

# Razvoj bezbednog softvera

Projekat 2021/2022

## Sadržaj

1.	Uvod .....	2
2.	Primena alata za statičku analizu .....	4
3.	SQL Injection.....	4
4.	Cross-site request forgery i Cross-site scripting.....	5
5.	Implementacija autorizacije .....	6
6.	DevOps .....	6
7.	Priprema rešenja projekta.....	7
8.	Odbrana projekta .....	7

## 1. Uvod

Projekat se izvodi na aplikaciji *Deliveries* koja pruža uslugu naručivanja i isporučivanja hrane.

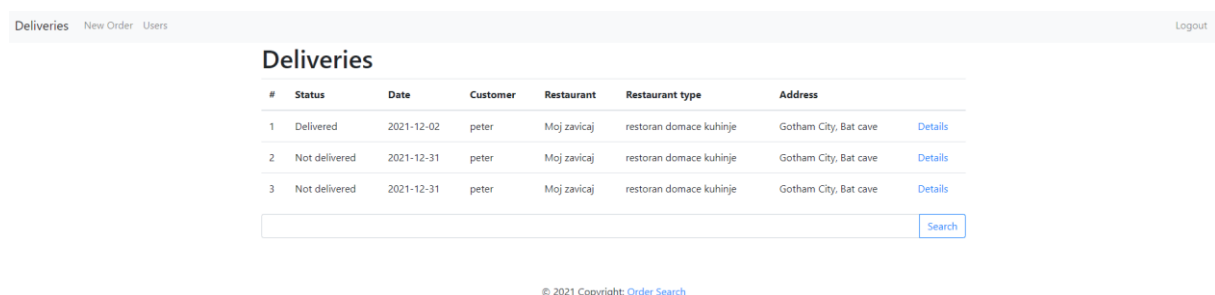
Projekat je potrebno skinuti sa sledećeg linka:

<https://github.com/5arV/SecureSoftwareDevelopmentProject2021>

Aplikacija *Deliveries* omogućava sledeće:

- Pregled i pretragu narudžbina (Slika 1.1 Narudžbine), kao i pregled svake pojedinačne narudžbine (Slika 1.2 Narudžbina).
- Naručivanje (Slika 1.3 Naručivanje) hrane biranjem jednog od postojećih restorana.
- Pregled korisnika aplikacije (Slika 1.4 Korisnici).

Preko stranice za pregled svih korisnika aplikacije (Slika 1.4 Korisnici) moguće je preći na stranice za izmenu podataka svakog korisnika (Slika 1.5 Pregled restorana i Slika 1.6 Pregled korisnika).

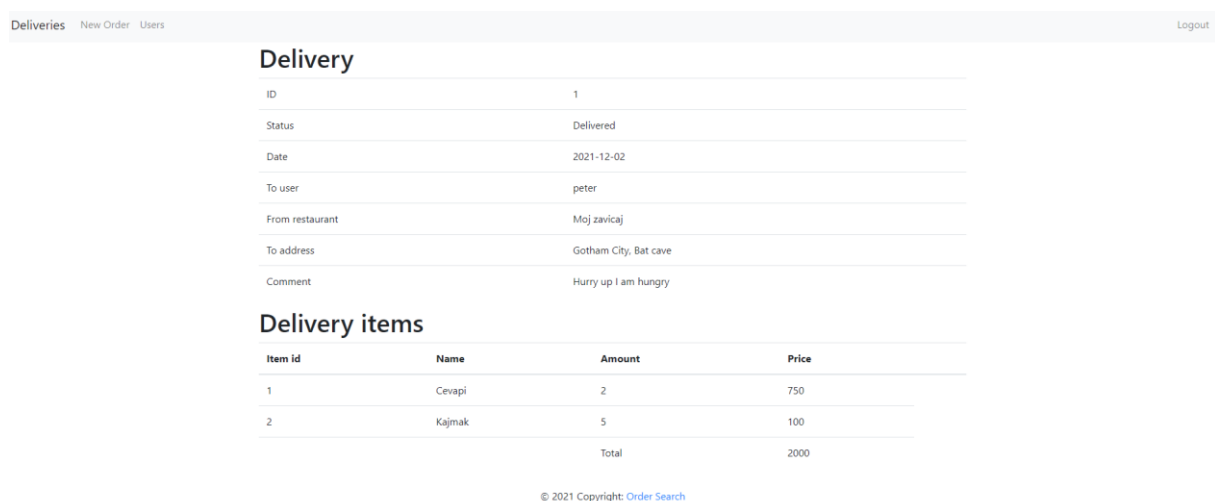


The screenshot shows the 'Deliveries' page of the application. At the top, there is a navigation bar with 'Deliveries', 'New Order', and 'Users' links, and a 'Logout' button on the right. The main heading is 'Deliveries'. Below it is a table with columns: #, Status, Date, Customer, Restaurant, Restaurant type, and Address. There are three rows of data. Each row has a 'Details' link. At the bottom of the table is a search bar with a 'Search' button. Below the table, there is a copyright notice: '© 2021 Copyright: Order Search'.

#	Status	Date	Customer	Restaurant	Restaurant type	Address	
1	Delivered	2021-12-02	peter	Moj zavicaj	restoran domace kuhinje	Gotham City, Bat cave	<a href="#">Details</a>
2	Not delivered	2021-12-31	peter	Moj zavicaj	restoran domace kuhinje	Gotham City, Bat cave	<a href="#">Details</a>
3	Not delivered	2021-12-31	peter	Moj zavicaj	restoran domace kuhinje	Gotham City, Bat cave	<a href="#">Details</a>

© 2021 Copyright: Order Search

Slika 1.1 Narudžbine



The screenshot shows the 'Delivery' page of the application. At the top, there is a navigation bar with 'Deliveries', 'New Order', and 'Users' links, and a 'Logout' button on the right. The main heading is 'Delivery'. Below it is a form with fields for ID, Status, Date, To user, From restaurant, To address, and Comment. Below the form is a section titled 'Delivery items' with a table showing the items in the delivery. At the bottom of the table is a 'Total' row. Below the table, there is a copyright notice: '© 2021 Copyright: Order Search'.

ID	1
Status	Delivered
Date	2021-12-02
To user	peter
From restaurant	Moj zavicaj
To address	Gotham City, Bat cave
Comment	Hurry up I am hungry

Item Id	Name	Amount	Price
1	Cevapi	2	750
2	Kajmak	5	100
Total			2000

© 2021 Copyright: Order Search

Slika 1.2 Narudžbina

Deliveries
New Order
Users
Logout

### Make a new order

Restaurant

Moj zavicaj

Dish	Amount
Cevapi	
Pijesakavica	
Kajmak	
Svadbarski kupus	
Becka snicla	

Address

Gotham City, Bat cave

Additional Remark

Submit

© 2021 Copyright: [Order Search](#)

Slika 1.3 Naručivanje

Deliveries
New Order
Users
Logout

### Restaurants

#	Name	Address	Type	Change
1	Delivered	Maksima Gorkog 12, Beograd	restoran domace kuhinje	<a href="#">Details</a>
2	Delivered	Obilicev venac 5, Beograd	pizza bar	<a href="#">Details</a>

### Customers

#	Username	Details
1	bruce	<a href="#">Details</a>
2	peter	<a href="#">Details</a>
3	tom	<a href="#">Details</a>

© 2021 Copyright: [Order Search](#)

Slika 1.4 Korisnici

Deliveries
New Order
Users

Name

Moj zavicaj

Address

Maksima Gorkog 12, Beograd

Type

Restoran domaće kuhinje

Save

Delete

© 2021 Copyright: [Order Search](#)

Slika 1.5 Pregled restorana

Deliveries New Order Users Logout

### User info

Username  
bruce

Password  
wayne

Save Delete

### Addresses

Gotham City, Bat cave	Save	Delete
Beograd, Gazela	Save	Delete
Beogradska industrija piva	Save	Delete

### Add new address

Save

© 2021 Copyright: Order Search

Slika 1.6 Pregled korisnika

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

## 2. Primena alata za statičku analizu

**Broj poena: 3**

Koristeći alat SonarQube (<https://www.sonarqube.org/>) pokrenuti statičku analizu i sastaviti izveštaj na osnovu dobijenih rezultata.

Od interesa za ovu analizu su stavke koje se vode kao **Security Hotspots**. Za svaku stavku potrebno je odrediti da li je u pitanju lažno pozitivna (**false positive**) stavka analize (i dati kratko objašnjenje zašto) ili potvrditi stavku ukoliko je ona istinski tačna (**true positive**). Izveštaj je slobodne forme, može biti *excel* tabela, *screenshot* ili bilo kakav drugi pregledan izveštaj.

**Napomena:** U zavisnosti od verzije alata *SonarQube* koju koristite mogu da se razlikuju nazivi funkcionalnosti, način obrade nalaza i broj ranjivosti koje alat pronalazi.

## 3. SQL Injection

**Broj poena: 10**

### 1. Napad

Izvesti SQL Injection napad na stranici za naručivanje hrane (Slika 1.3 Naručivanje) koji u bazu podataka ubacuje nov restoran (vrednosti za ime restorana, adresu restorana i tip restorana popuniti po sopstvenoj želji) i ubacuje samo jedno jelo za ubačeni restoran (vrednosti za naziv jela i cenu jela popuniti po sopstvenoj želji). Voditi računa o jedinstvenim identifikatorima prilikom sastavljanja upita.

### 2. Implementacija zaštite

Implementirati zaštitu na nivou repozitorijum sloja.

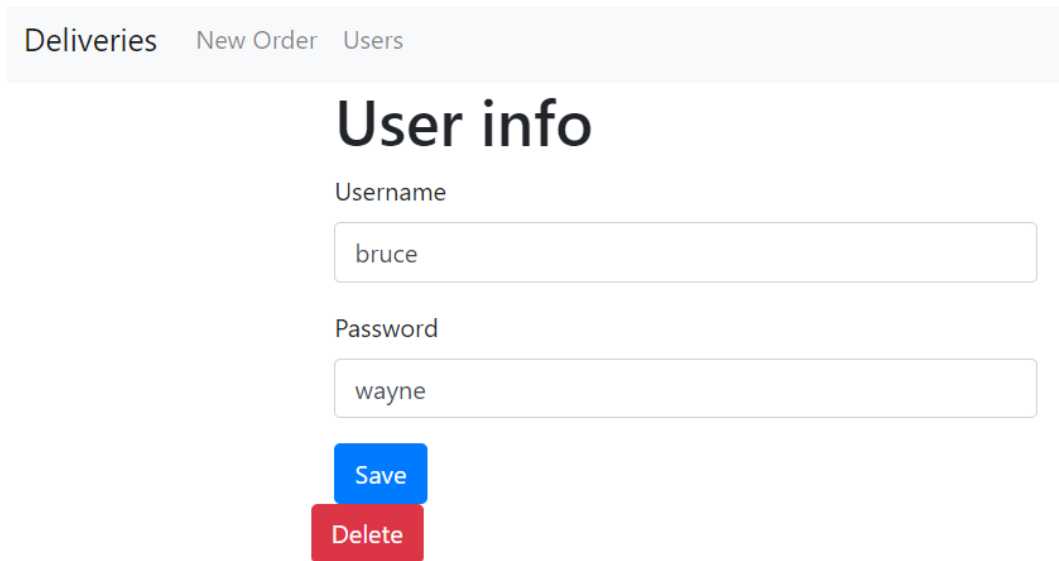
Zaštita ne sme da promeni funkcionalnost na korisničkom interfejsu.

## 4. Cross-site request forgery i Cross-site scripting

Broj poena: 12

### 1. Napad

Izvesti kombinaciju CSRF i XSS napada nad procesom promene korisničkog imena i lozinke. Proces započinje kada korisnik unese novi *username* i *password* i klikne na dugme *Save* na stranici *User Info* (Slika 4.1 *User info*).



Deliveries New Order Users

### User info

Username

Password

Save

Delete

Slika 4.1 User info

Efekat napada treba bude uspešan POST zahtev koji šalje payload sa novim korisničkim imenom i lozinkom (po Vašem izboru) za korisnika čiji je ID = 1.

Priložiti napad kao tekst ili kao *screenshot* uz rešenje projekta.

Za CSRF napad koristiti stranicu *csrf-exploit/index.html* koja je korišćena na vežbama.

### 2. Implementacija zaštite CSRF

Implementirati zaštitu od CSRF napada koristeći CSRF token. Zaštita ne sme da promeni funkcionalnost na korisničkom interfejsu.

### 3. Implementacija zaštite XSS

Implementirati zaštitu od XSS napada.

Zaštita ne sme da promeni funkcionalnost na korisničkom interfejsu.

## 5. Implementacija autorizacije

Broj poena: 10

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

Postavite da korisnik **peter** ima rolu *RESTAURANT* a **tom** rolu *ADMIN*. Napravite u bazi nedostajuće role i nedostajuće permisije.

Permisija	Rola		
	CUSTOMER	RESTAURANT	ADMIN
Naručivanje hrane (ORDER_FOOD)	✓		✓
Pregled liste korisnika (USERS_LIST_VIEW)		✓	✓
Pregled detalja o korisniku (USERS_DETAILS_VIEW)		✓	✓
Izmena detalja o korisniku (USERS_EDIT)			✓
Brisanje korisnika (USERS_DELETE)			✓
Pregled liste restorana (RESTAURANT_LIST_VIEW)	✓	✓	✓
Pregled detalja o restoranu (RESTAURANT_DETAILS_VIEW)		✓	✓
Izmena detalja o restoranu (RESTAURANT_EDIT)		✓	✓
Brisanje restorana (RESTAURANT_DELETE)			✓

Tabela 5.1 Tabela permisija

**Napomena:**

- Koristite pomoćne metode *hasPermission* i *getCurrentUser* iz klase *SecurityUtil*.
- Testirajte sve permisije za sve role.
- Ukoliko je potrebno, transformišite jednu stranicu sa više funkcionalnosti u više stranica sa po jednom funkcionalnošću.

## 6. DevOps

Broj poena: 5

### 1. Rukovanje izuzecima i logovanje

Uvesti obradu i logovanje svih izuzetaka u aplikaciji. Dodati logove koji bi bili korisni u analizi logova 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.

## 2. Auditing

Uvesti *auditing* u aplikaciju. Ocenjivaće se:

- Implementacija *auditing*-a.
- Izbor korisničkih akcija za koje se vrši *audit* prema principima koji su predstavljeni 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 smisljena poruka na korisničkom interfejsu. Ovaj deo se neće ocenjivati.

## 7. Priprema rešenja projekta

Projekat se predaje kao ZIP file sa sledećom strukturom direktorijuma:

- *Project*
  - *SonarQube* izveštaj
  - *Code*
    - *SecureSoftwareDevelopmentProject2021* [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.