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

|  |
| --- |
| Uroš Dragojević  05.08.2024 |

# Istorija izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Verzija** | **Datum** | **Izmenio/la** | **Komentar** |
| **1.0** | 05.08.2024. | Lazar Dunjić | Kreiran izveštaj |
| **1.1** | 05.08.2024. | Lazar Dunjić | SQL injection/XSS |
| **1.2** | 05.08.2024. | Lazar Dunjić | Cross-site scripting |
|  |  |  | [12](#_Toc160050520) |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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.*

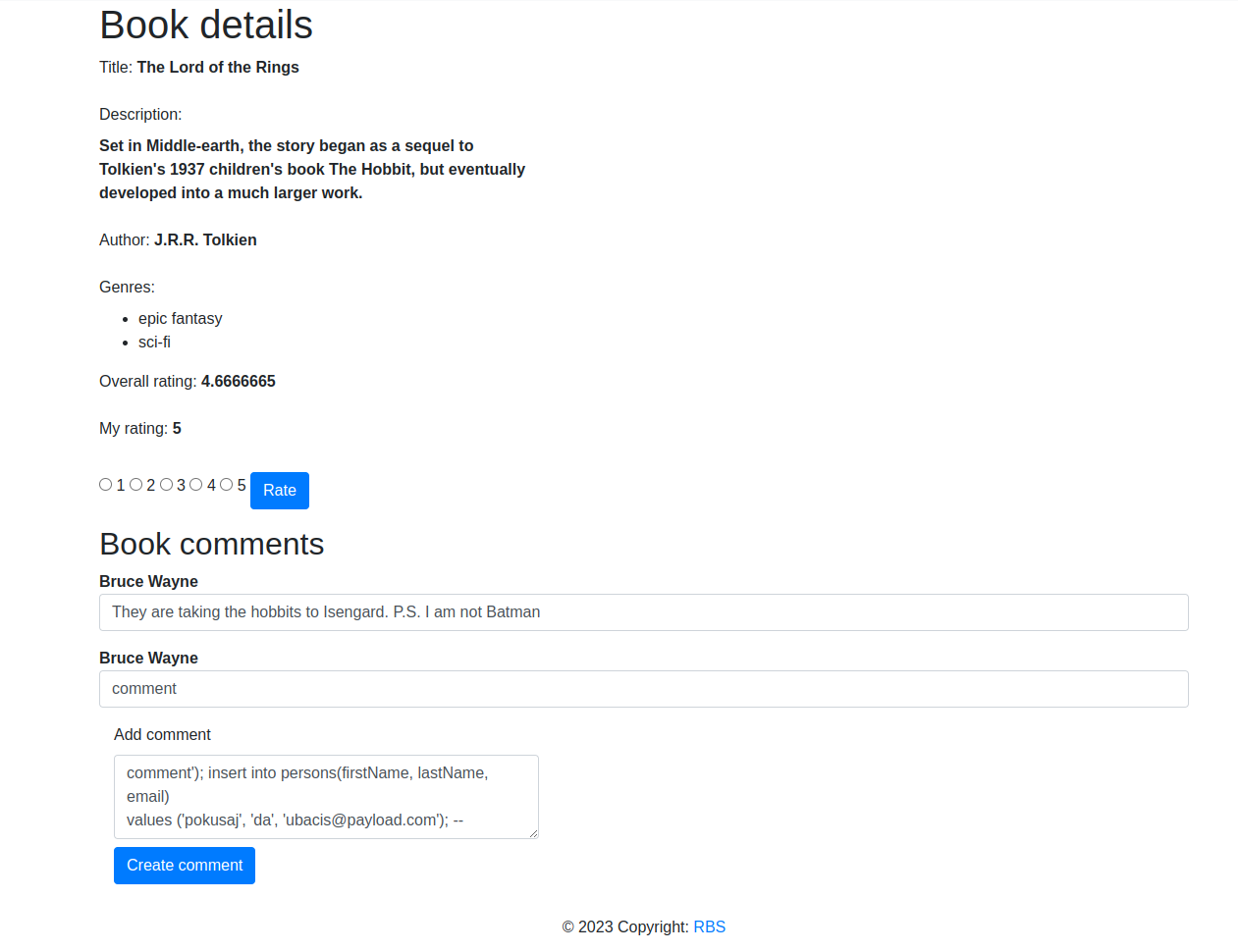
|  |  |
| --- | --- |
| ***Nivo opasnosti*** | ***Broj ranjivosti*** |
| ***Low*** | *3* |
| ***Medium*** | *2* |
| ***High*** | *1* |

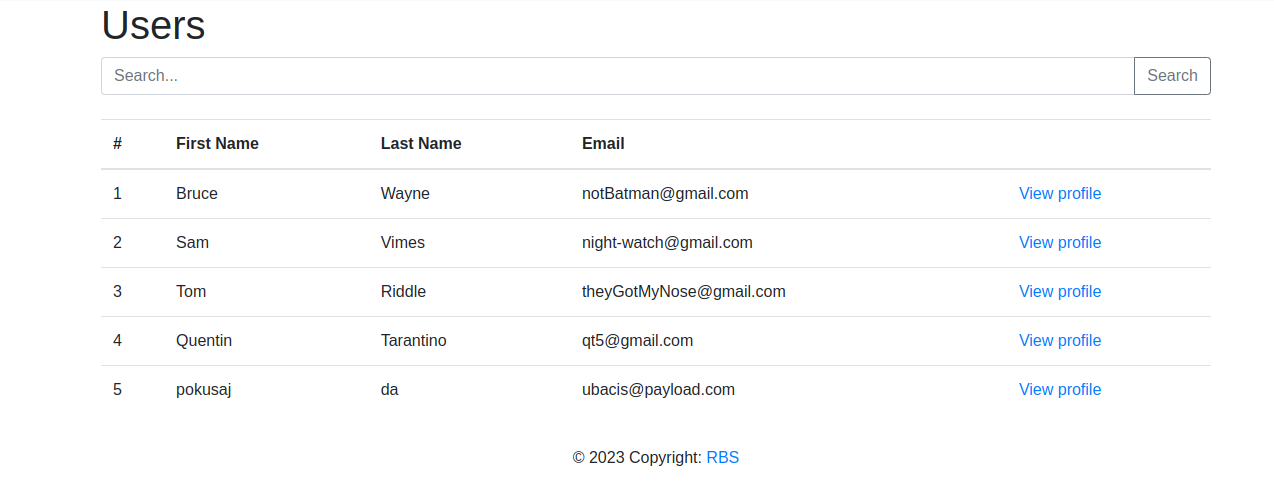
# SQL injection

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

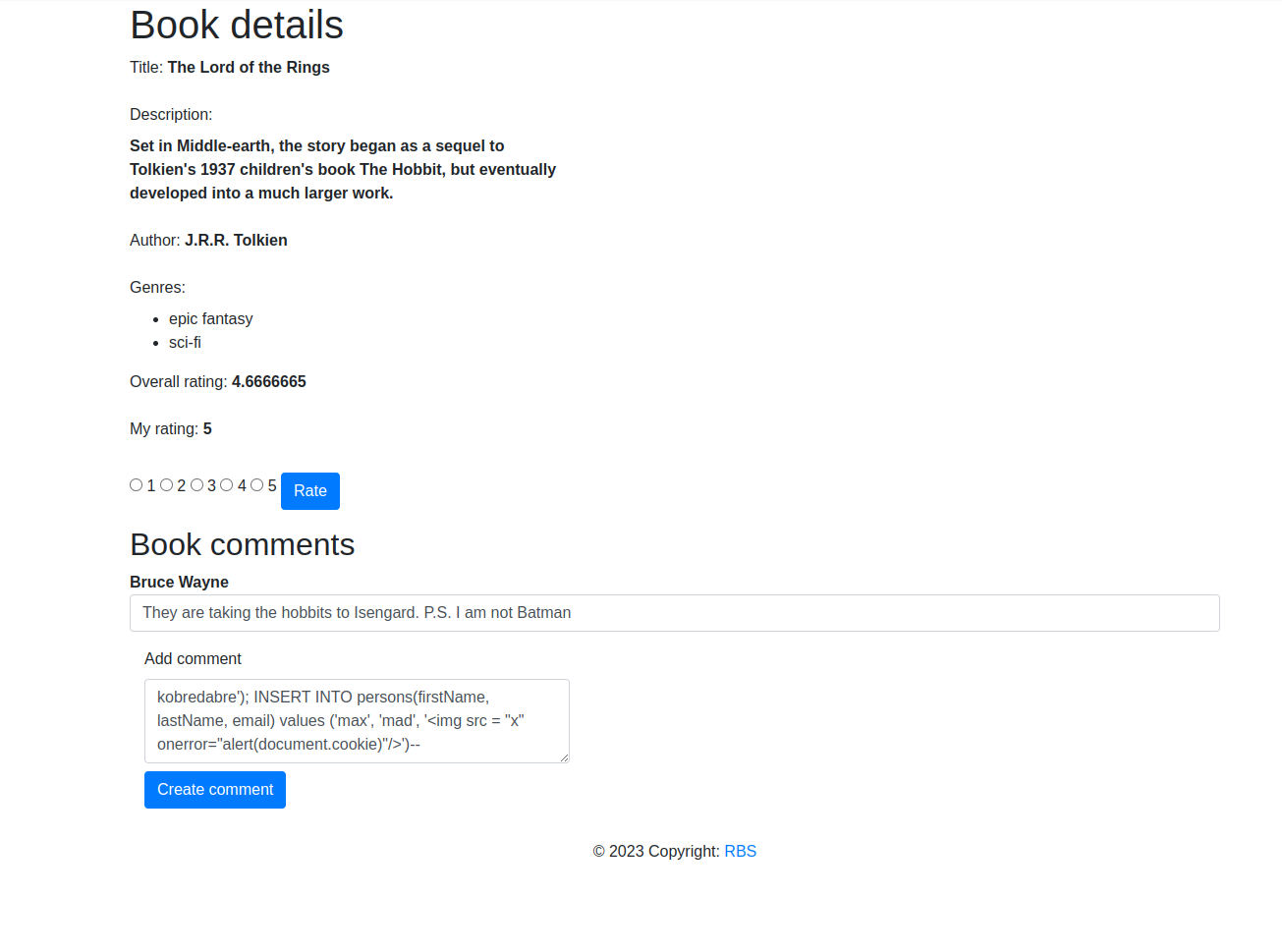
### Metod napada:

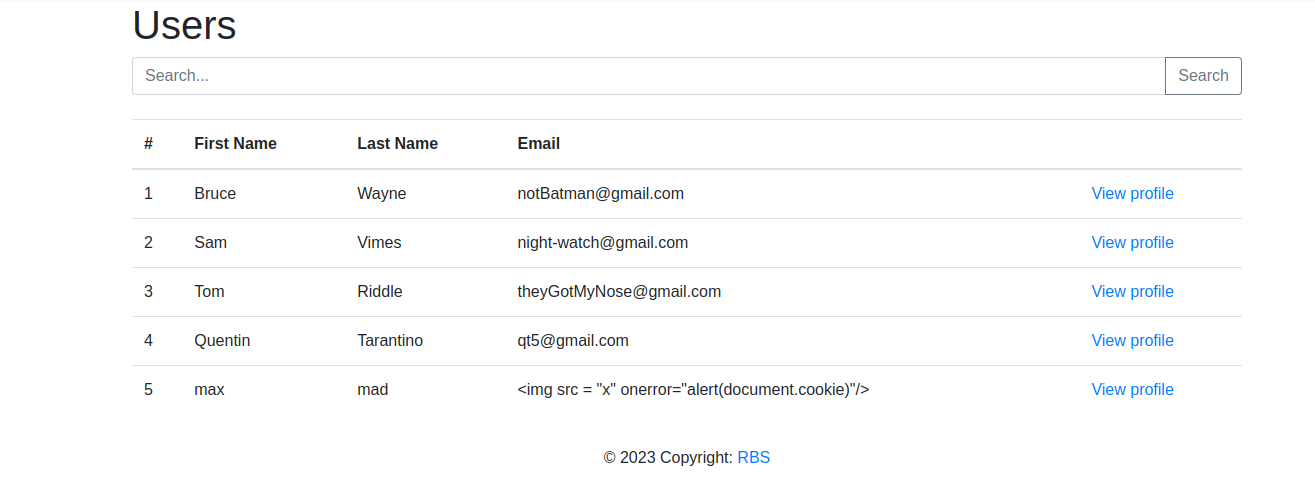
**1. SQL injection: Napad izvršavamo tako što unosom upita u polje **„Book comments“** pokušavamo da unesemo novog korisnika u našu listu **„Persons“**.**

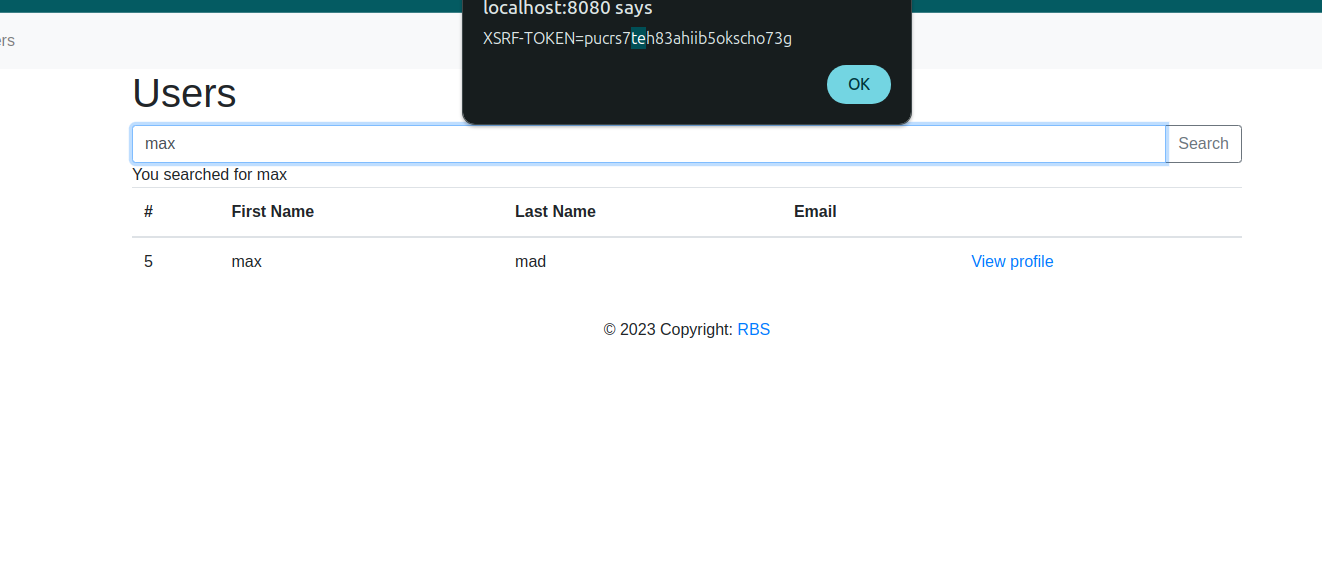
Slika 1: Unosimo maliciozni komentar

Slika 2: Korisnik sa datim podacima je unter u tabelu

**2. XSS: Izvodimo XSS napad tako što u polje „Book comments“ umesto email adrese unosimo JavaScript kod koji ima za cilj da prikaže kolačiće korisnika. Za razliku od prethodnog primera gde je unos bio bezopasan, ovaj put pokušavamo da izvršimo maliciozni skript kroz komentar.**

 Slika 3: Unosimo maliciozni komentar

Slika 4: Korisnik sa datim podacima je unet u tabelu

Slika 5: Pretraga korisnika

## Predlog odbrane:

**1. Zaštita od SQL injction-a** postiže se zamenom klase Statement sa klasom PreparedStatement. Na ovaj način sprečavamo izvršavanje zlonamernih SQL upita. Nakon ove izmene, komentar će i dalje biti uspešno dodat, ali nepoželjno kreiranje novog korisnika neće biti moguće.

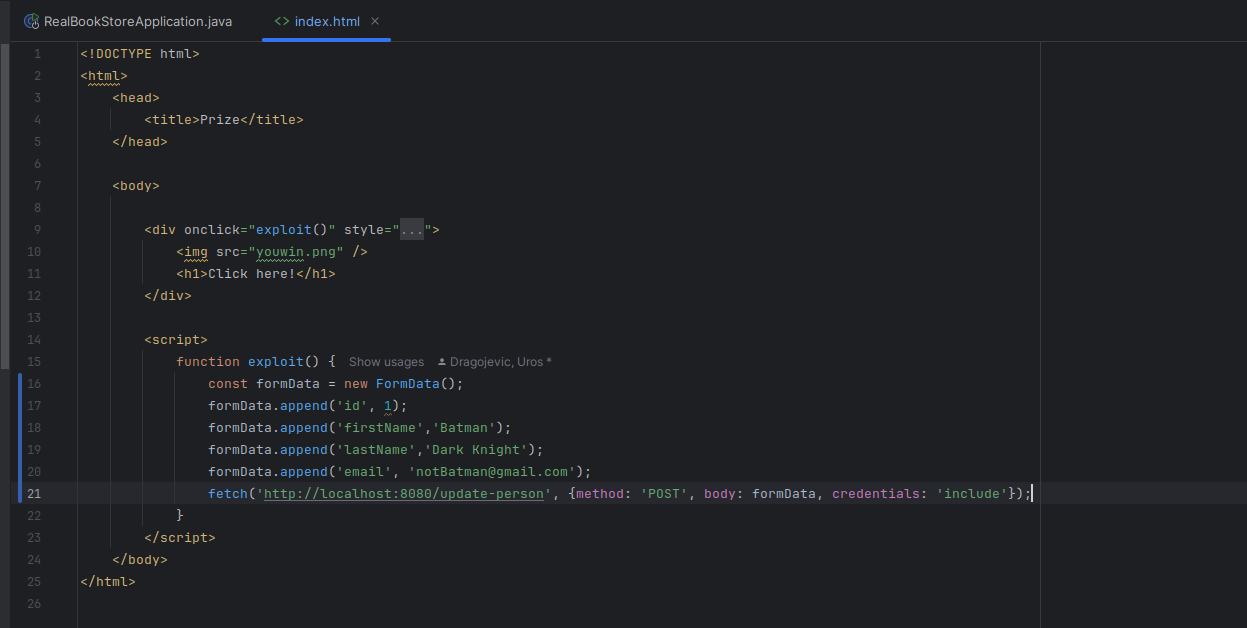
**2. Zaštita od XSS napada** obuhvata više koraka. Osim korišćenja PreparedStatement objekata, potrebno je u fajlu persons.html zameniti upotrebu innerHTML atributa sa textContent, kako bi se onemogućilo izvršavanje umetnutog JavaScript koda. Takođe, u fajlu book.html, potrebno je zameniti th:utext sa th:text, čime se sadržaj automatski escape-uje i prikazuje kao običan tekst. Nakon ovih izmena, aplikacija je zaštićena od XSS napada – komentar će biti prikazan, ali izvršavanje malicioznog koda i dodavanje novog korisnika neće biti moguće.

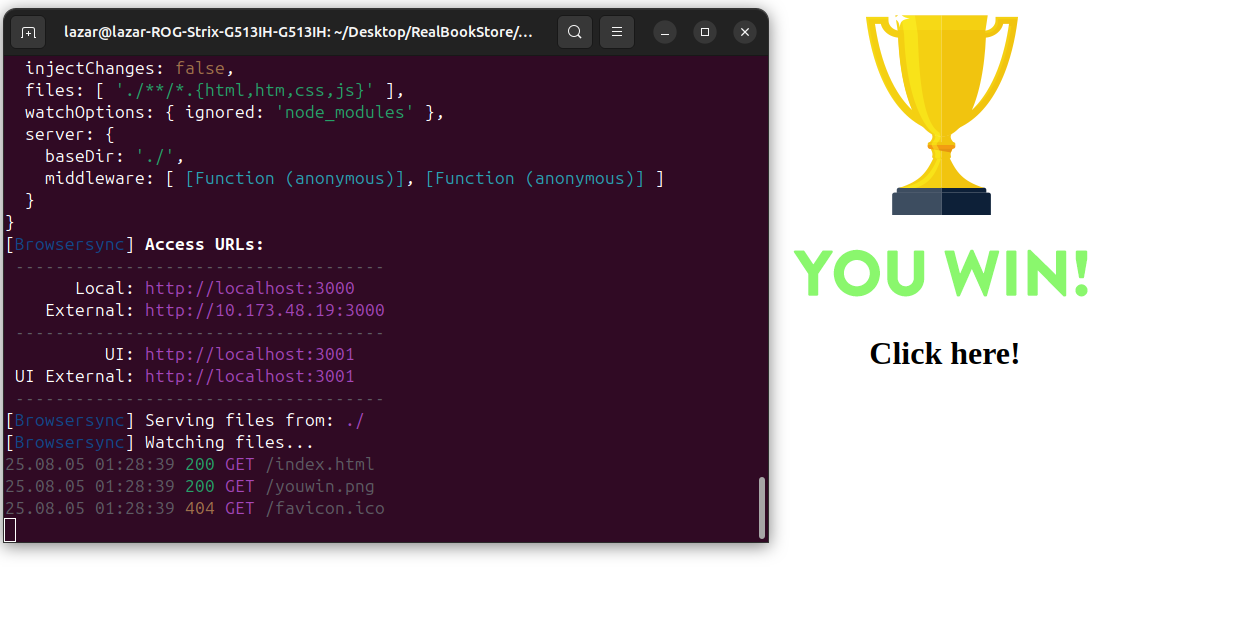
# Cross-site scripting

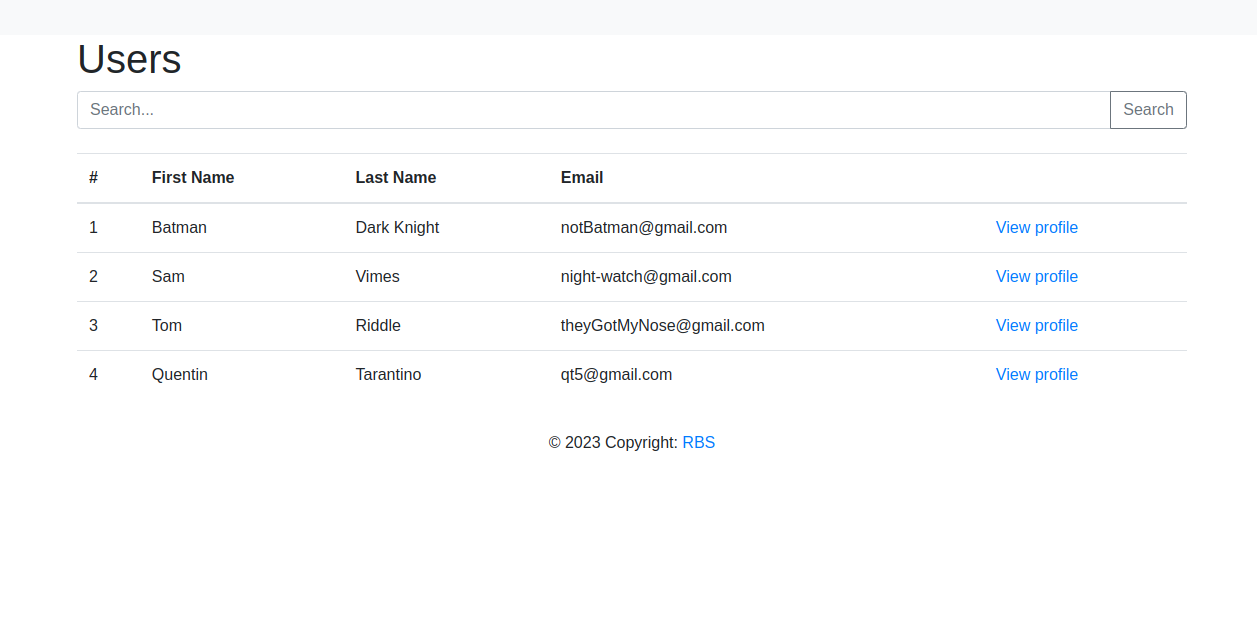
## Napad: Cross – site scripting

### Metod napada:

U direktorijumu csrf-exploit projekta nalazi se fajl index.html u koji je potrebno implementirati funkciju exploit. Ta funkcija treba da izmeni lične podatke korisnika sa id=1, tako da mu firstName postane „Batman“, a lastName „Dark Knight“.

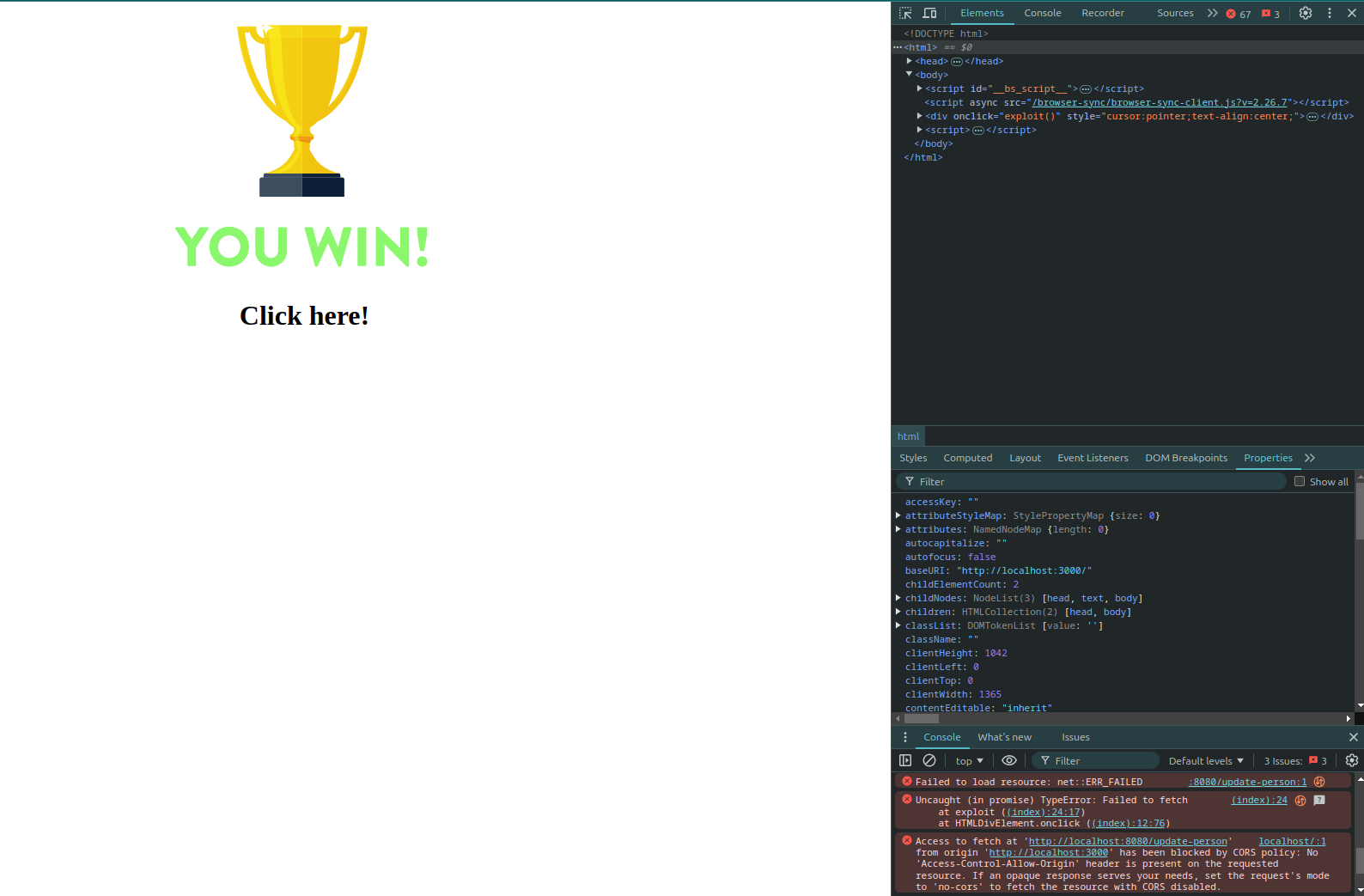
Slika 6: Skripta

Slika 7: Pokrecemu skriptu iz terminala. Na adresi http://localhost:3000/ je pokrenut sajt napadača. Klikom na pehar, pokreće se Cross-site request koji menja podatke.

Slika 8: Nakon napada vidimo promenjene podatke

## Predlog odbrane:

Za zaštitu u klasi PersonController, u metodi person učitavamo CSRF token iz sesije i prosleđujemo ga u model. Zatim, u metodi updatePerson preuzimamo token koji je poslat zajedno sa formom i upoređujemo ga sa tokenom sačuvanim u sesiji. Takođe, u formu za izmenu korisničkih podataka dodajemo poseban input element koji sadrži vrednost CSRF tokena.

Slika 9: Napad je odbijen

# Zaključak

Da bismo zaštitili web aplikaciju, koristimo različite tehnike koje sprečavaju napade. PreparedStatement nas štiti od SQL injekcija, dok menjanje načina prikaza sadržaja sprečava XSS napade. Takođe, CSRF zaštita sa tokenima osigurava da samo legitimni korisnici mogu menjati podatke. Sve ove mere zajedno čine aplikaciju sigurnijom i štite je od najčešćih napada.