|  |
| --- |
| Stručni kurs Razvoj bezbednog softvera |
| Izveštaj |
| Pronađene ranjivosti u projektu “RealBookStore” |

|  |
| --- |
| Jelena Milošević |

# Istorija izmena

|  |  |  |  |
| --- | --- | --- | --- |
| Verzija | Datum | Izmenio/la | Komentar |
| 1.0 | 28.2.2024. | Uroš Dragojević | Kreiran izveštaj |
| 1.1 | 8.9.2025. | Jelena Milošević | Primena alata za statičku analizu |
| 1.2 | 9.9.2025. | Jelena Milošević | SQL injection |
| 1.3 | 9.9.2025. | Jelena Milošević | Cross-site scripting |
| 1.4 | 9.9.2025. | Jelena Milošević | Cross-site request forgery |
| 1.5 | 11.9.2025. | Jelena Milošević | Autorizacija |
| 1.6 | 12.9.2025. | Jelena Milošević | DevOps |

Sadržaj

[Istorija izmena 1](#_Toc160050508)

[Uvod 4](#_Toc160050509)

[O veb aplikaciji 4](#_Toc160050510)

[Kratak pregled rezultata testiranja 4](#_Toc160050511)

[SQL injection 5](#_Toc160050512)

[Napad: Ubacivanje novog usera u tabelu “persons” (SQL injection) 5](#_Toc160050513)

[Metod napada: 5](#_Toc160050514)

[Predlog odbrane: 6](#_Toc160050515)

[Cross-site scripting 7](#_Toc160050516)

[Napad: Ubacivanje novog usera u tabelu “persons” 7](#_Toc160050517)

[Metod napada: 7](#_Toc160050518)

[Predlog odbrane: 9](#_Toc160050519)

[Cross-site request forgery........................................................................................10](#_Cross-site)

[Napad: Menjanje ličnih podataka..........................................................................10](#_Napad:_Men)

[Metod napada:.................................................................................................10](#_Metod_napa)

[Predlog odbrane:..................................................................................................11](#_Predlog_od)

[Implementacija autorizacije.....................................................................................12](#_Implementa)

[DevOps...................................................................................................................14](#_DevOps
)

[Zaključak.................................................................................................................15](#_Zaključak)

# Uvod

Ovaj izveštaj se bavi ranjivostima pronađenim u dole opisanoj veb aplikaciji.

## O veb aplikaciji

**RealBookStore** je veb aplikacija koja pruža mogućnosti pretrage, ocenjivanja i komentarisanja knjiga.

Aplikacija RealBookStore omogućava sledeće:

* Pregled i pretragu knjiga.
* Dodavanje nove knjige.
* Detaljan pregleda knjige kao i komentarisanje i ocenjivanje knjige.
* Pregled korisnika aplikacije.
* Detaljan pregled podataka korisnika.

## 

## Kratak pregled rezultata testiranja

Rezultati primene alata za statičku analizu **SonarQube**:

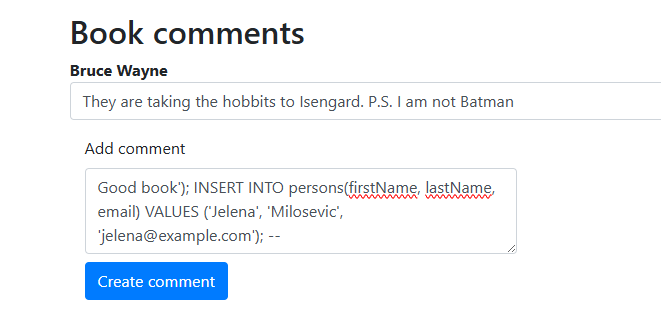
|  |  |  |
| --- | --- | --- |
| *Pronađene ranjivosti* | *Nivo opasnosti* | *Broj ranjivosti* |
| *Cross-site request forgery* | *High* | *1* |
| *SQL injection* | *High* | *20* |
| *Insecure Configuration* | *Low* | *20* |

# SQL injection

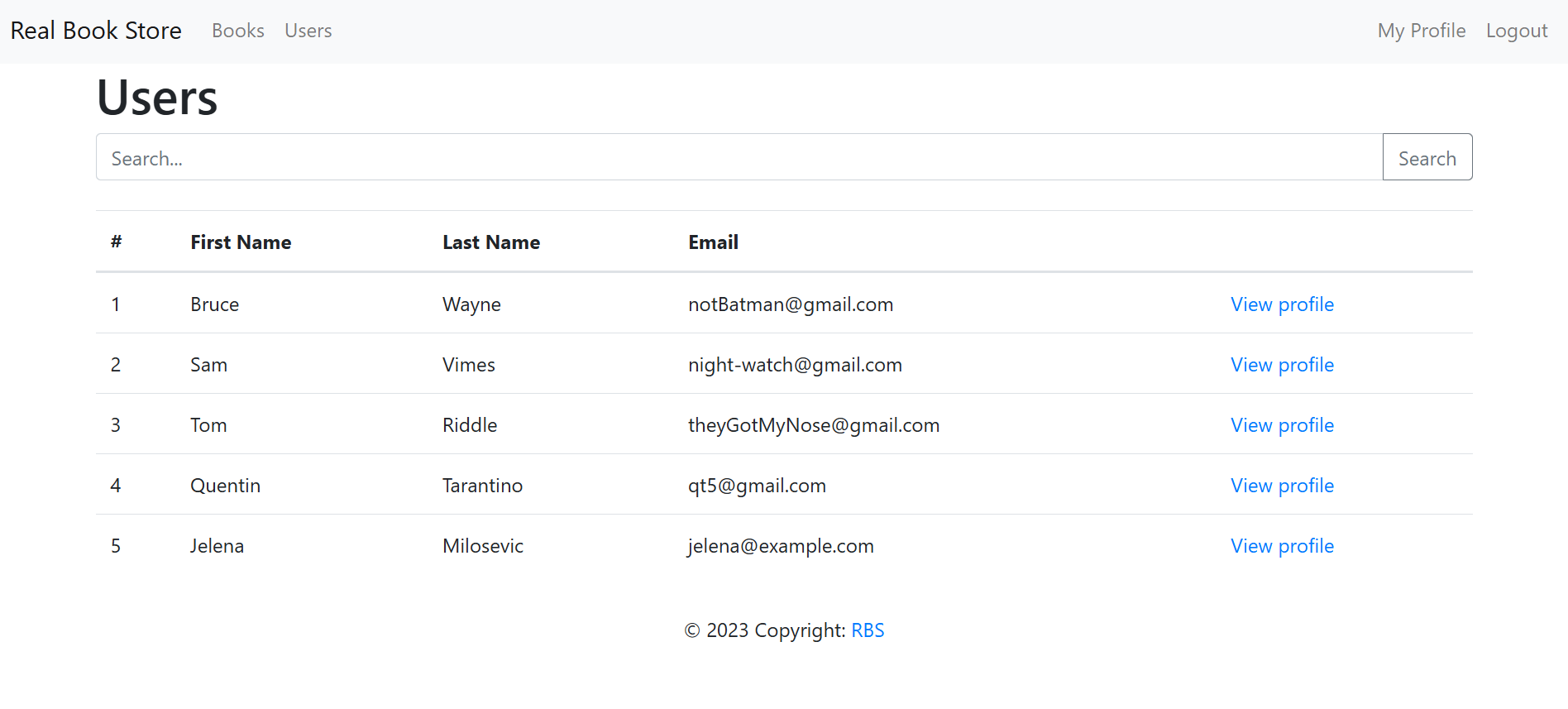
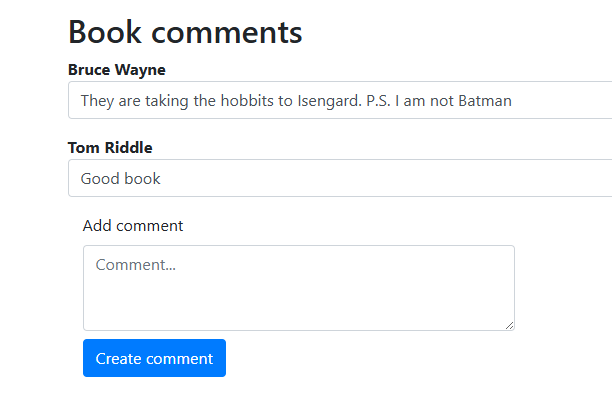
## Napad: Ubacivanje novog usera u tabelu “persons” (SQL injection)

### Metod napada:

Na stranici za pregledanje pojedinačne knjige, uneti sledeći kod u input polje “comment”:



Prikaz:



## Predlog odbrane:

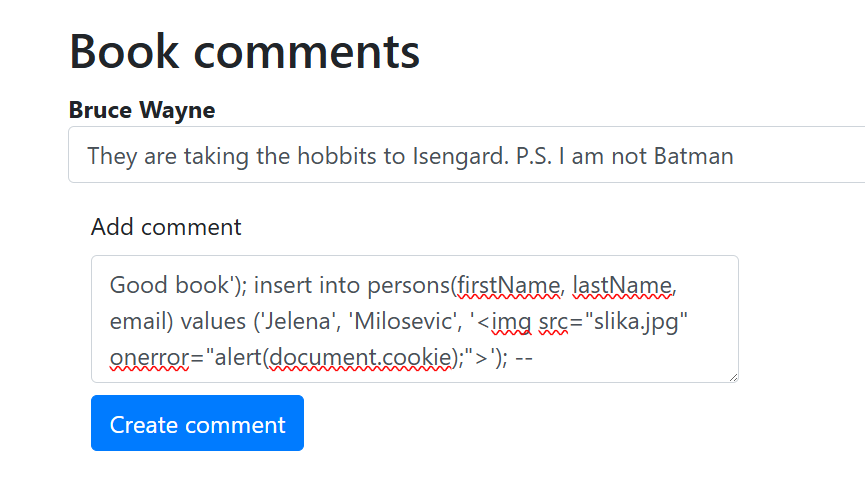
Koristiti PreparedStatement umesto Statement, odnosno koristiti parametrizovane upite.

# Cross-site scripting

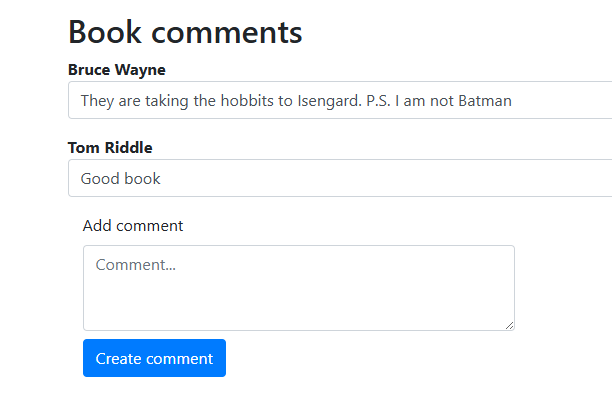
## Napad: Ubacivanje novog usera u tabelu “persons”

### Metod napada:

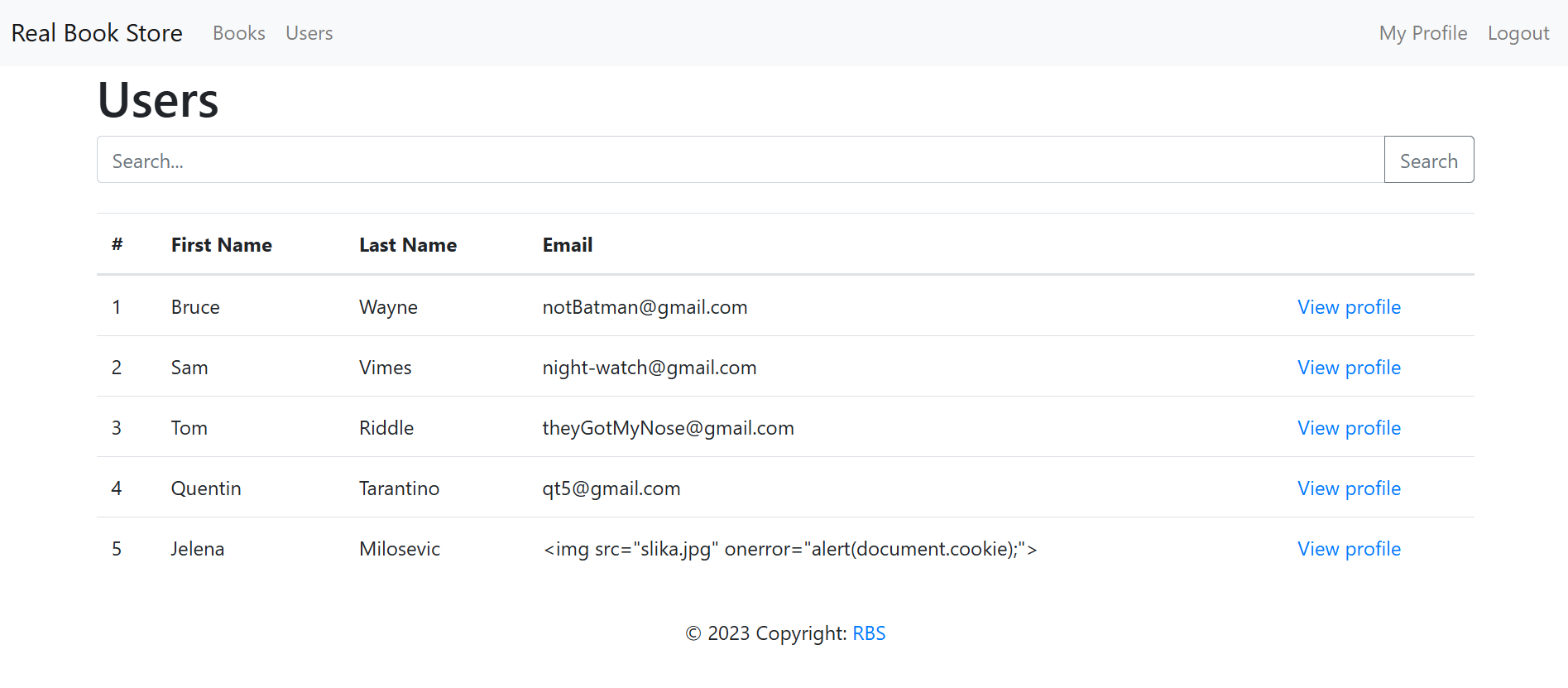
Na stranici za pregledanje pojedinačne knjige, uneti sledeći kod u input polje “comment” (kombinujemo sa SQL Injection-om):



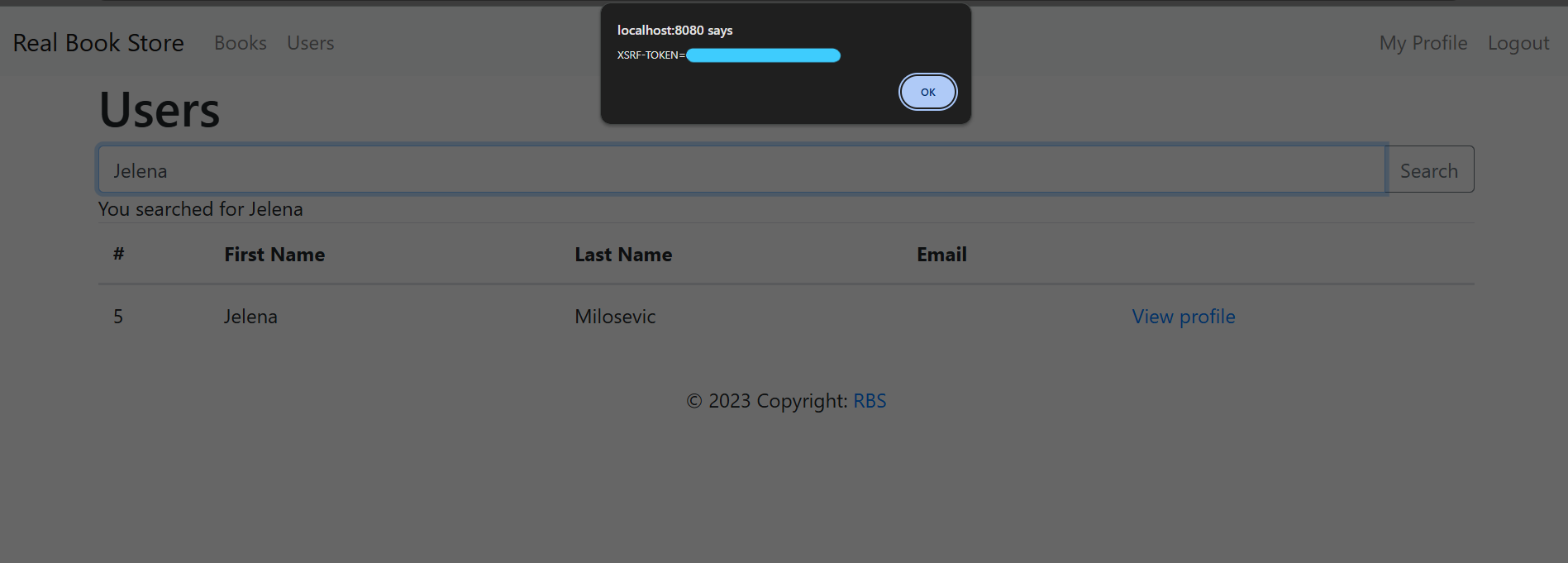
Prikaz komentara:



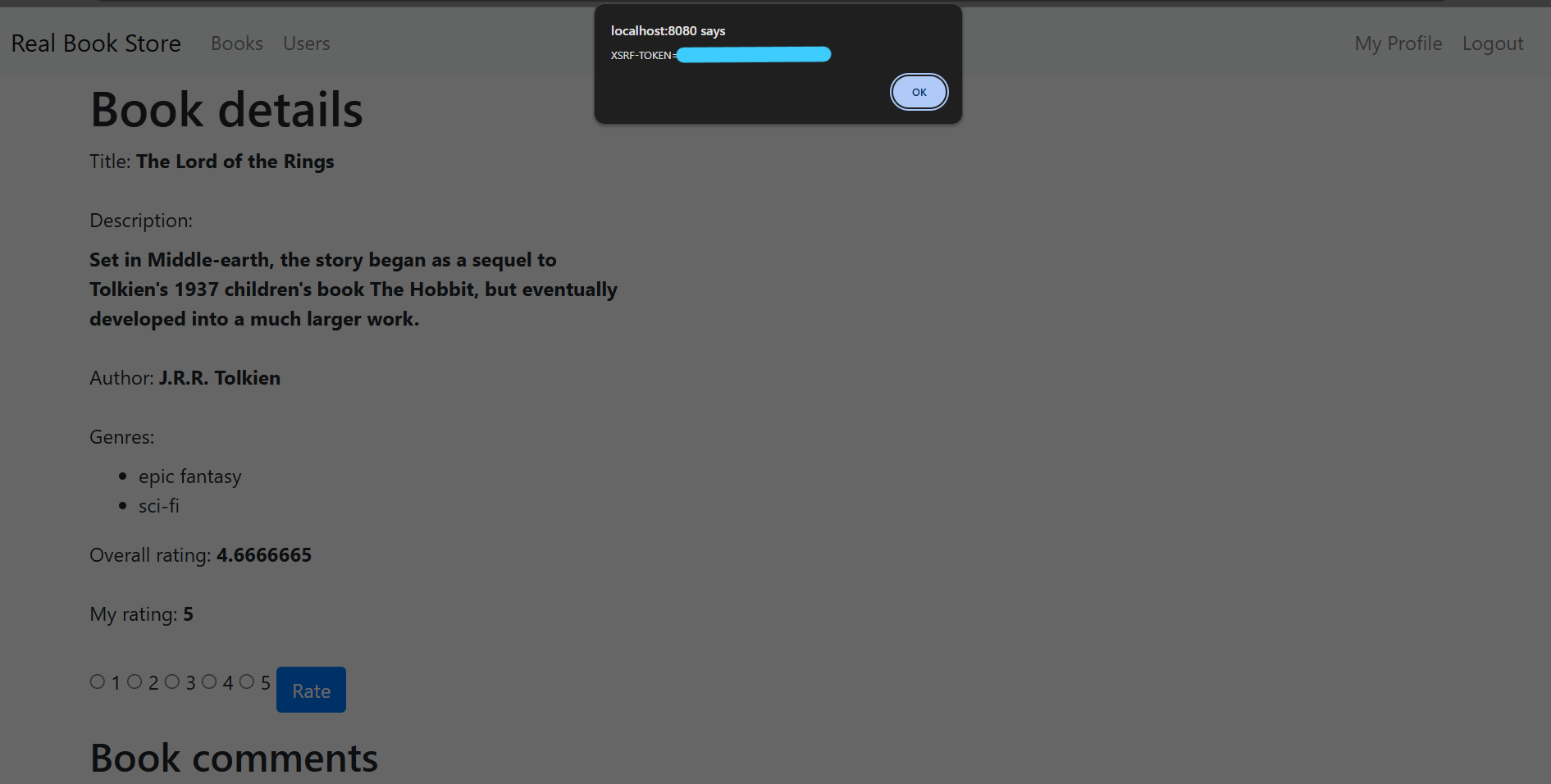
Prikaz stranice sa korisnicima (Users):



Prikaz nakon pretrage ubačenog korisnika u polje za pretragu (Search bar) na stranici sa korisnicima (pre popravke SQL Injection):



Prikaz nakon unetog komentara (nakon popravke SQL Injection):



## Predlog odbrane:

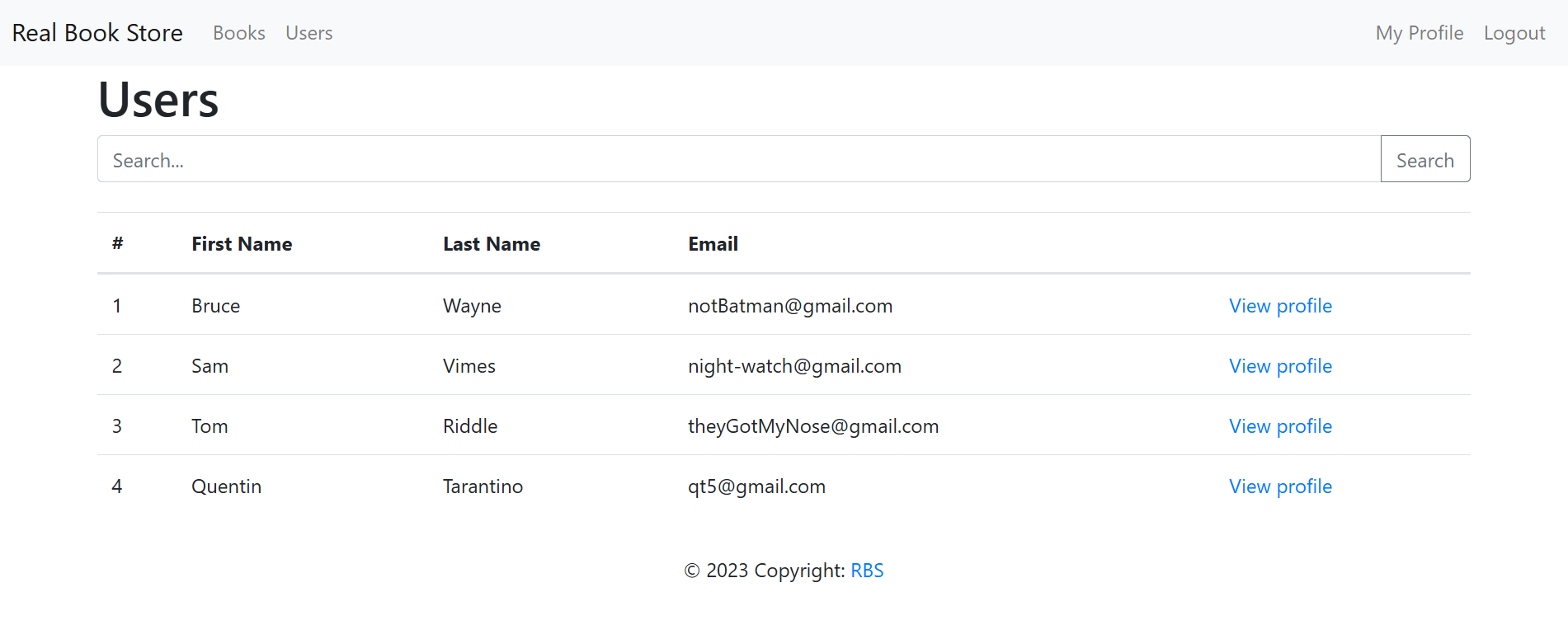
Koristiti textContent umesto innerHTML i koristiti th:text umesto th:utext na HTML tagu koji prikazuje polje "comment".

# Cross-site request forgery

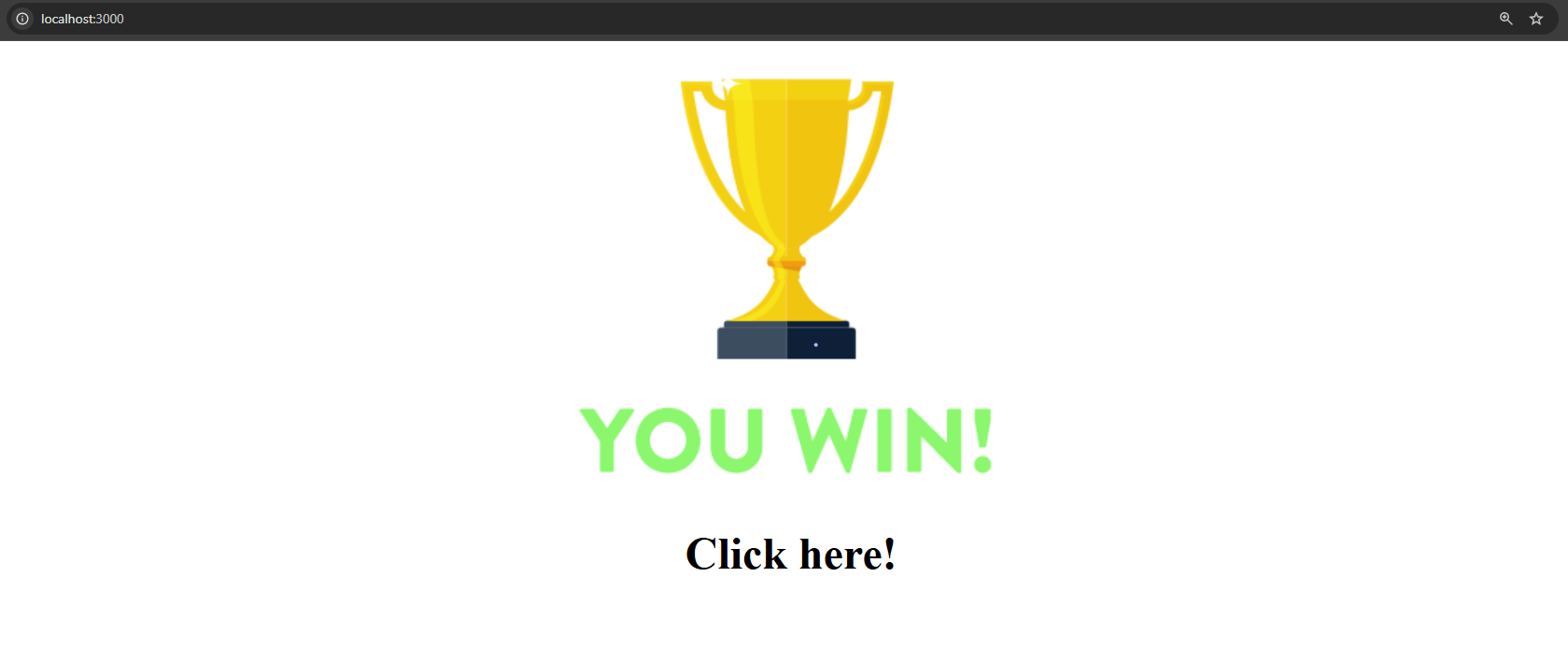
## Napad: Menjanje ličnih podataka.

### Metod napada:

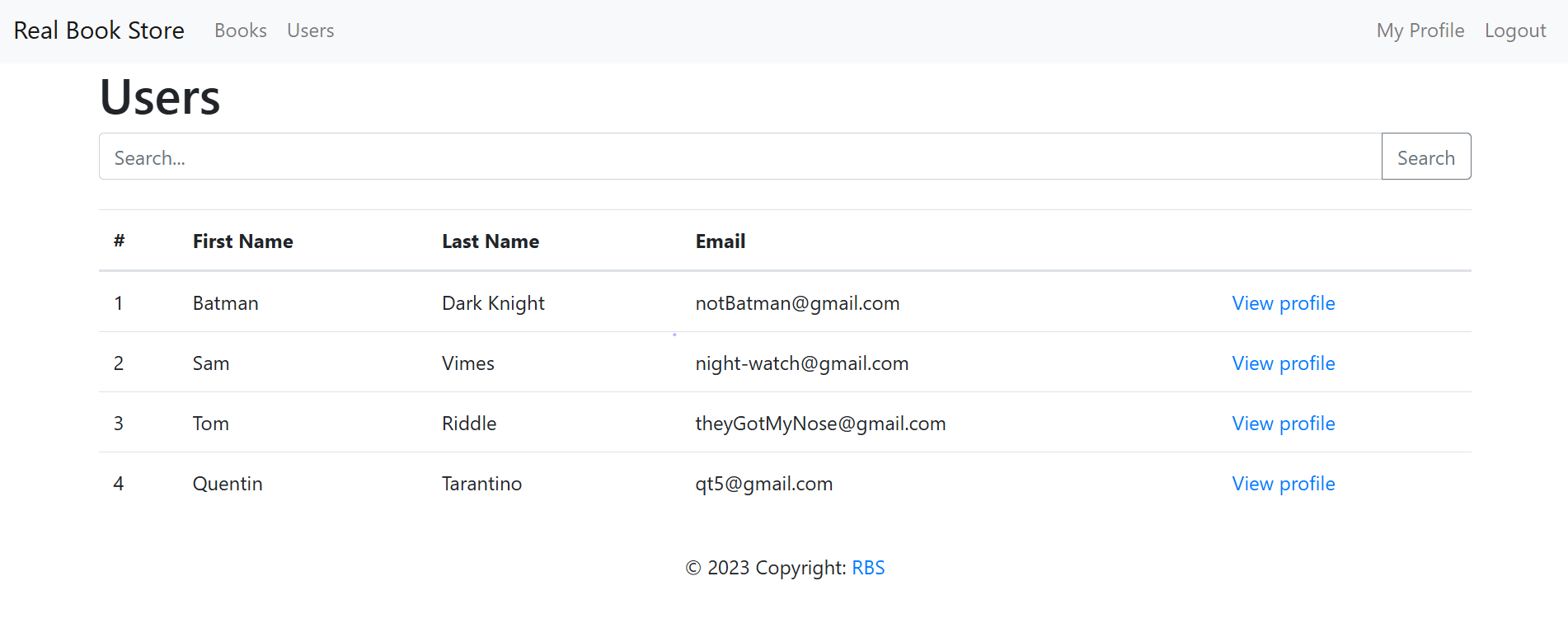
1. Stanje pre izvršenja napada:



2) Napisana je funkcija exploit() za demonstracionu stranicu index.html.

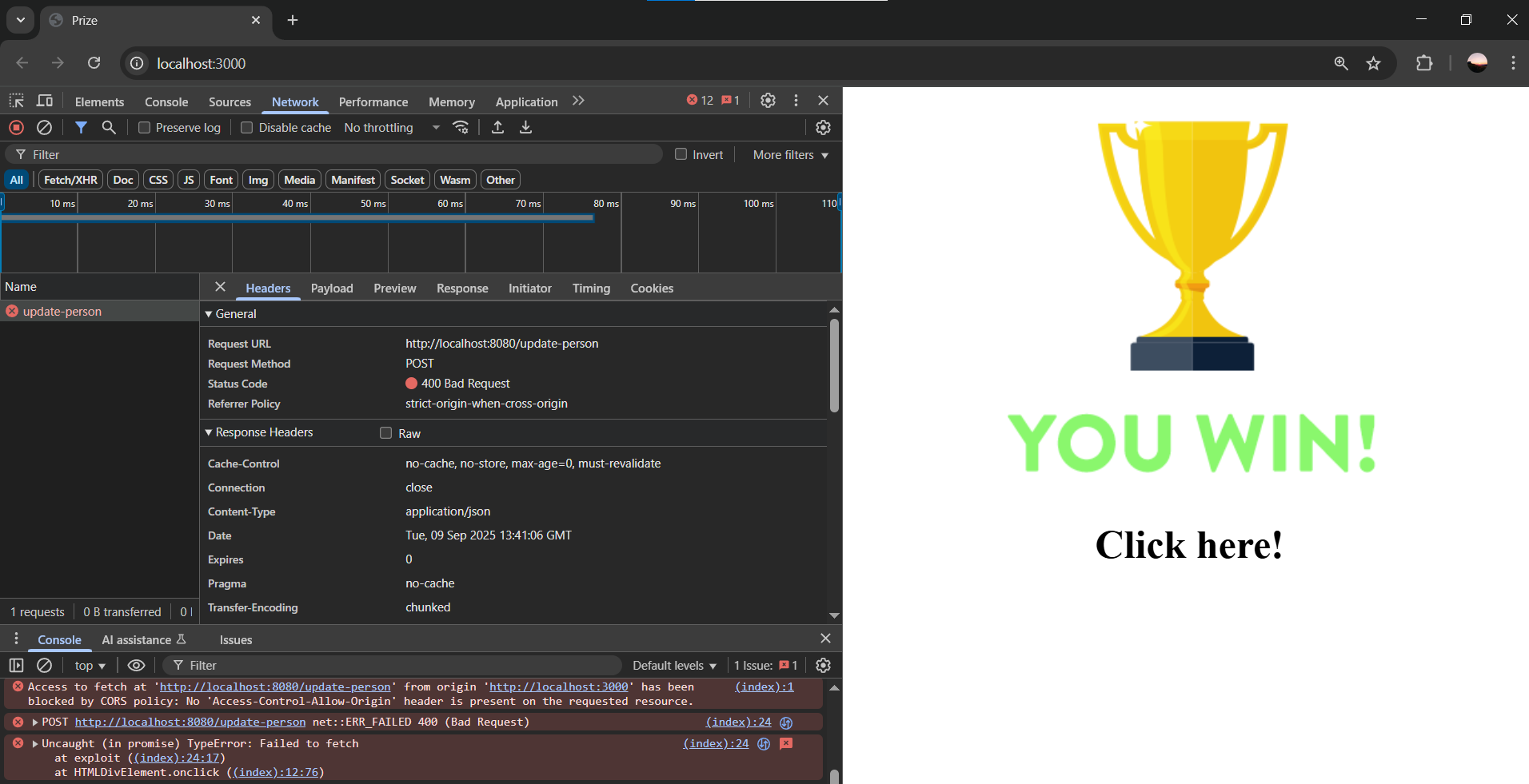


3) Kada korisnik klikne na prikazani link („Click here!“), pokreće se skripta koja šalje neovlašćeni POST zahtev ka serveru. Aplikacija bez zaštite protiv CSRF napada prihvata zahtev i menja podatke korisnika sa id = 1, tako da je sada firstName = “Batman” i lastName = “Dark Knight”.



## Predlog odbrane:

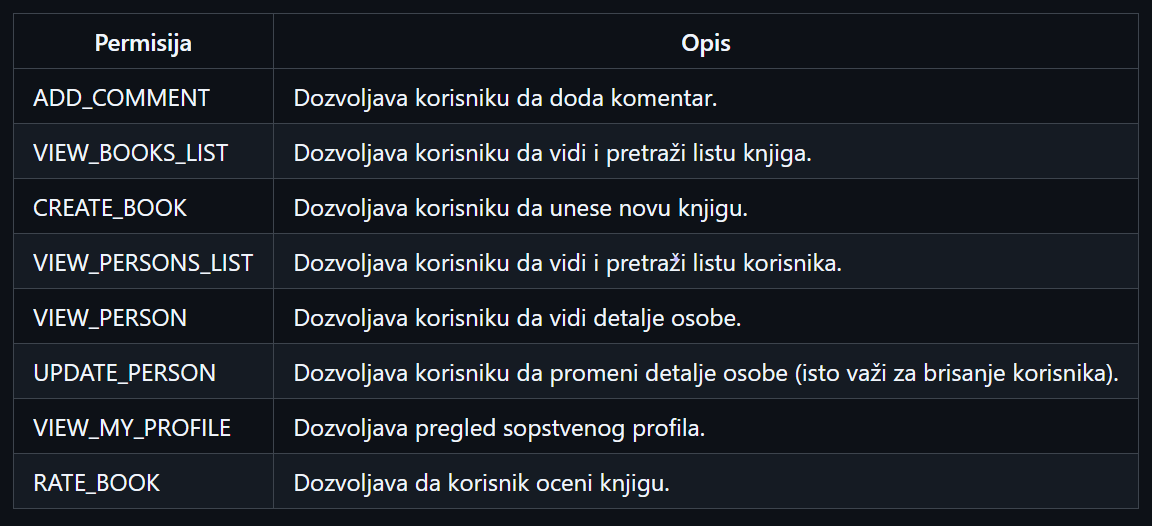
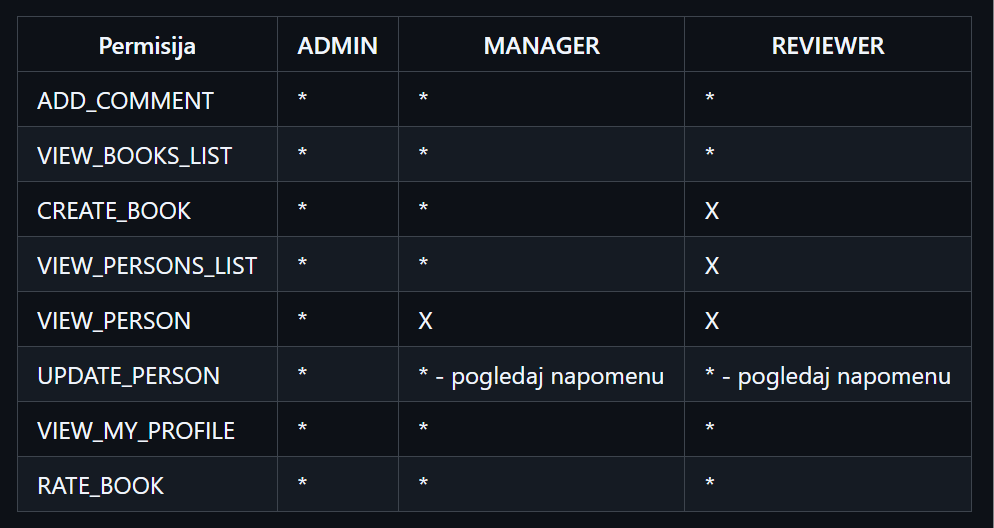
Koristiti CSRF token (Synchronizer Token Pattern). Poželjno je uključiti i CORS zaštitu na backend-u (ali sama po sebi nije dovoljna protiv CSRF).



# Implementacija autorizacije

Korišćeni su **Spring Security** i **Thymeleaf** koncepti.

Implementiran je autorizacioni model (matrica permisija) u bazi podataka.

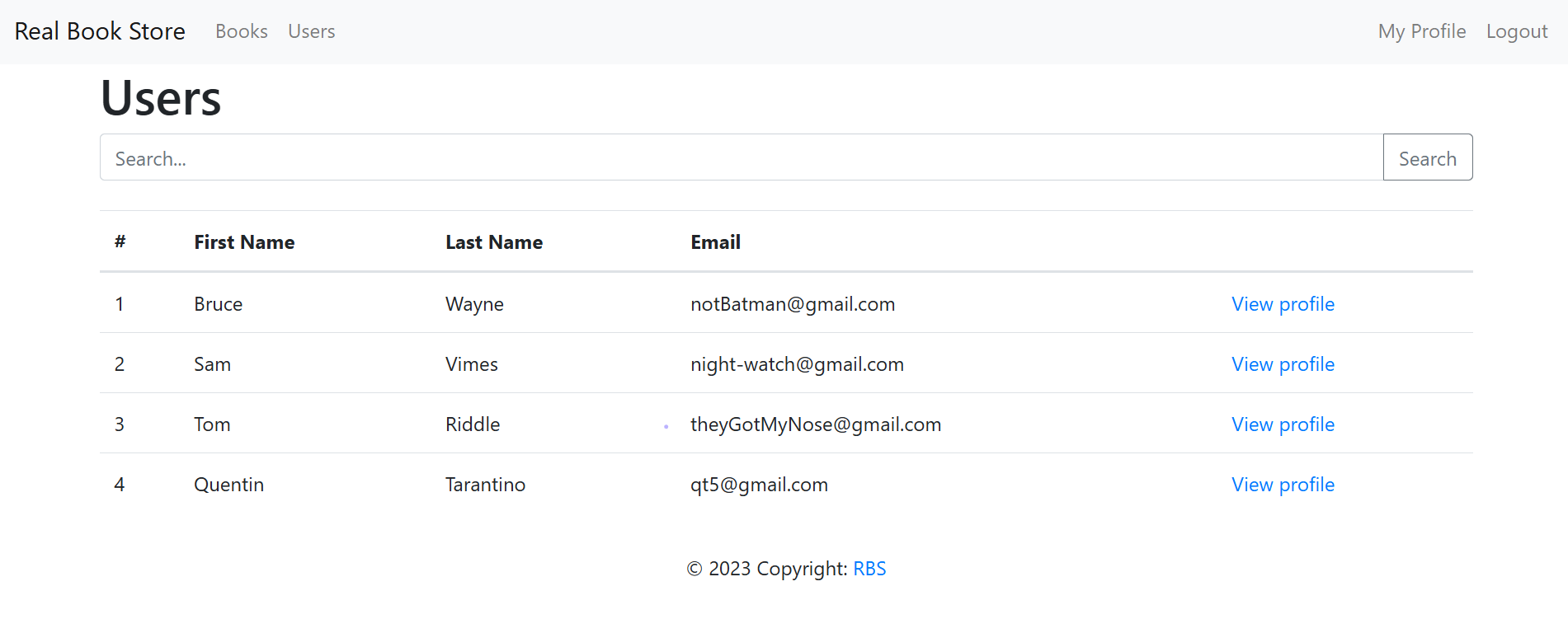


*Napomena: Rolama MANAGER i REVIEWER je dozvoljena promena isključivo svojih sopstvenih podataka.*

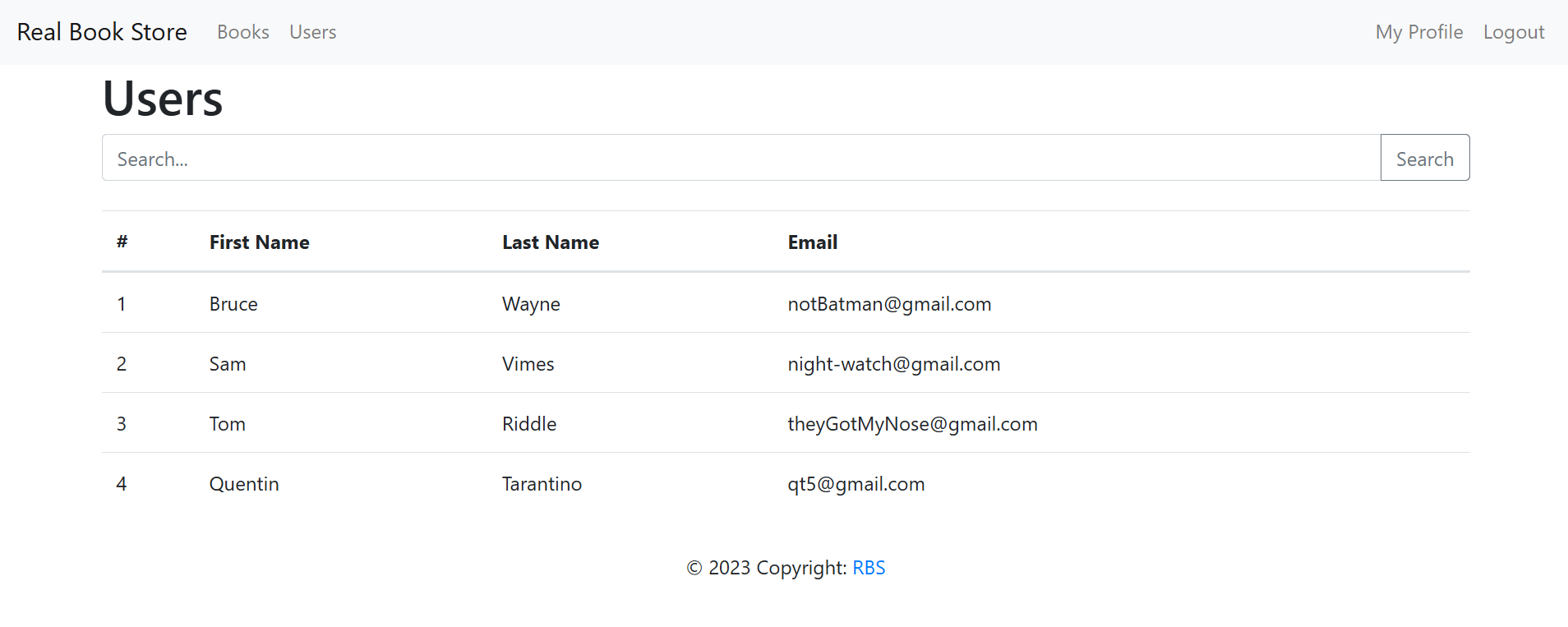
Postavljeno je da korisnik:

* “tom” ima rolu ADMIN
* “toeLover” rolu MANAGER
* “bruce” ima rolu REVIEWER
* “sam” ima rolu REVIEWER.

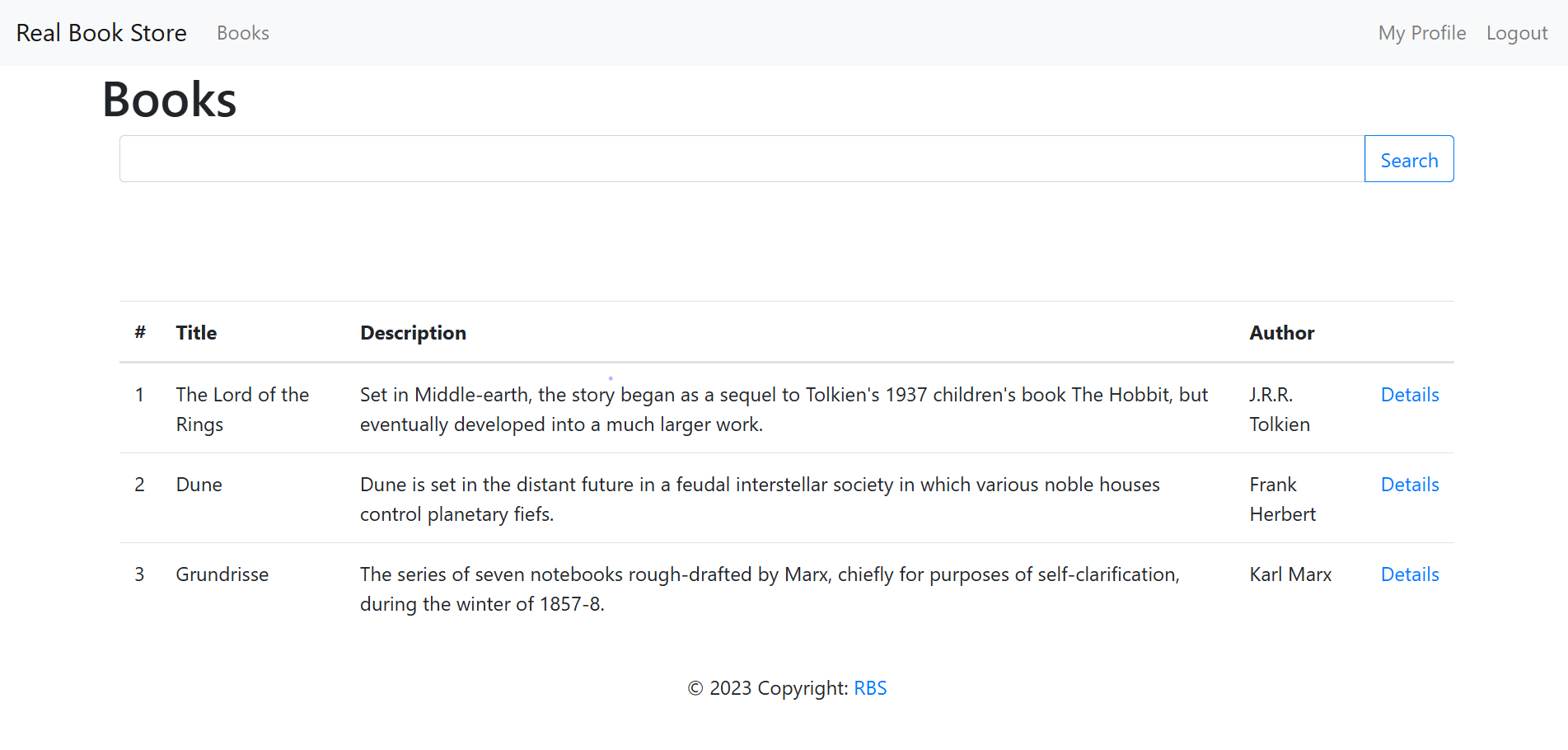
**Perspektiva ADMIN**:



**Perspektiva MANAGER**:



**Perspektiva REVIEWER**:



# DevOps

Za praćenje promena i grešaka u aplikaciji uvedeno:

**Logovanje**: Centralizovano beleženje svih izuzetaka i kritičnih događaja u aplikaciji radi lakše analize grešaka i potencijalnih napada.

**Auditing**: Praćenje ključnih operacija (kreiranje, izmena, brisanje podataka) radi evidentiranja aktivnosti korisnika i promena u sistemu.

# 

# Zaključak

* **Statička i dinamička analiza**: Alati za statičku i dinamičku analizu otkrivaju ranjivosti i pomažu u održavanju sigurnog koda.
* **SQL Injection**: Parametrizovani upiti (PreparedStatement) štite od SQL injekcija.
* **XSS**: Korišćenje th:text i textContent sprečava unos malicioznog HTML/JS koda.
* **CSRF**: CSRF tokeni (Synchronizer Token Pattern) i CORS zaštita smanjuju rizik od neautorizovanih zahteva.
* **Sigurna autentifikacija**: Spring Security obezbeđuje sigurno prijavljivanje korisnika.
* **Autorizacioni modeli**: Matrica permisija i uloge kontrolišu pristup funkcionalnostima aplikacije.
* **DevOps**: Logging i auditing omogućavaju praćenje grešaka i aktivnosti korisnika, podržavajući brzu reakciju u slučaju problema ili napada.