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

|  |
| --- |
| Ana Arsić  Aug 22, 25 |

# Istorija izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Verzija** | **Datum** | **Izmenio/la** | **Komentar** |
| **1.0** | 22.08.2025. | Ana Arsić | Kreiran izveštaj |
| **1.1** | 22.08.2025. | Ana Arsić | SQL injection/XSS |
| **1.1** | 22.08.2025 | Ana Arsić | Cross-site scripting |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

[Istorija izmena 1](#_Toc160050508)

[Uvod 3](#_Toc160050509)

[O veb aplikaciji 3](#_Toc160050510)

[Kratak pregled rezultata testiranja 3](#_Toc160050511)

[SQL injection 4](#_Toc160050512)

[Napad: Ubacivanje novog usera u tabelu “persons” (SQL injection) 4](#_Toc160050513)

[Metod napada: 4](#_Toc160050514)

[Predlog odbrane: 4](#_Toc160050515)

[Cross-site scripting 5](#_Toc160050516)

[Napad: Ubacivanje novog usera u tabelu “persons” 5](#_Toc160050517)

[Metod napada: 5](#_Toc160050518)

[Predlog odbrane: 5](#_Toc160050519)

[Zaključak 6](#_Toc160050520)

# 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

*Ovde idu kratko opisani rezultati testiranja: pronađene ranjivosti i nivo opasnosti.*

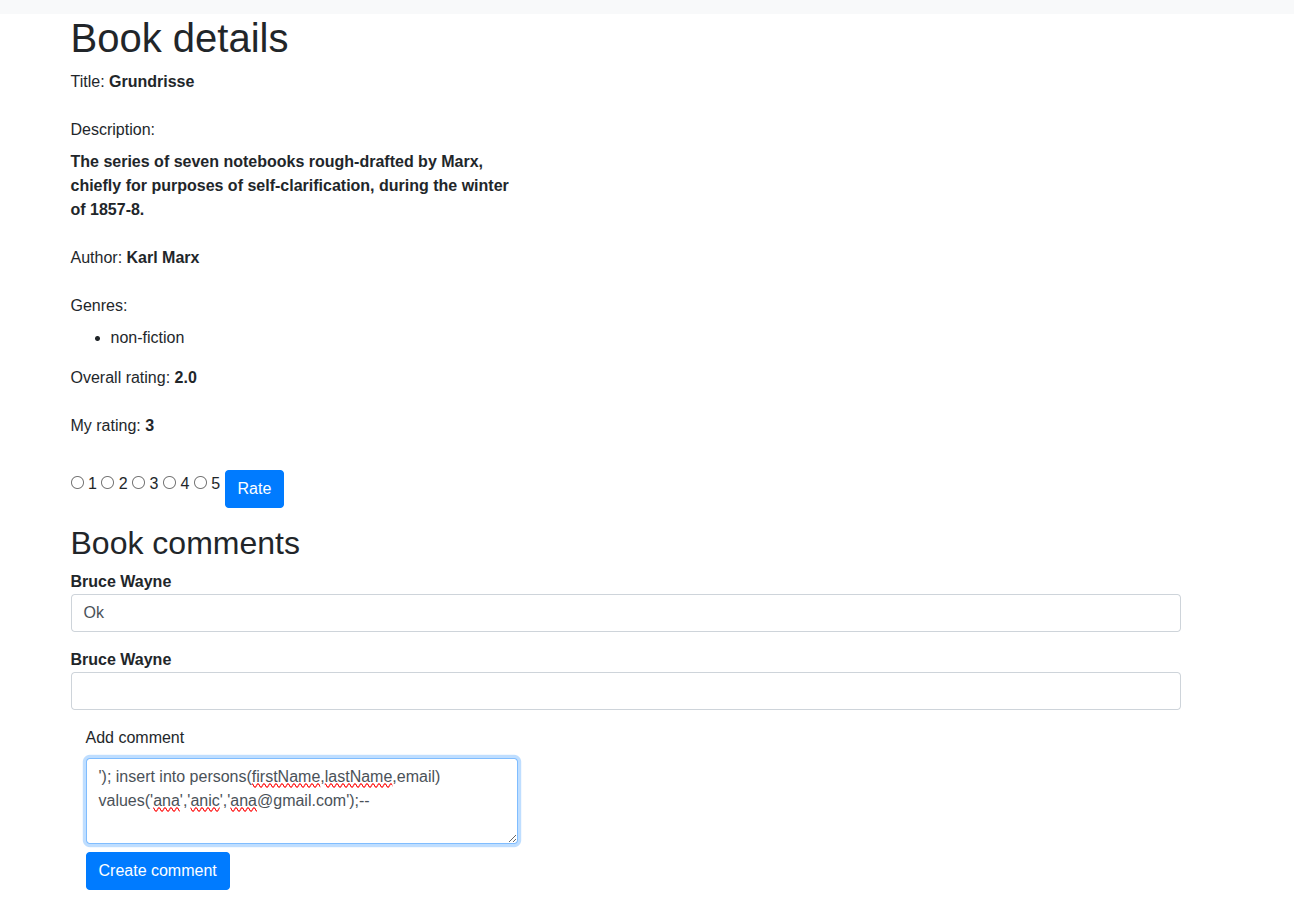
|  |  |
| --- | --- |
| ***Nivo opasnosti*** | ***Broj ranjivosti*** |
| ***Low*** | *3* |
| ***Medium*** | *2* |
| ***High*** | *1* |

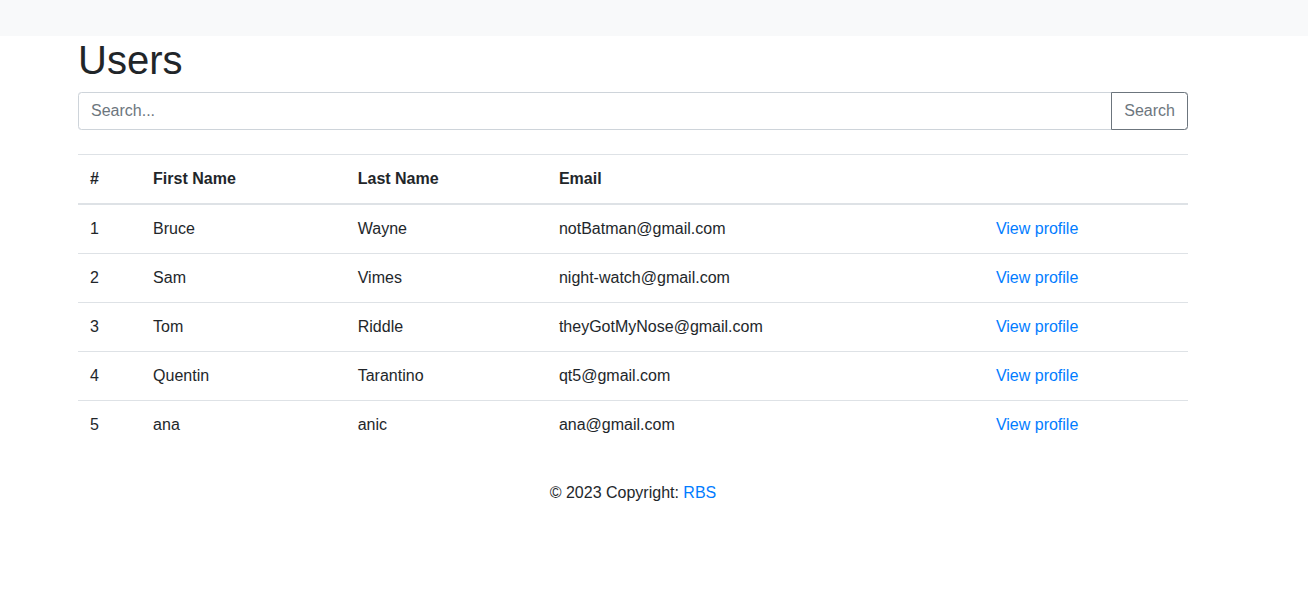
# SQL injection

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

### Metod napada:

1.SQL injection: Unosom upita u polje za komentare, pokušava se unos novog korisnika u Persons tabelu.

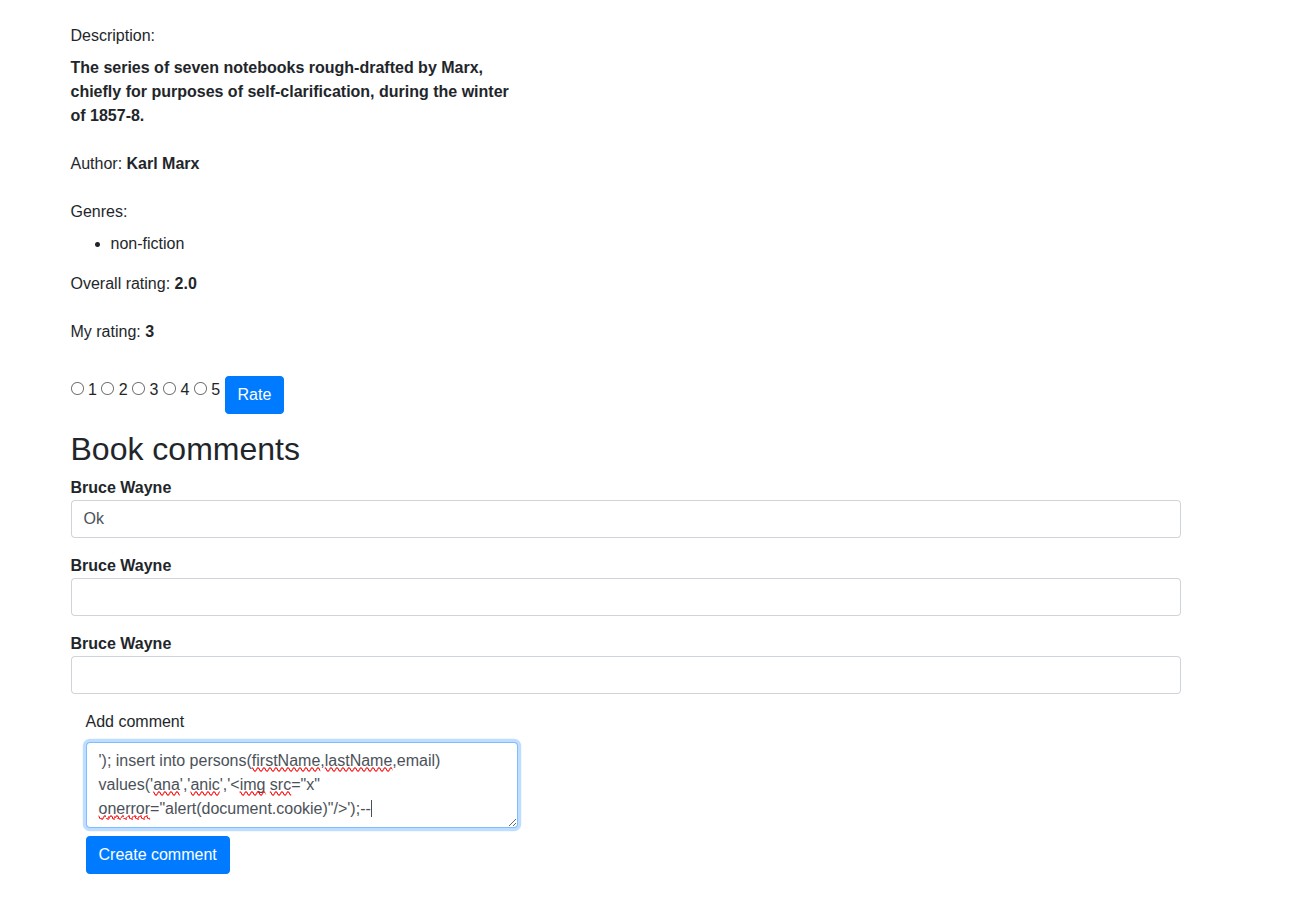
Slika 1: Unosimo komentar

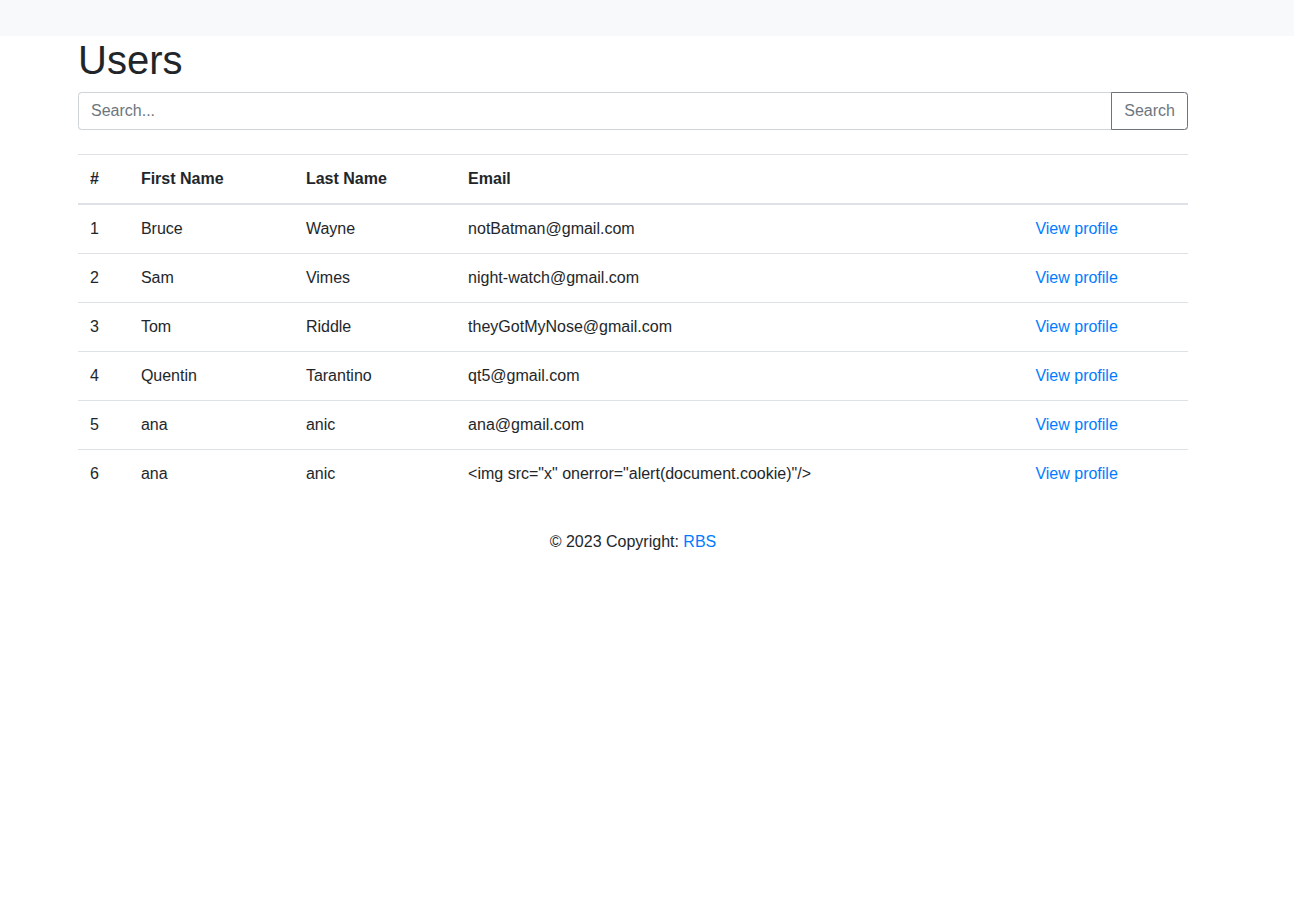
Slika 2: Novi korisnik je unet u tabelu

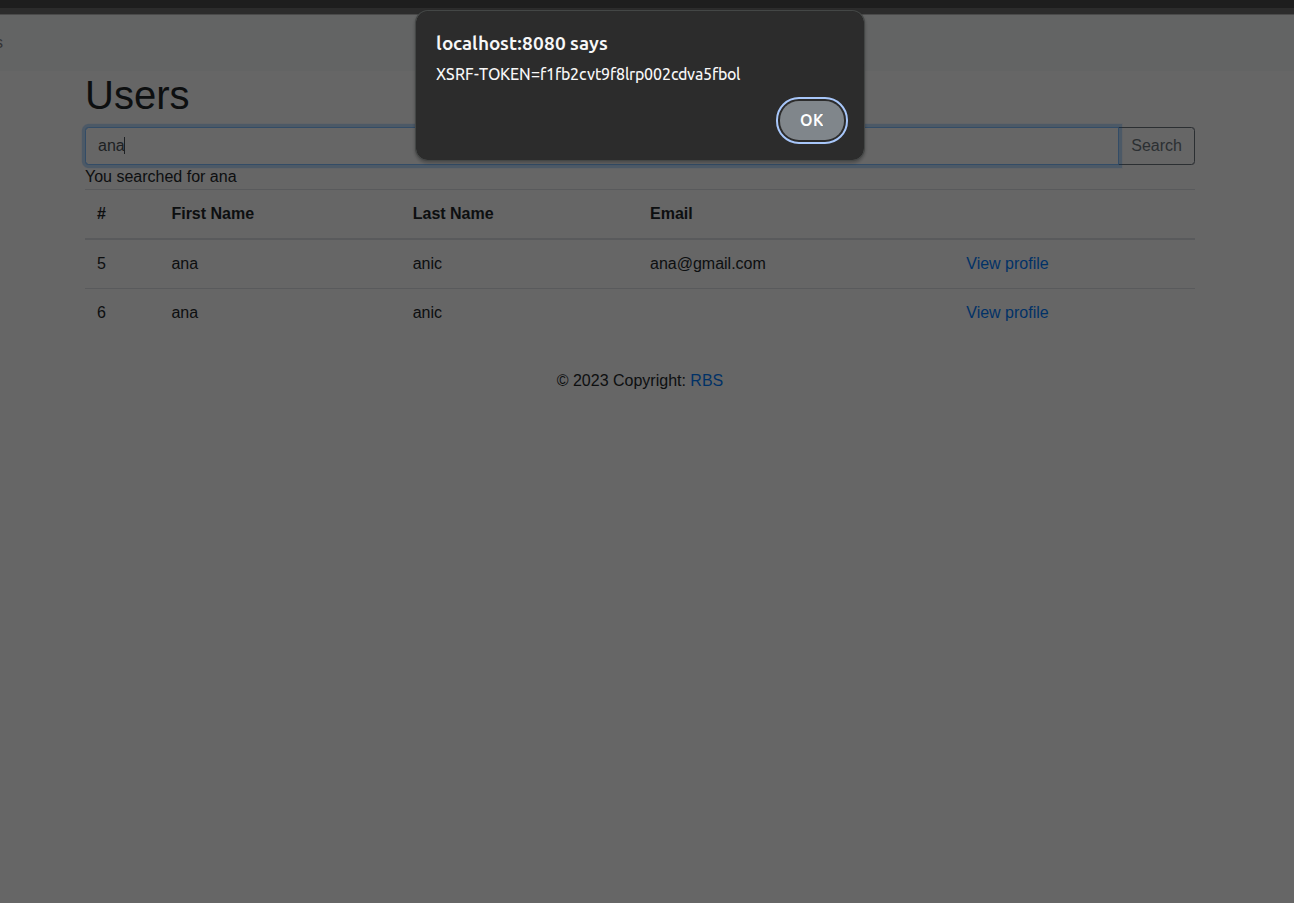
## Predlog odbrane:

Sprečavanje SQL injection-a se ostvaruje tako što se umesto klase Statement koristi klasa PreparedStatement. Ovim pristupom onemogućava se izvršavanje neovlašćenih SQL upita. Nakon izmene, komentari se i dalje uspešno upisuju u bazu, ali napadač više ne može putem unosa da doda novog korisnika.

2.XSS: U polje za komentare umesto email adrese unosimo JavaScript koji treba da prikaže kolačiće korisnika. Pokušava se sa malicioznim skriptom.

Slika 3: Unosi se maliciozni komentar

Slika 4: Novi korisnik je unet u tabelu

Slika 5: Pretraga korisnika

## Predlog odbrane:

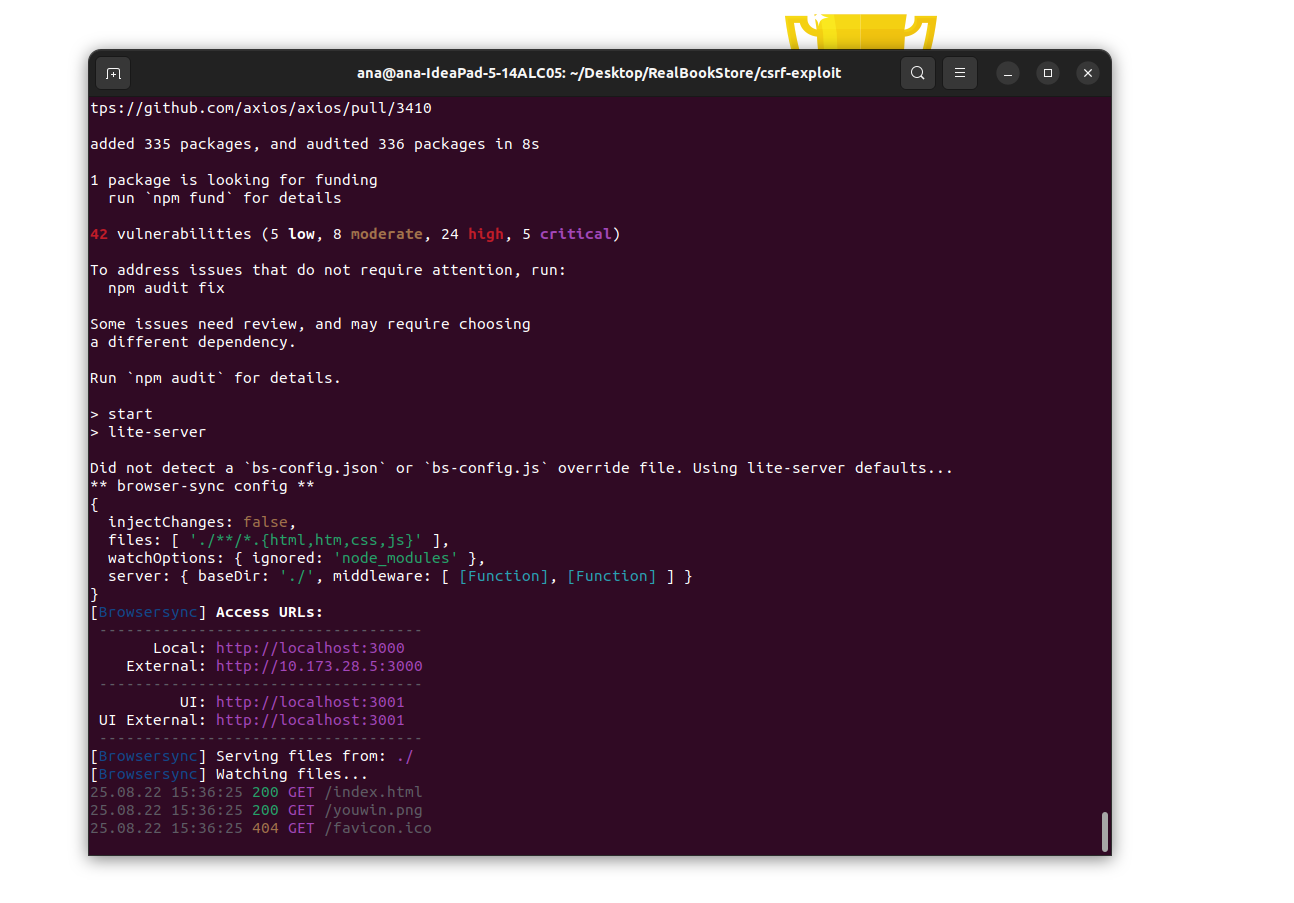
Odbrana od XSS napada podrazumeva nekoliko prilagođavanja. U fajlu persons.html umesto innerHTML koristi se textContent, čime se neutralizuje mogućnost pokretanja umetnutog JavaScript-a. Na ovaj način komentar ostaje vidljiv korisniku, ali zlonamerni kod se neće izvršiti, niti je moguće ubacivanje dodatnih korisnika.

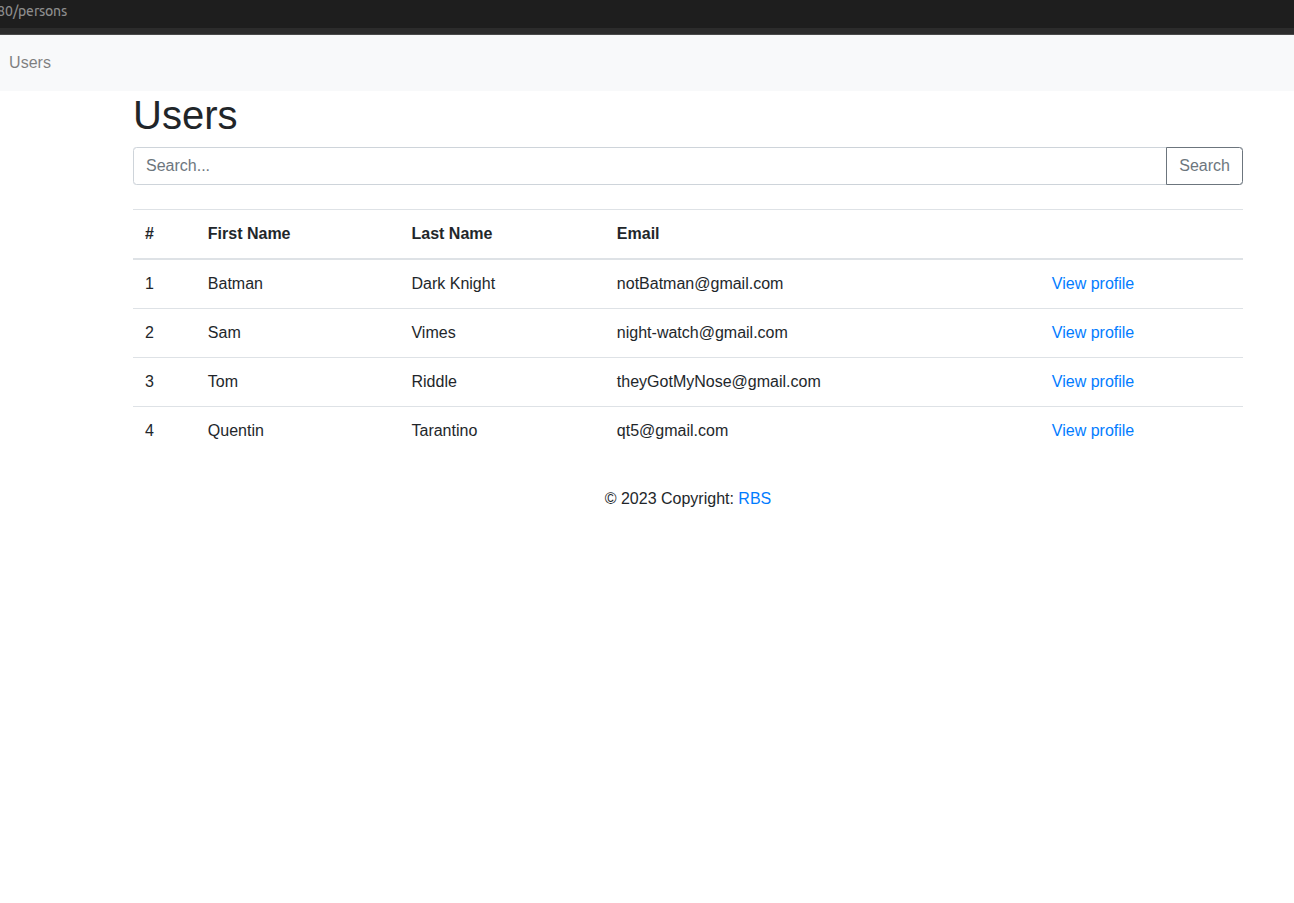
# Cross-site scripting

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

### Metod napada:

Napad treba da promeni lične podatke korisnika sa id=1 , tako da je FirstName=”Batman” i lastName=”Dark Knight”.

Slika 6: Pokrecemo skriptu iz terminala

Slika 7: Promenjeni podaci nakon napada

## Predlog odbrane:

Radi zaštite od CSRF napada u klasi PersonController, u metodi person dohvatamo CSRF token iz sesije i prosleđujemo ga kroz model. Zatim, u metodi updatePerson proveravamo da li se token koji je stigao iz forme poklapa sa onim koji je sačuvan u sesiji. Pored toga, u samu formu za izmenu podataka o korisniku dodajemo posebno input polje u kojem se čuva vrednost CSRF tokena.

# Zaključak

Tokom testiranja utvrđeno je da aplikacija sadrži određene ranjivosti koje mogu ugroziti sigurnost podataka i funkcionalnost sistema (SQL injection, XSS, CSRF). Primena zaštitnih mehanizama poput korišćenja PreparedStatement-a, enkodiranja sadržaja, validacije unosa i CSRF tokena značajno smanjuje mogućnost zloupotrebe.

Kao sledeći korak preporučuje se:

* sprovođenje detaljnijih bezbednosnih testova (penetration testing),
* uvođenje dodatnih slojeva zaštite (npr. autentikacija sa višestrukim faktorom, validacija podataka na serverskoj strani),
* redovno ažuriranje zavisnosti i biblioteka,
* kao i stalno praćenje bezbednosnih preporuka i smernica.

Na ovaj način aplikacija može dostići viši nivo sigurnosti i otpornosti na potencijalne napade.