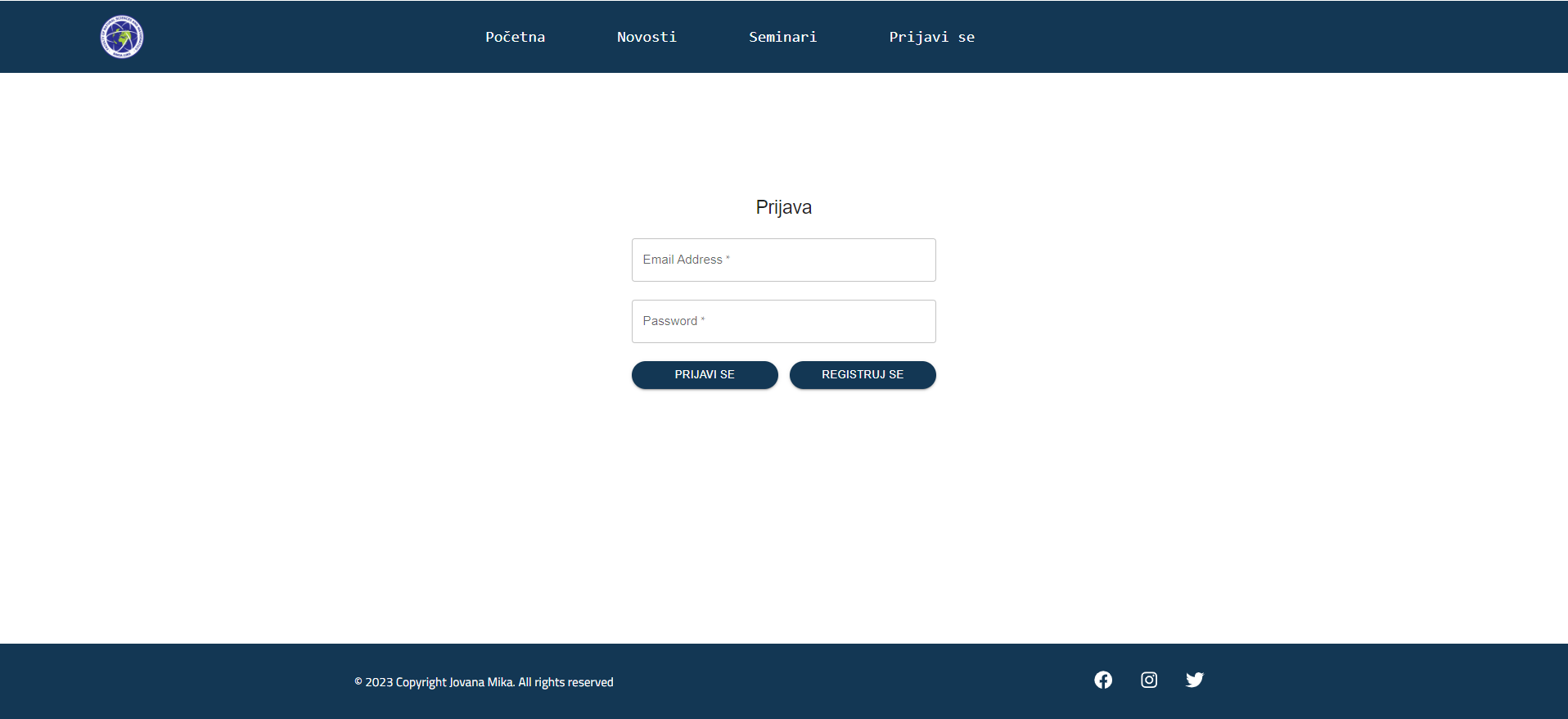
***Слика 5.2.2.1.*** *Приказ модула за регистрацију.*

Након што корисник унесе све неопходне податке, кликом на дугме „Региструј се“ иницира се процес слања захтјева серверу за упис новог корисника у базу података. Прије него што се подаци упишу у базу врше се провјере које обухватају валидацију унијете електронске адресе како би се осигурало да дата адреса одговара дефинисаним стандардима. Поред тога, систем провјерава да ли унијета адреса већ постоји у бази података како би се спријечило дуплирање корисничких налога.

Процес провјере података прије уписа у базу представља кључну улогу у осигуравању тачности и досљедности података.

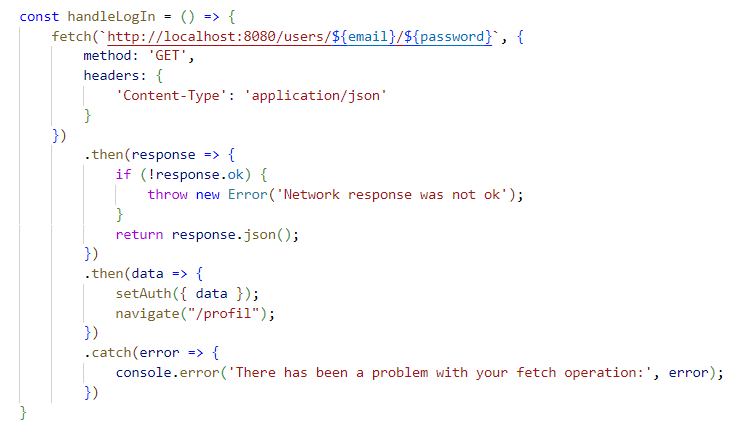
### 5.2.3. Модул за пријаву

Модул за пријаву се састоји од форме у коју корисници уносе своје легитимационе податке, односно електронску адресу и лозинку.



***Слика 5.2.3.1.*** *Приказ модула за пријаву.*

Након уношења података, кликом на дугме „Пријави се“ иницира се слање захтјева за дохватање информација о кориснику из базе података (*Слика 5.2.3.2.*). Уколико такав корисник постоји у бази података, пријава је успјешна и корисник се преусмјерава на почетну страну.



**Слика 5.2.3.2.** *Функција која се извршава као акција на дигме „Пријави се“.*

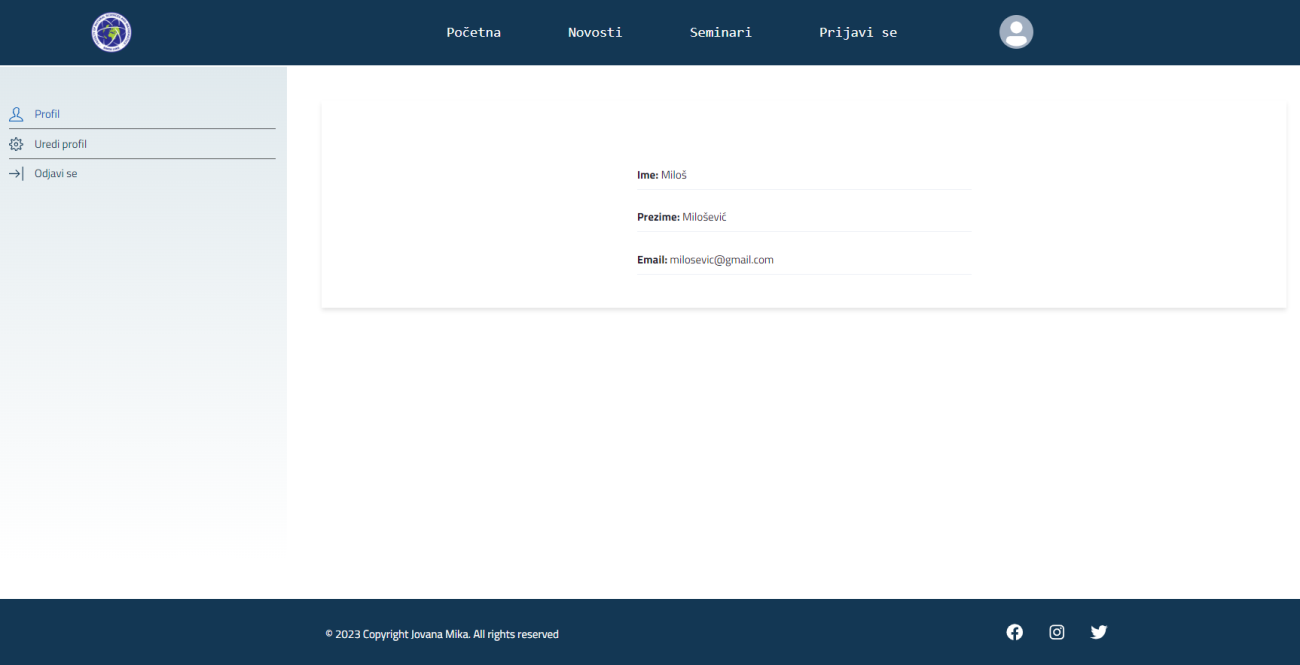
### 5.2.4. Модул за приказ корисничког профила

Са лијеве стране модула за приказ корисничког профила налази се панел који садржи следеће опције:

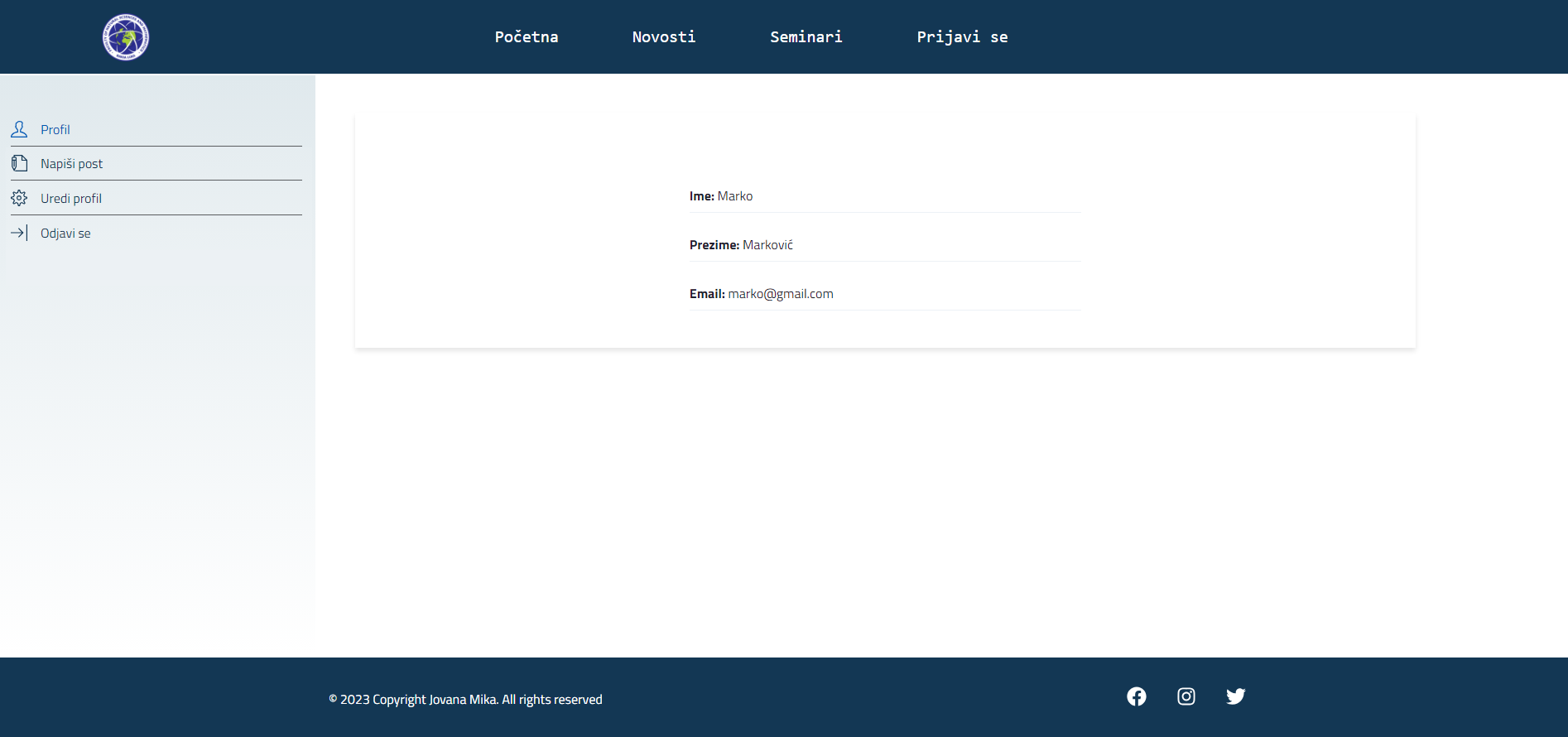
1. ***Profil*** - Приказ основних информација о пријављеном кориснику.
2. ***Napiši post*** - Опција за слање захтјева за додавање новог поста (уколико пријављени корисник има улогу предавача).
3. ***Uredi profil*** - Омогућава мијењање корисничких података.
4. ***Odjavi se*** – Служи за одјављивање са корисничког профила.

На *Слици 5.2.4.1.* приказан је профил корисника чија је улога „Корисник“, док је на *Слици 5.2.4.2.* приказан профил корисника чија је улога „Предавач“. Информације које се приказују у оквиру изабране секције су име, презиме и електронска адреса пријавњеног корисника.

Разлика између наведених улога је у томе што предавач има могућност слања захтјева администратору за додавање новог поста, док обичан корисник нема ту опцију.

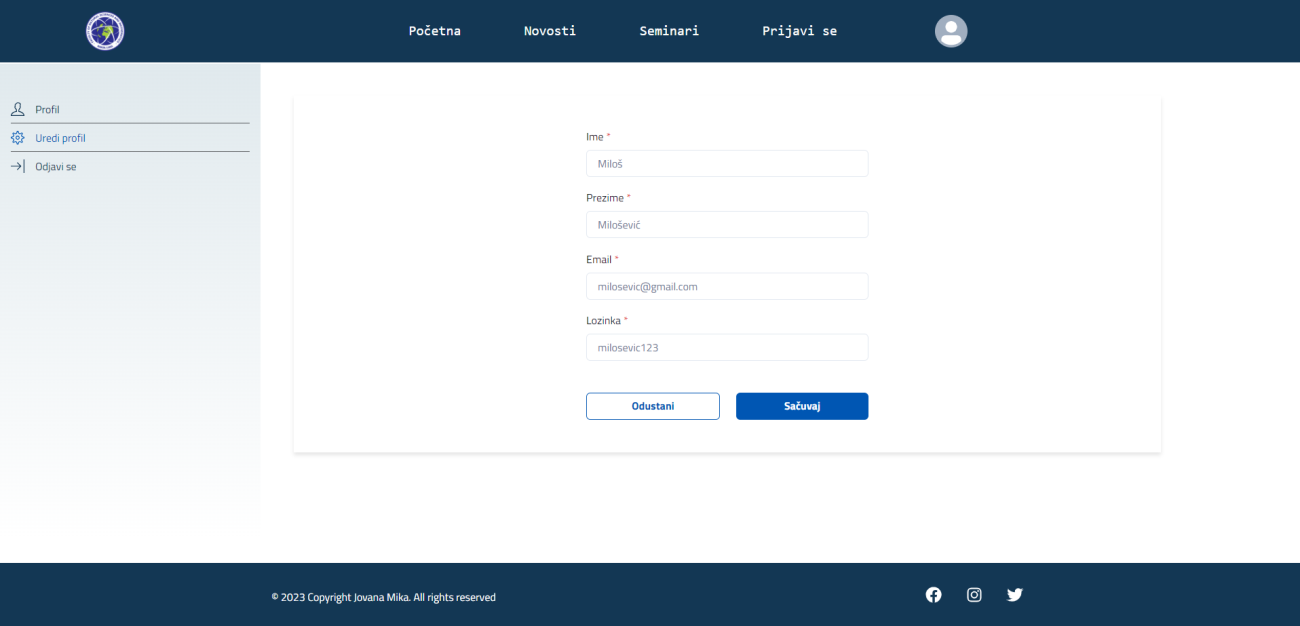


***Slika 5.2.4.1.*** *Приказ профила у случају да је пријављен корисник чија је улога „Корисник“.*



***Slika 5.2.4.2.*** *Приказ профила у случају да је пријављен корисник чија је улога „Предавач“.*

На *Слици 5.2.4.3.* приказана је опција за уређивање профила, која пружа корисницима могућност да лако и сигурно измијене своје личне податке. Могуће је ажурирати следећа поља: име, презиме, електронску пошту и лозинку. Након уписа података које желимо да измијенимо, потребно је кликнути на дугме „Сачувај“ чиме се серверу шаље захтјев за ажурирање измјена у бази података.



***Слика 5.2.4.3.*** *Приказ опције за уређивање профила.*

### 5.2.5. Модул за додавање новог поста

На *Слици 5.2.5.1.* приказана је опција за слање захтјева администратору за додавање новог поста. Ова опција приказује се само у случају да пријављени корисник има улогу предавача.

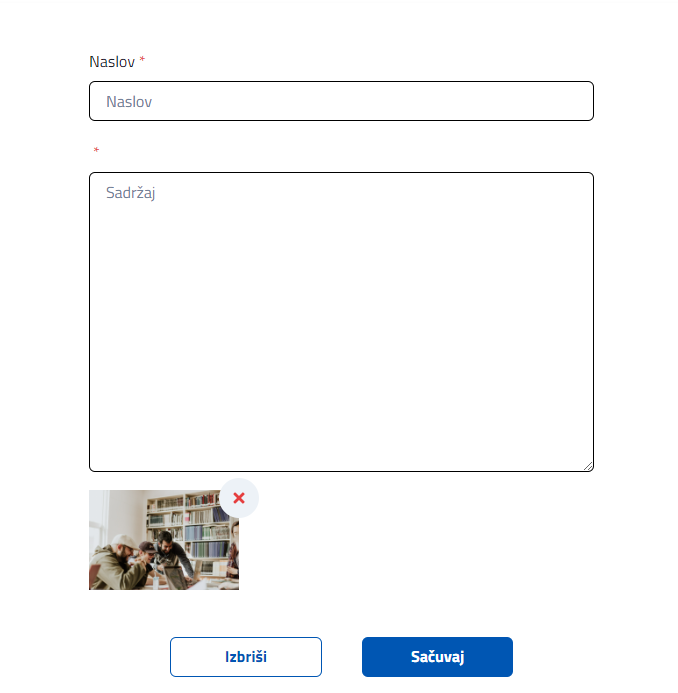
Приликом додавања новог поста, предавач уноси наслов поста, садржај (текст) поста, као и фотографију која ће се приказивати у картици доданог поста. Уколико предавач не изабере ниједну од понуђених фотографија, посту се додјељује подразумијевана фотографија. Кликом на дугме „Додај фотографију“ отвара се модал који садржи све фотографије које је могуће изабрати (*Слика 5.2.5.2.).* Кликом на изабрану фотографију, она се појављује умјесто дугмића „Додај фотографију“ (*Слика 5.2.5.3.*)*.* Изабрану фотографију могуће је избрисати кликом на иконицу „X“ која се налази у десном горњем углу фотографије, и у том случају дугме „Додај фотографију“ опет постаје видљиво те корисник може да изабере другу фотографију. Приликом креирања новог поста од стране предавача, систем аутоматски шаље захтјев за објављивање администратору система. Након тога, администратор даје финално одобрење да се тај пост појави на сајту. На овај начин спречава се појава постова који нису релевантни садржају који апликација нуди.

## 

***Слика 5.2.5.1.*** *Приказ опције за слање захтјева за додавање новог поста у случају да је пријавњени корисник предавач.*

## 

**Слика 5.2.5.2.** *Приказ модула за приказ свих насловних фотографије које предавач може да изабере.*



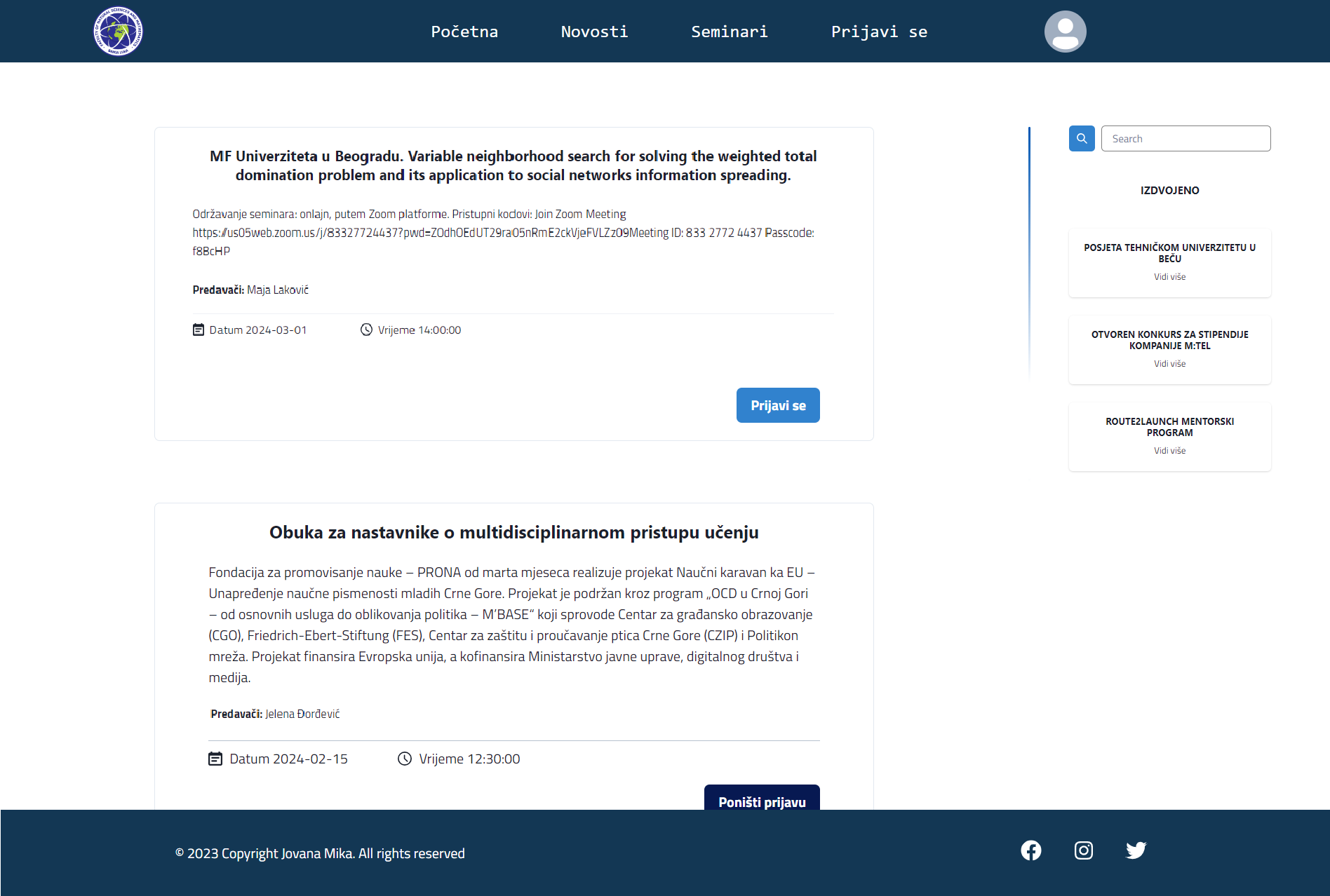
**Слика 5.2.5.3.** *Приказ форме за додавање новог поста, у случају да је предавач изабрао насловну фотографију.*

### 5.2.6. Модул за приказ семинара

Модул за приказ семинара састоји се из лијевог и десног панела (*Слика 5.2.6.1*).

У лијевом панелу приказани су сви семинари које учитавамо из базе. Сваки семинар приказан је у виду картице на којој се налазе информације о семинару, а то су наслов, опис, предавачи који учествују у семинару, датум и вријеме одржавања. У доњем десном углу картице налази се дугме „Пријави се“ које омогућава корисницима да се пријаве на жељени семинар. Пријавити се могу само они корисници који су регистровани у систему, при чему улога корисника у овом случају није битна. Другим ријечима, на семинар се може пријавити било који регистровани корисник. Корисницима који су већ пријављени на семинар умјесто опције за пријаву приказује се дугме „Поништи пријаву“. Кликом на одговарајуће дугме, иницира се захтјев серверу за пријаву или одјаву са семинара.

Важно је напоменути да се ове акције не могу извршити уколико је семинар већ одржан, чиме се обезбјеђује јасан и усклађен ток управљања пријавама на семинар.



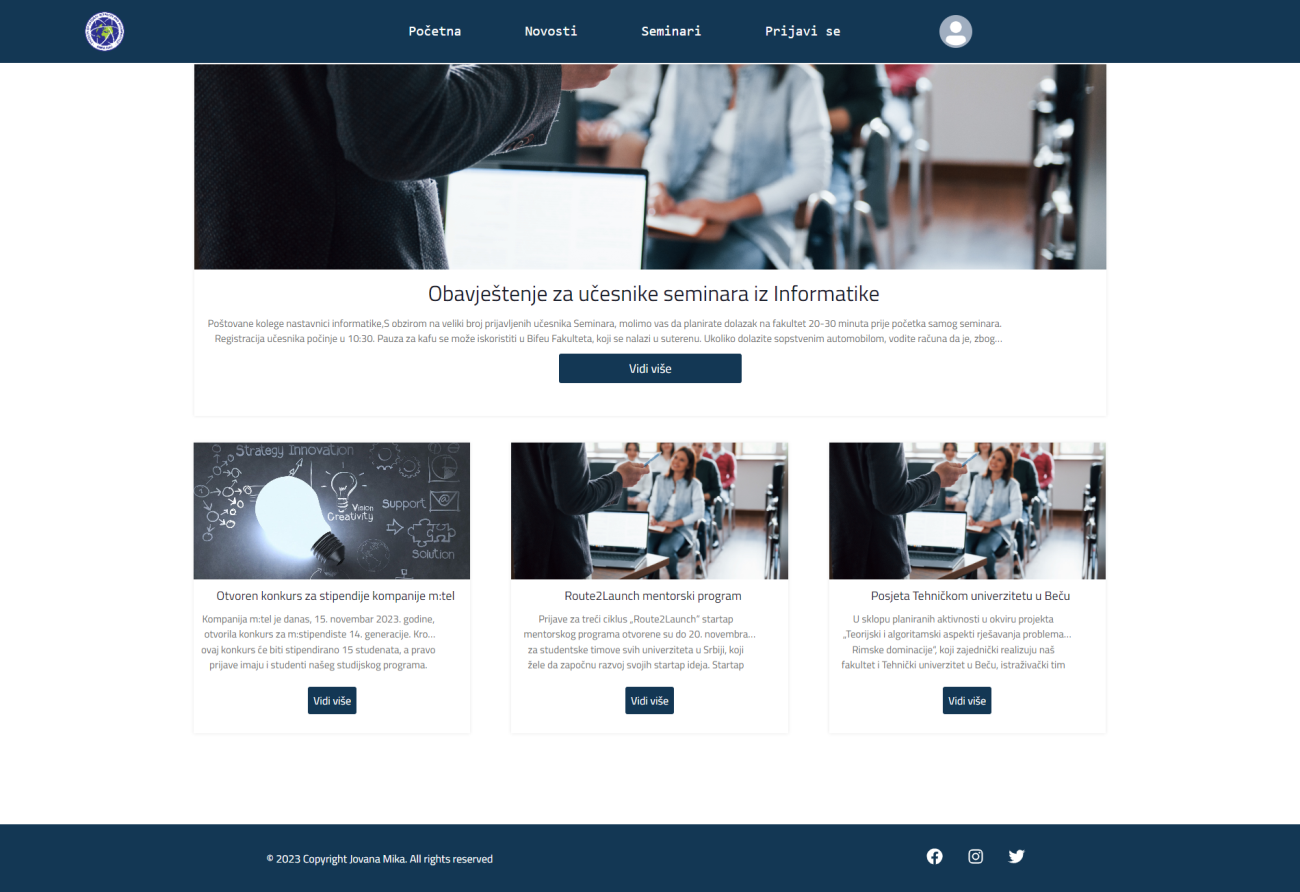
***Слика 5.2.6.1.*** *Визуелна репрезентација модула за приказ семинара.*

У десном панелу налази се опција за претрагу семинара по називу. Корисник уноси дио или пун наслов семинара који жели да пронађе. Кликом на иконицу за претрагу, серверу се шаље захтјев за проналажење свих семинара који задовољавају наведени услов. Ова опција омогућава корисницима да брзо и ефикасно пронађу жељени семинар путем једноставног уноса информација. Испод ове опције издвојено је и неколико актуелних објава.

Кликом на картицу која садржи информације о објави, кориснику се отвара модул за приказ појединачне објаве.

### 5.2.7. Модул за приказ новости

Модул за приказ новости садржи картице на којима се налазе подаци везани за пост, односно наслов, слика и опис поста (*Слика 5.2.7.1.*). Кликом на дугме „Види више“ отвара се нови модул за приказ појединачног поста.

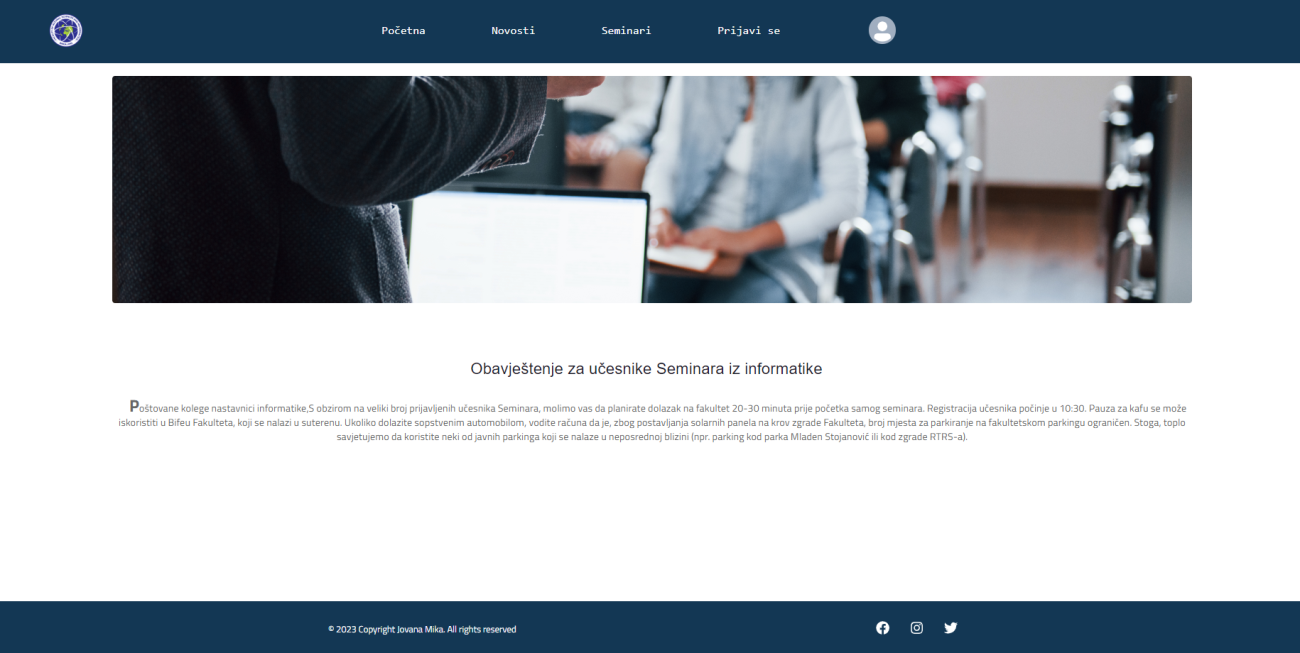


**Слика 5.2.7.1.** *Модул за приказ новости.*

### 5.2.8. Модул за приказ појединачног поста

Модул за приказ појединачног поста пружа свеобухватни увид у садржај изабраног поста. Кроз овај модул корисници могу да виде све информације које су повезане са датим постом.

На *Слици 5.2.8.1.* приказан је изглед модула за приказ појединачног поста.



***Слика 5.2.8.1.*** *Модул за приказ појединачног поста.*

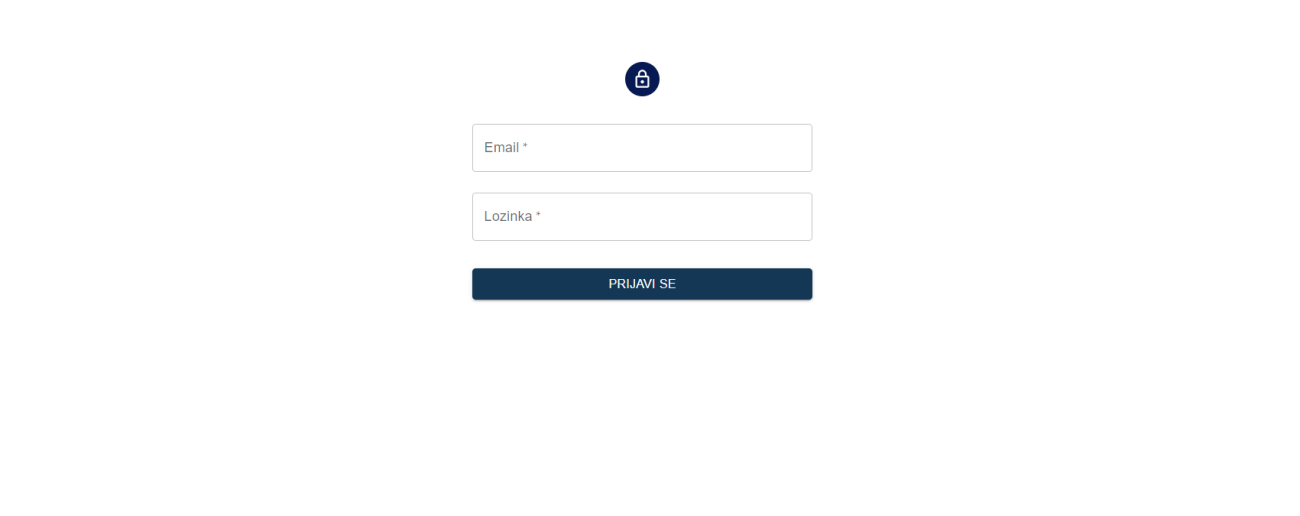
### 5.2.9. Администраторски панел

Администраторски панел представља посебан дио апликације, који овлаштеним лицима, односно корисницима са администраторским привилегијама, омогућава приступ и управљање различитим функционалностима система.

Приликом приступа администраторском панелу, корисницима се приказује форма за пријаву (*Слика 5.2.9.1.).* У форму се уносе приступи подаци, електронска пошта и лозинка.

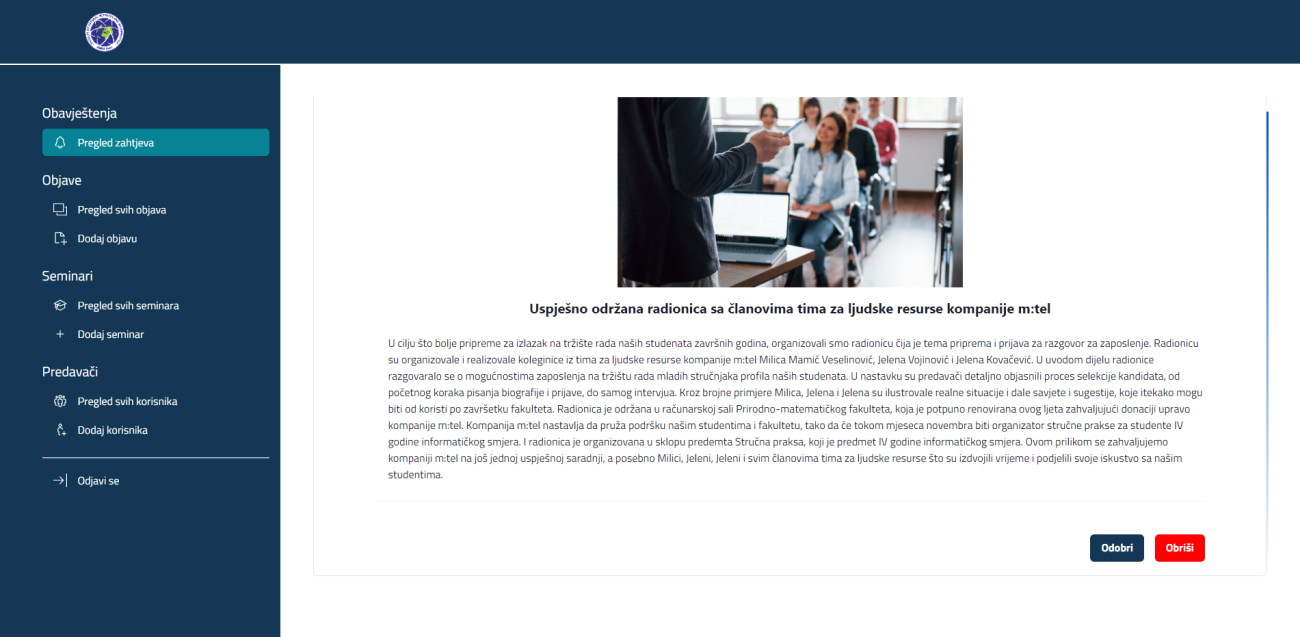
Након успјешне пријаве омогућено је:

1. Преглед свих захтјева за креирање новог поста.
2. Преглед свих постова, брисање или измјена поста.
3. Форма за креирање нове објаве.
4. Преглед свих семинара, брисање или измјена изабраног семинара.
5. Форма за креирање новог семинара.
6. Преглед свих корисника, брисање или измјена изабраног корисника.
7. Форма за додавање новог корисника.
8. Одјава.



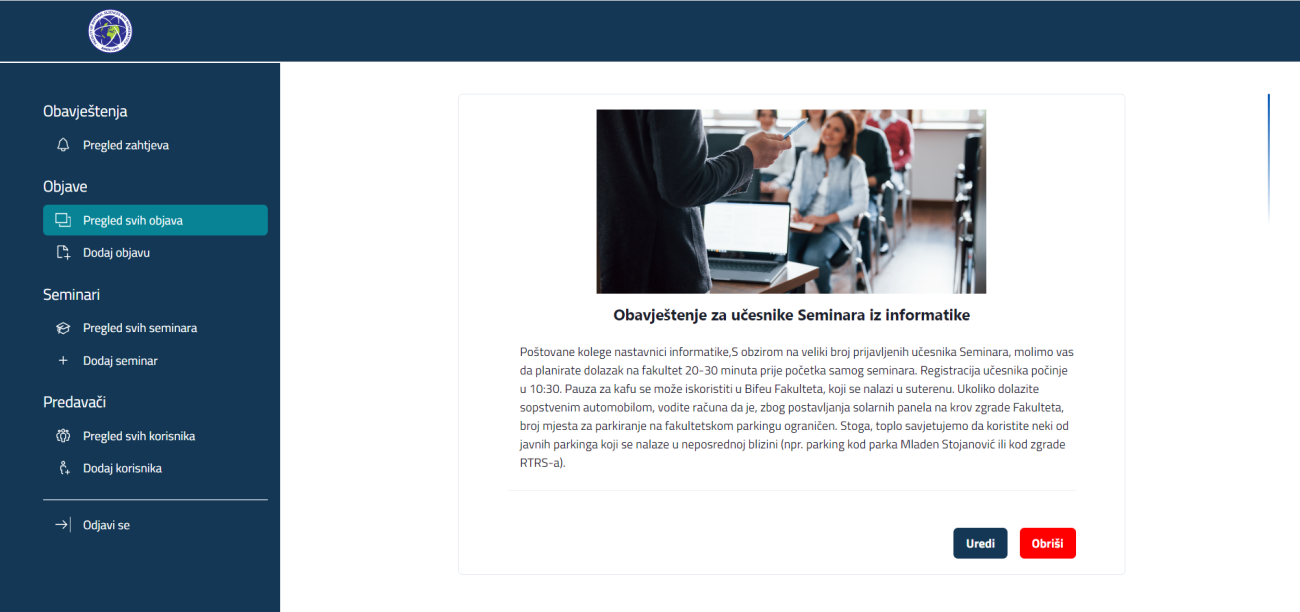
**Слика 5.2.9.1.** *Форма за пријаву на администраторски панел.*

На *Слици 5.2.9.2*. приказан је преглед свих обавјештења за додавање новог поста. Администратор може да прегледа све захтјеве и одлучи које ће да одобри, а које не. Кликом на дугме „Објави“, шаље се захтјев серверу за измјену статуса поста.

****

**Слика 5.2.9.2.** *Преглед свих обавјештења.*

На *Слици 5.2.9.3.* приказан је преглед свих постова. Администратору се приказују сви постови који су видљиви корисницима. Сваки пост приказан је у виду картице на којој се налазе два дугмића. Кликом на дугме „Уреди“ администратору се приказује модал за уређивање поста (*Слика 5.2.9.4*.). Након што администратор унесе у форму жељење измјене, кликом на дугме „Сачувај“, серверу се шаље захтјев за измјене одговарајућег поста.

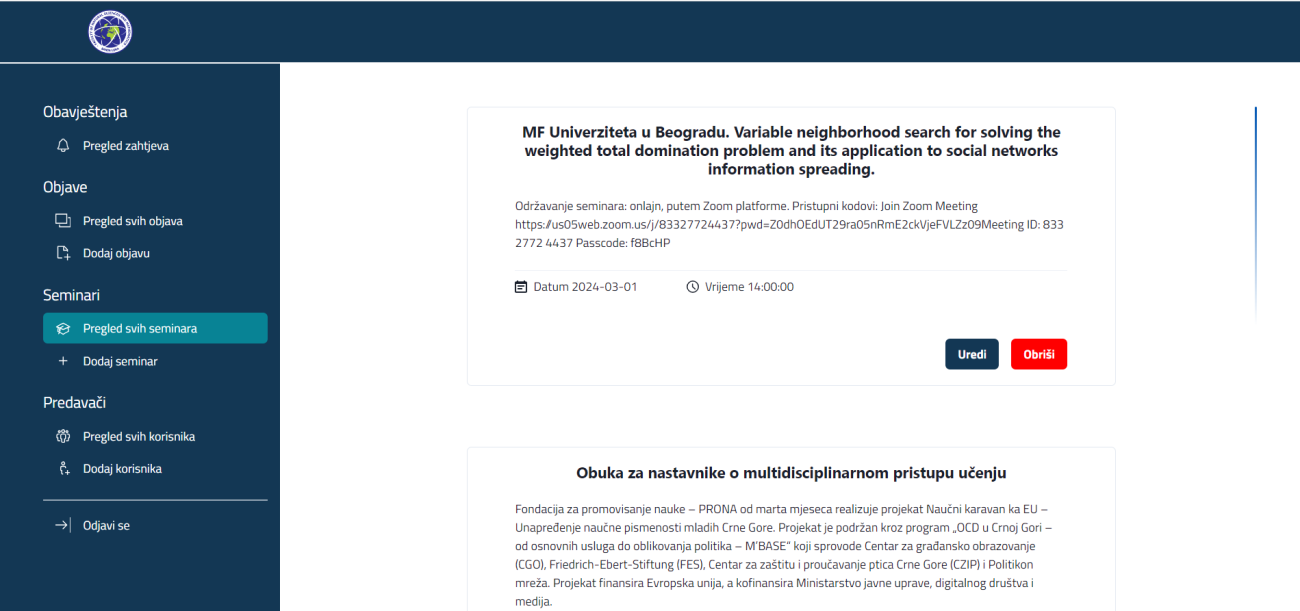
****

**Слика 5.2.9.3.** *Преглед свих постова.*

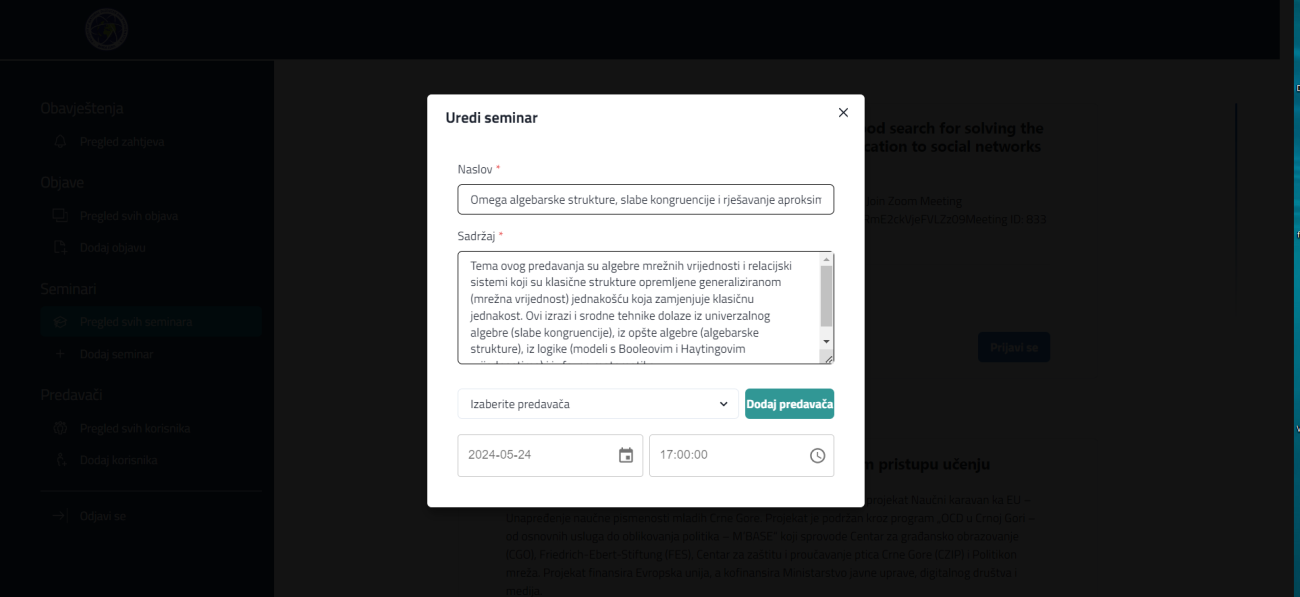
****

**Слика 5.2.9.4.** *Приказ модула за измјену поста.*

На сличан начин администратор управља семинарима који се налазе на страници. На *Слици 5.2.9.5.* налази се приказ прегледа свих семинара, док је на *Слици 5.2.9.6.* приказан модал за измјену изабраног семинара.

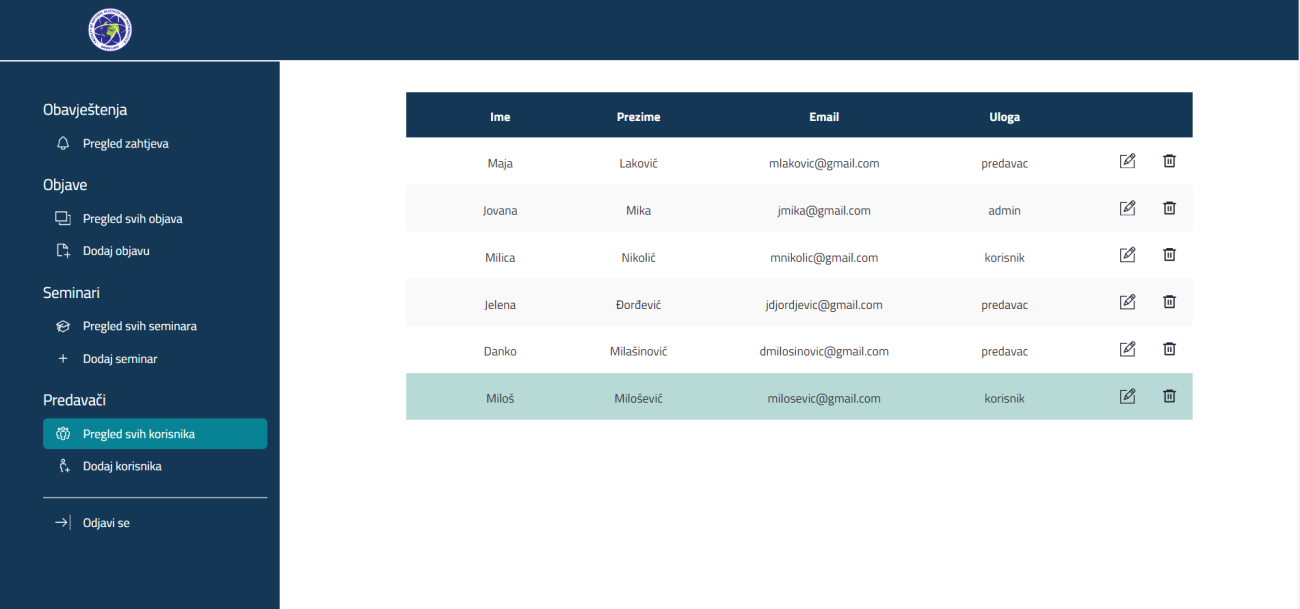
****

**Слика 5.2.9.5.** *Приказ свих семирара.*

****

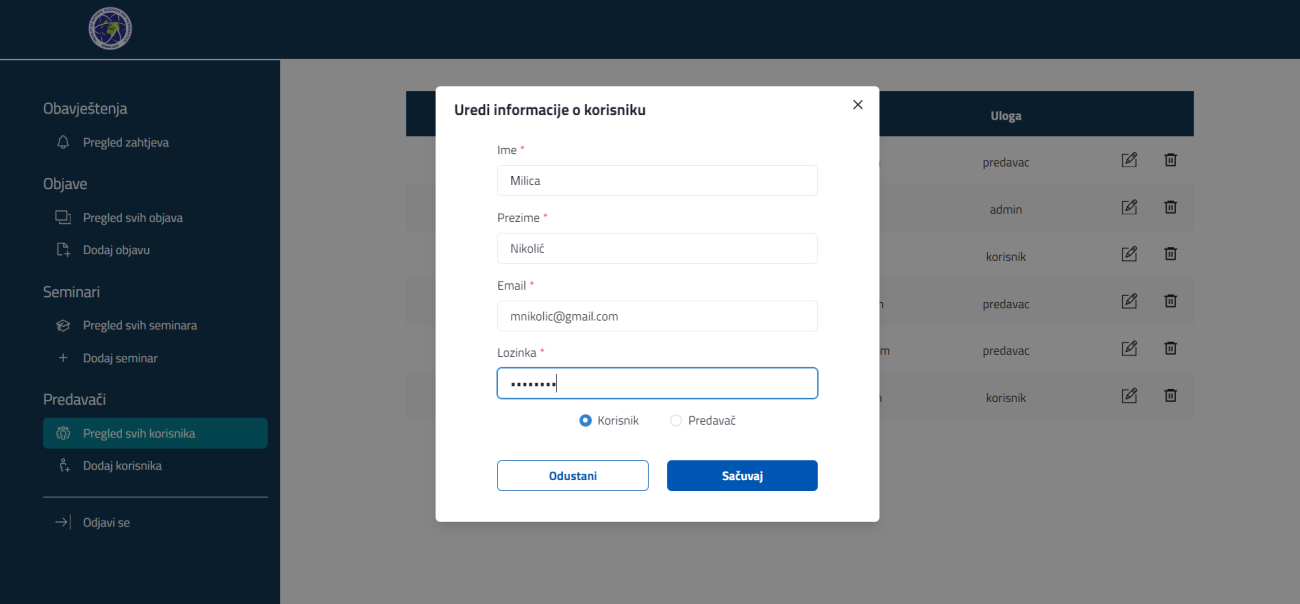
**Слика 5.2.9.6.** *Приказ модула за измјену семинара.*

На *Слици 5.2.9.7.* налази се приказ табеле свих корисника. У табелу су уписани сви регистровани корисници. У посљедњој колони, поред података сваког од корисника, налазе се иконице које се користе за измјену и брисање корисника.



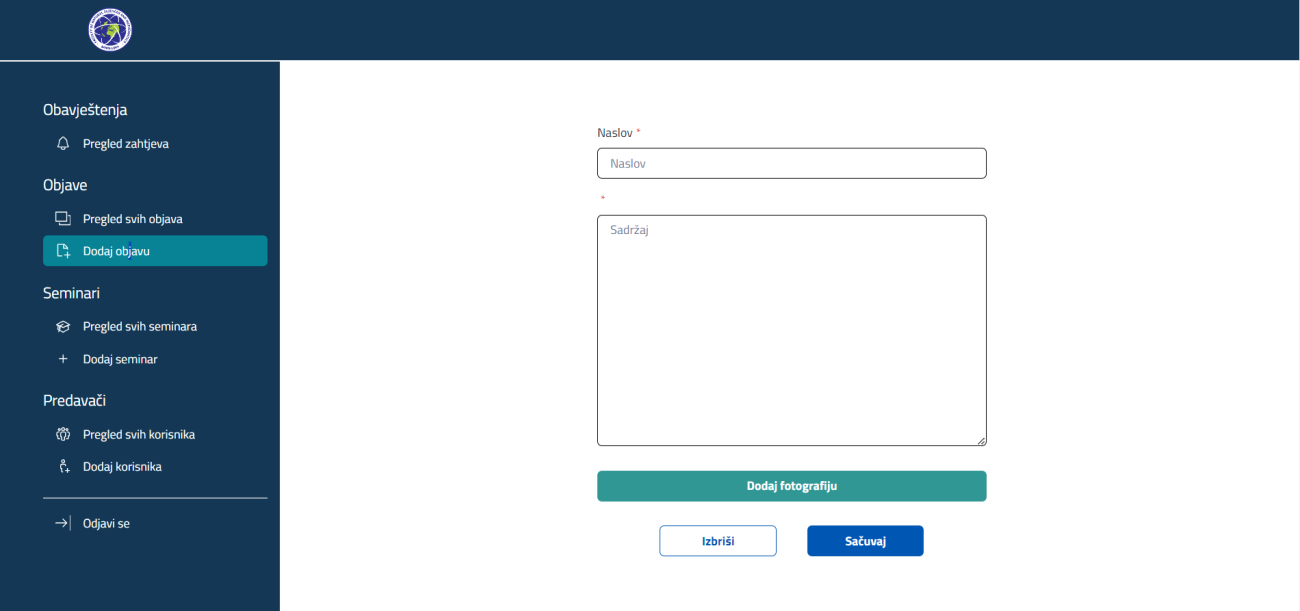
**Слика 5.2.9.7.** *Преглед свих регистрованих корисника.*

Модал за измјену корисника приказан је на *Слици 5.2.9.8.,* а процес измјене исти је као претходно описани.

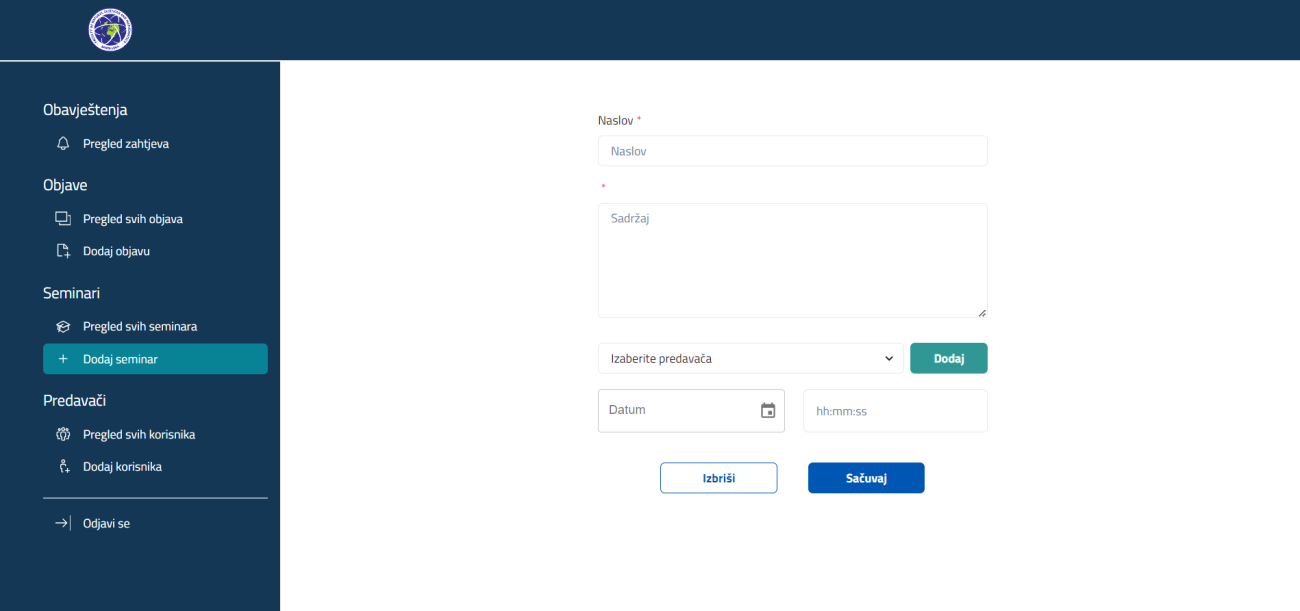
****

**Слика 5.2.9.8.** *Приказ модула за измјену корисничких података – примјер измјене лозинке.*

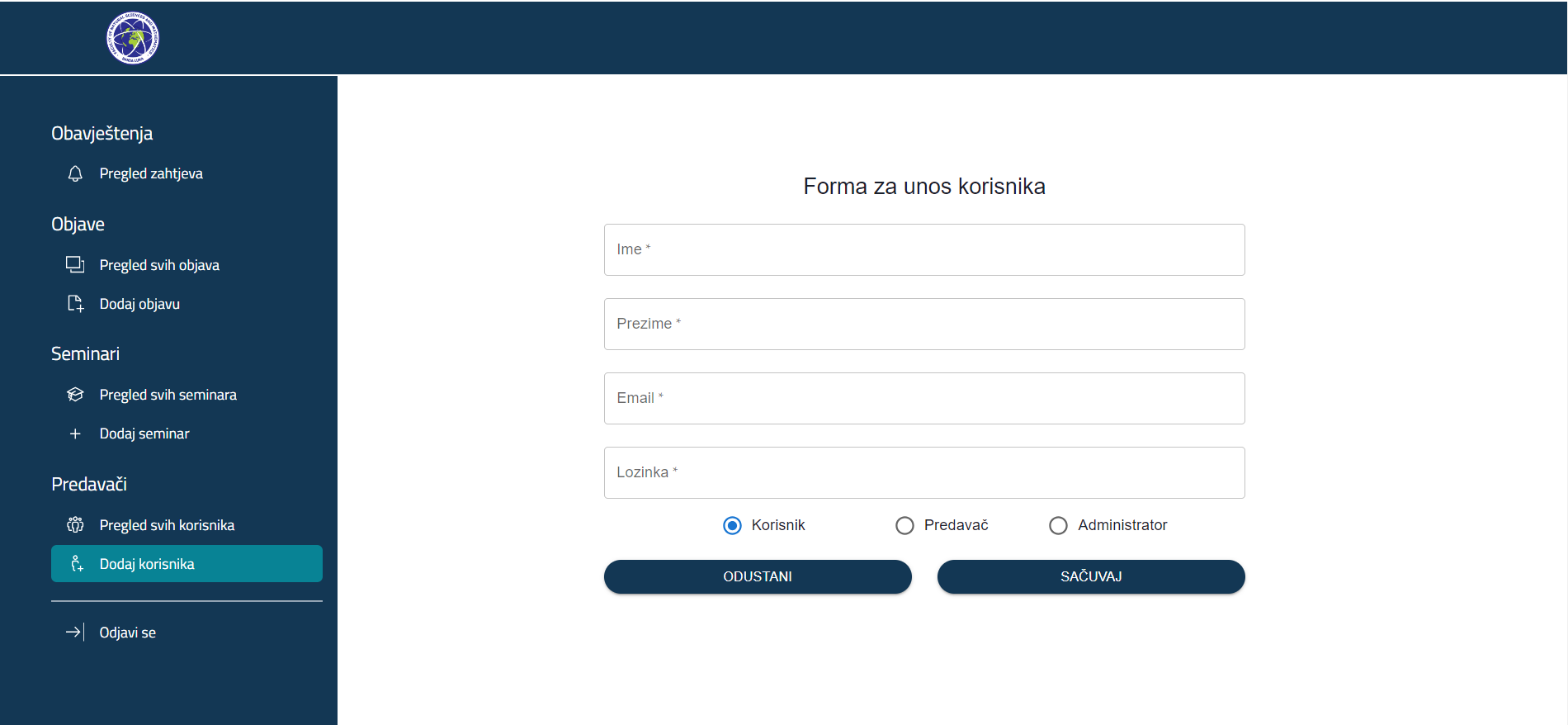
У наставку су приказане слике за додавање новог поста, додавање новог семинара и додавање новог корисника, редом.

****

**Слика 5.2.8.8.** *Приказ форме за додавање нове објаве.*

****

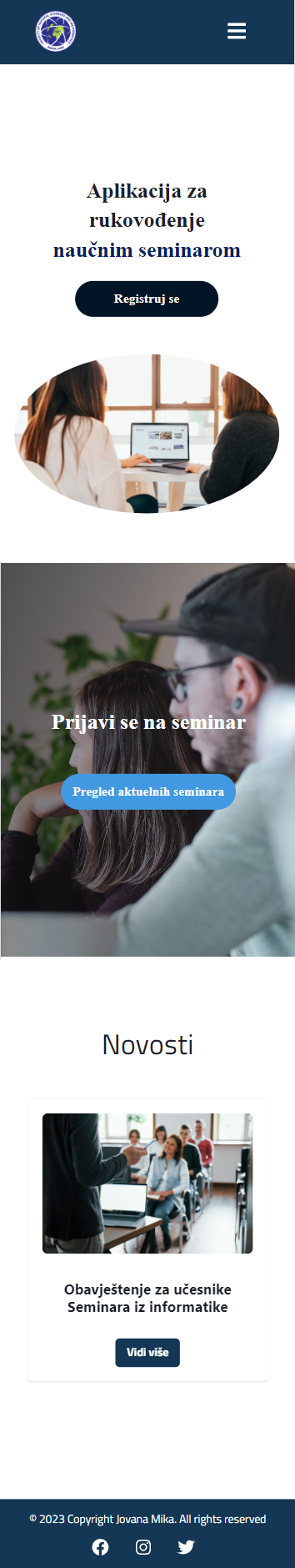
**Слика 5.2.8.10.** *Приказ форме за додавање новог семинара.*

****

**Слика 5.2.8.10.***Форма за додавање корисника.*

При изради апликације посебна пажња посвећена је адаптивности корисничког интерфејса за различите величине екрана. Респонзивни дизајн апликације обезбјеђује лако управљање без обзира на уређај који корисници користе.

На *Слици 5.2.8.11.* налази се приказ мобилне верзије екрана почетне стране у случају да је корисник „Гост“.

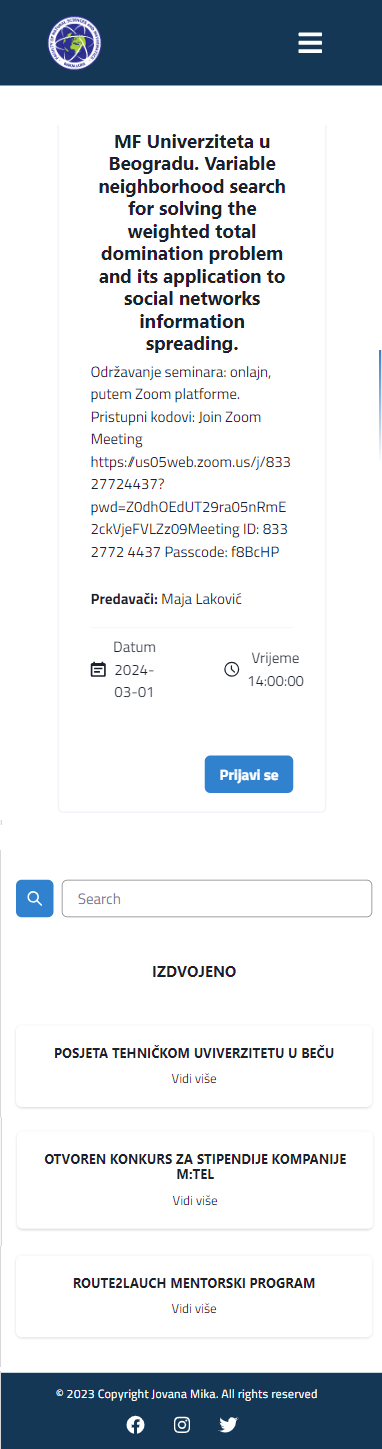


**Слика 5.2.8.11.** *Почетна страна – мобилна везија.*

На *Слици 5.2.8.12.* налази се мобилна верзија модула за приказ постова, а на *Слици 5.2.8.13.* мобилна верзија модула за приказ семинара.

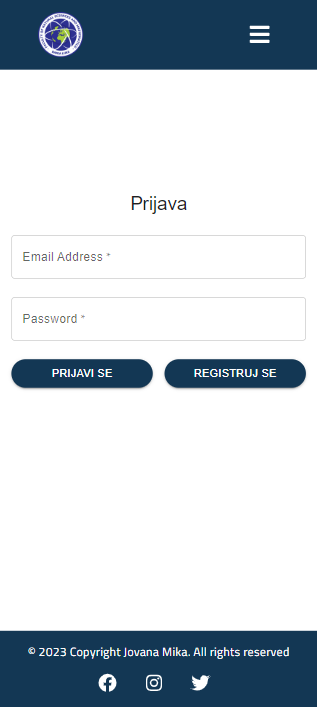
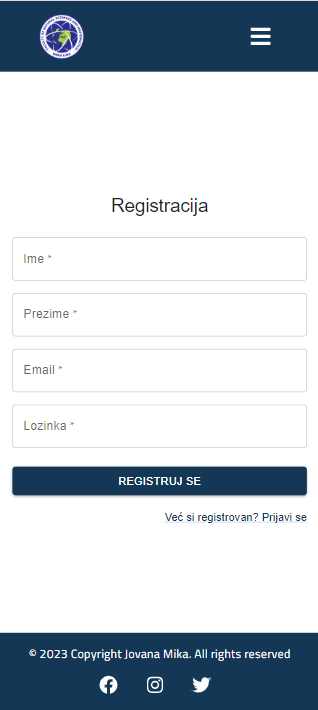


**Слика 5.2.8.12.** *Модул за приказ постова – мобилна верзија*.



**Слика 5.2.8.13.** *Модул за приказ семинара – мобилна верзија.*

На *Сликама 5.2.8.14.* и *5.2.8.15.* приказане су мобилне верзије модула за пријаву корисника и модула за регистрацију корисника.

**Слика 5.2.8.15.** *Модул за регистрацију – мобилна верзија.*

**Слика 5.2.8.14.** *Модул за пријаву корисника – мобилна верзија.*

# 6. Закључак

Развој интернета доводи до све веће популаризације веб апликација. Корисници приступају овим апликацијама путем веб прегледача, што елиминише потребу за ограничавање на једну платформу.

Као што је уобичајено у данашње вријеме, приступ апликацији омогућен је са различитих уређаја повезаних на интернет, тј. уређаја који посједују веб претраживач. Значајна предност јесте да корисници не морају чувати податке на својим уређајима, с обзиром да су сви подаци са којима раде складиштени на серверу.

Развијена веб апликација за руковођење научним семинаром представља корак ка унапређењу процеса организације научних семинара, нудећи модерно и кориснички оријентисано рјешење за ефикасно вођење и праћење ове врсте образовних и истраживачких активности.

Апликација која је приказана у оквиру овог рада нуди могућности за будућа унапређења. Један од следећих корака би могао бити додавање функционалности коментарисања научних објава, што би додатно обогатило интеракцију корисника са платформом. Такође, могућност организовања виртуелних семинара директно путем апликације могло би додатно побољшати квалитет ове апликације. Све ове могућности чине веб апликацију скалабилном и прилагодљивом за будуће иновације у области организације научних семинара.

# 7. Литература

[1] [*http://www.eknfak.ni.ac.rs/srl/Naucni-seminar.php*](http://www.eknfak.ni.ac.rs/srl/Naucni-seminar.php)  
[2] *https://imi.pmf.kg.ac.rs/*  
[3] Leon Shklar, Richard Rosen: *Web Application Architecture*.  
[4] Cory Gackenheimer: *Introduction to React*.  
[5] Naimul Islam Naim: *React JS: An Open Source JavaScript Library for Front-end Development.*[6] HTML Living Standard: [*https://html.spec.whatwg.org/*](https://html.spec.whatwg.org/) [приступано 16.12.2023].  
[7] MDN contributors, “What is CSS?”: [*https://developer.mozilla.org/en-US/docs/Learn/CSS/First\_steps/What\_is\_CSS*](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS) [приступано 16.11.2023.].  
[8] Eric A. Meyer: *CSS:The Definitive Guide, Third Edition*.  
[9] ReactJS.org, ‘ReactJS official’: *http://www.ReactJs.org* [приступано 16.11.2023.].  
[10] Doel Sengupta, Manu Sighal, Danillo Corvalan: *Getting Started with React.*  
[11] Alex Banks, Eve Porcello: *Learning React – Functional Web Development with React and Redux.*  
[12] ReactJS.org, 'Introducing JSX': [*https://legacy.reactjs.org/docs/introducing-jsx.html*](https://legacy.reactjs.org/docs/introducing-jsx.html)[ приступано 29.11.2023.].  
[13] K. Siva Prasad Reddy: *Beginning Spring Boot 2: Applications and Microservices with the Spring Framework.*[14] Visual Studio  
[15] Oracle Corporation, „The Java Programming Language“: [*https://docs.oracle.com/javase/specs/jls/se8/html/*](https://docs.oracle.com/javase/specs/jls/se8/html/) [приступано 22.11.2023].   
[16] Spring Boot Official Documentation: [*https://docs.spring.io/spring-boot/docs/1.1.4.RELEASE/reference/html/getting-started-first-application.html*](https://docs.spring.io/spring-boot/docs/1.1.4.RELEASE/reference/html/getting-started-first-application.html) [приступано 30.11.2023.].  
[17]Apache Maven Project, „What is Appache Maven?“ : [*https://maven.apache.org/what-is-maven.html*](https://maven.apache.org/what-is-maven.html) [приступано 22.11.2023.].  
[18] Eclipse Foundatio, „Eclipse IDE“ : [*https://www.eclipse.org/ide/*](https://www.eclipse.org/ide/)[приступано 30.11.2023.].  
[19] Joshua D. Drake, John C. Worslay: *Practical PostgreSQL*.  
[20] Spring Framework, Getting szarted with Spring Boot: [*https://spring.io/guides/gs/spring-boot/*](https://spring.io/guides/gs/spring-boot/)[приступано 01.12.2023.]  
[21] Npx – npm: [*https://www.npmjs.com/package/npx*](https://www.npmjs.com/package/npx) [приступано 06.12.2023.]