Razvoj bezbednog softvera

Projekat 2023/2024

Sadržaj

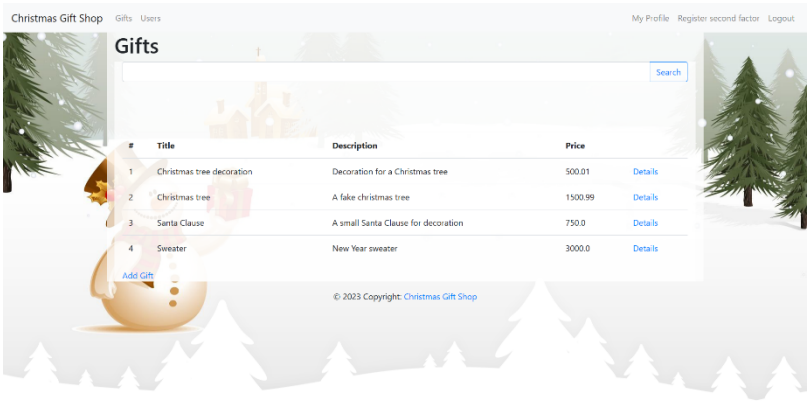
1. Uvod

Projekat se izvodi na aplikaciji *Christmas Gift Shop* koja pruža mogućnosti pretrage, ocenjivanja i kupovine poklona.

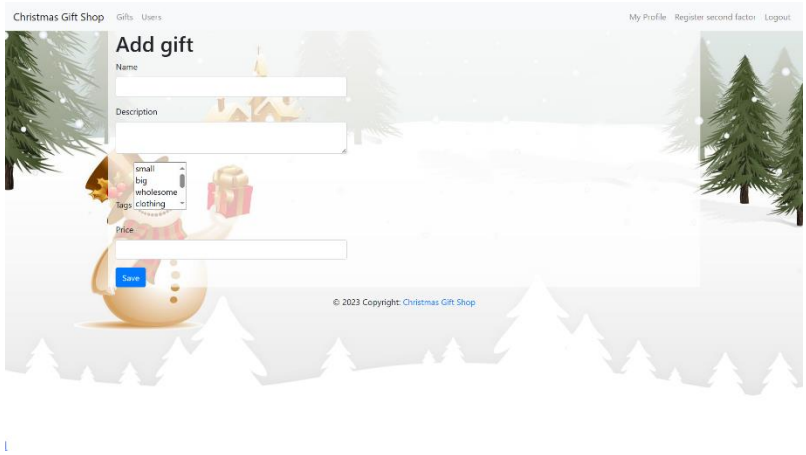
Projekat je potrebno skinuti sa sledećeg linka: https://github.com/danko-miladinovic/rbs2023-2024

Aplikacija *Christmas Gift Shop* omogućava sledeće:

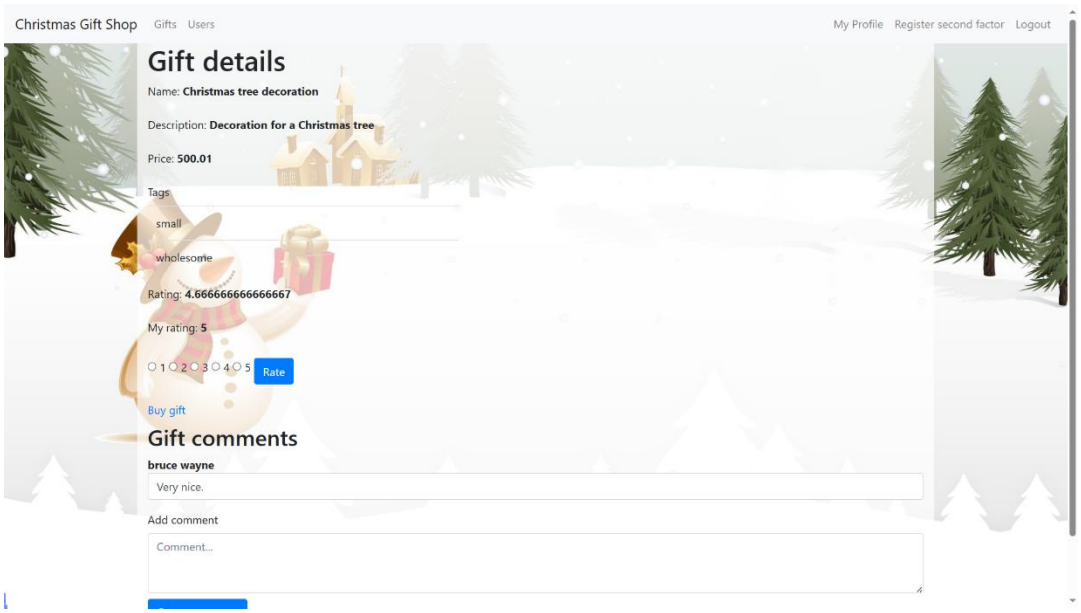
- Pregled i pretragu poklona (Slika 1.1).
- Dodavanje novog poklona (Slika 1.2).
- Detaljan pregleda poklona kao i komentarisanje i ocenjivanje poklona (Slika 1.3).
- Pregled korisnika aplikacije (Slika 1.4).
- Detaljan pregled podataka korisnika (Slika 1.5).



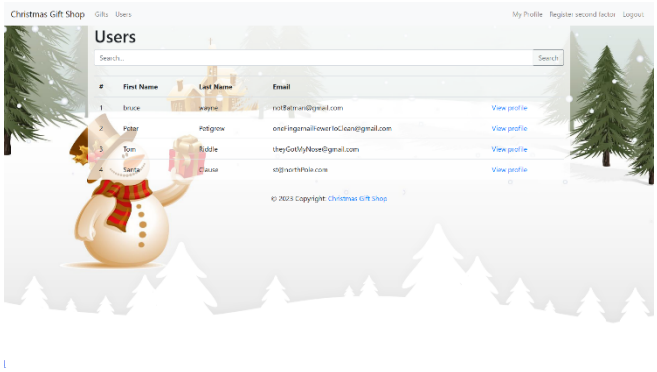
Slika 1.1 Pregled i pretragu poklona



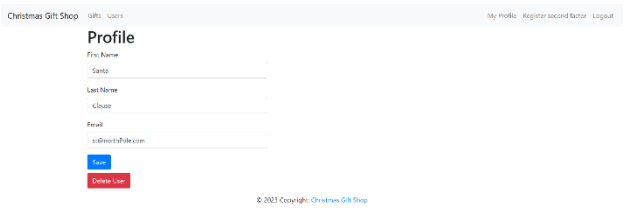
Slika 1.2 Dodavanje novog poklona



Slika 1.3 Detaljan pregleda poklona kao i komentarisanje i ocenjivanje poklona



Slika 1.4 Pregled korisnika aplikacije



Slika 1.5 Detaljan pregled podataka korisnika

U nastavku teksta su definisane stavke koje je potrebno uraditi u okviru projekta.

2. Primena alata za statičku analizu

Broj poena: 3

Pokrenuti alat za statičku analizu *SonarQube* i sastaviti izveštaj koji predajete sa projektom. Analizirati sve tačke (ukoliko postoje) u „*Vulnerability*“ i „*Security Hotspot*“ sekcijama alata. Za svaku stavku je potrebno:

- Obeležiti je kao *true positive* (*confirm*) ili *false positive* uz obrazloženje.
- Uraditi *screenshot* i priložiti uz rešenje projekta.

U nastavku se nalaze par **napomena** za statičku analizu:

1. U zavisnosti od verzije alata *SonarQube* koju koristite mogu da se razlikuju nazivi funkcionalnosti, način obrade nalaza i broj ranjivosti koje alat pronalazi.
2. Potrebno je da izveštaj bude pregledan.

Predlog: Preporuka je da se izveštaj preda u vidu Excel tabele sa screenshot-ovima (naravno, prihvataju se ostala rešenja). U svakom redu Excel tabele je potrebno da se navede jedan „*Vulnerability*“ ili „*Security Hotspot*“ i da se navede ime slike za naveden sigurnosni propust/ranjivost.

3. SQL injection i Cross-site scripting

Broj poena: 15

3.1. Napad

Na stranici na kojoj se pregleda pojedinačni poklon (Slika 1.3) nalazi se forma za ostavljanje komentara. Proveriti da li preko ove forme može da se izvrši neka vrsta *XSS* ili *SQLi* napada. Cilj napada je da se iskoristi kombinacija *XSS* i/ili *SQLi* napada za ubacivanje novog korisnika u bazu podataka i obezbedi mogućnost krađe kolačića sesije korisnika preko stranice *persons.html* (Slika 1.4). Voditi računa, novog korisnika je potrebno ubaciti u tabelu *person*, a ne u tabelu *user* u bazi. Stranica *persons.html* (Slika 1.4) je ranjiva na *XSS* napad, pa je konačni zadatak da ubačeni korisnik, kao neki od svojih atributa ima zlonamernu *JavaScript* skriptu koja će na konzolni izlaz da ispiše vrednost kolačića sesije korisnika.

Ranjivost se aktivira prilikom pretrage korisnika na stranici *persons.html* (Slika 1.4). Obavezno je dokumentovati ovaj napad i poslati ga sa projektnim zadatkom.

3.2. Odbrana

Neophodno je zaštititi se od prethodno opisanog napada tako da ne može da se izvrši napad preko forme za ostavljanje komentara. Odnosno, neophodno izvršiti zaštitu od *SQLi* napada i od *XSS* napada. Takođe, na stranici *persons.html* (Slika 1.4) neophodno je popraviti ranjivosti koje omogućuju *XSS* napad. Popravka ranjivosti ne sme da naruši ili promeni funkcionalnost aplikacije.

Napomena: Nije potrebno zaštititi celu aplikaciju od *SQLi* i *XSS* napada, već samo na mestima na kojima je uočen propust u ovoj sekciji (sekciji 3).

4. *Cross-site request forgery*

Broj poena: 7

4.1. Napad

Aplikacija nema zaštitu protiv *CSRF* napada. Neophodno je demonstrirati napad na sledeći način: koristeći aplikaciju koja simulira napad (folder *csrf-exploit* unutar projekta) treba napraviti skriptu tj. poziv ka *endpointu* **`/update-person`** unutar **`PersonsController.java`** klase. Napad treba da promeni lične podatke korisnika sa ID = 1, tako da je *fisrtname = "Dobby"* i *lastName = "Free Elf"*.

Obavezno dokumentovati napad i predati ga sa projektom.

4.2. Odbrana

Neophodno je implementirati zaštitu od *CSRF* napada korišćenjem tokena. Odbranu je potrebno primeniti samo na gorenavedeni endpoint. Popravka ranjivosti ne sme da naruši ili promeni funkcionalnost aplikacije.

5. Implementacija autorizacije

Broj poena: 10

Implementirajte matricu permisija kako je definisano u tabeli (Tabela 5.1) koristeći *Spring Security* i *Thymeleaf* koncepte koji su rađeni na vežbama.

Tabela 5.1 Tabela permisija

Permisija/rola	ADMIN	MANAGER	BUYER
ADD_COMMENT	*	*	*
VIEW_GIFT_LIST	*	*	*
CREATE_GIFT	*	*	
VIEW_PERSONS_LIST	*	*	
VIEW_PERSON	*		
UPDATE_PERSON	*	*-pogledaj napomenu	*-pogledaj napomenu
VIEW_MY_PROFILE	*	*	*
RATE_GIFT	*		*
BUY_GIFT	*	*	*

Postavite da korisnik **santa** ima rolu *ADMIN*, **tom** rolu *MANAGER* i korisnici **bruce** i **peter** imaju rolu *BUYER*. Napravite u bazi nedostajuću rolu i nedostajuće permisije.

U nastavku se nalaze kratki opisi permisija/rola:

ADD_COMMENT	–	Dozvoljava korisniku da doda komentar.
VIEW_GIFT_LIST	–	Dozvoljava korisniku da vidi i pretraži listu poklona.
CREATE_GIFT	–	Dozvoljava korisniku da unese nov poklon.
VIEW_PERSONS_LIST	–	Dozvoljava korisniku da vidi i pretraži listu korisnika.
VIEW_PERSON	–	Dozvoljava korisniku da vidi detalje osobe.
UPDATE_PERSON	–	Dozvoljava korisniku da promeni detalje osobe (isto važi za brisanje korisnika).
VIEW_MY_PROFILE	–	Dozvoljava pregled sopstvenog profila.
RATE_GIFT	–	Dozvoljava da korisnik oceni poklon.
BUY_GIFT	–	Dozvoljava da korisnik kupi poklon.

Napomena: Rolama *MANAGER* i *BUYER* je dozvoljena promena isključivo svojih sopstvenih podataka. Koristite pomoćne metode **`hasPermission`** i **`getCurrentUser`** iz klase **`SecurityUtil`** Ili metodu **`getPrincipal`** klase **`Authentication`** kako bi ste dohvatili trenutno ulogovanog korisnika.

6. DevOps

Broj poena: 5

6.1. Rukovanje izuzecima i logovanje

Uvesti obradu i logovanje svih izuzetaka u aplikaciji. Dodati logove koji bi bili korisni u analizi u slučaju napada. Ocenjivaće se:

- Izbor mesta u kodu gde je napravljen unos u log.
- Izbor log kategorije prema principima koji su predstavljeni na vežbama.
- Relevantnost opisa i podataka koji se nalaze u log poruci.

6.2. Auditing

Uvesti *auditing* aplikaciji. Ocenjivaće se:

- Implementacija *auditing*-a.
- Izbor korisničkih akcija za koje se vrši *audit* prema principima koji su predstavljenim na vežbama.
- Tačnost *audit*-a u pružanju sigurnosne usluge neporecivosti („*non-repudiation*“).

Napomena: Za lakšu implementaciju *auditing* dela zadatka mogu se koristiti metode iz klase *AuditLogger*. Pri obradi izuzetaka treba uzeti u obzir da li će se korisnikovo iskustvo poboljšati ukoliko mu se prikaže smislena poruka na korisničkom interfejsu. Ovaj deo se neće ocenjivati.

7. Priprema rešenja projekta

Projekat se predaje kao ZIP fajl se sledećom strukturom direktorijuma:

- Project
  - SonarQube izveštaj
  - Code
    - rbs2023-2024 [kod sa GitHub repozitorijuma sa implementiranim zaštitama]
  - Attacks
    - [*Microsoft Word* fajl/fajlovi sa koracima za napad ili *screenshot*-ovima sa uspešnim napadima i kodom korišćenim za napad]

8. Odbrana projekta

Odbrana projekta će se vršiti u tri ispitna roka:

- Januarski ispitni rok
- Februarski ispitni rok
- Julski ispitni rok

Svaki student će samostalno braniti svoj projekat.