

QuickHarvest e-shop

Slovenská technická univerzita, Fakulta elektrotechniky a informatiky

Bc. Simona Zlatohlávková, Bc. Adam Kučmín, Bc. Dagmar Trbalíková, Bc. Martin Nachtmann, Bc. Miroslav Sáráz

1. Výber technológií

Backend

1.1. Java 17

Java je viacúčelový, objektovo orientovaný programovací jazyk ktorý je rýchly, bezpečný a spoľahlivý (1). V našom projekte využívame jazyk Java pre implementáciu biznis logiky a manipuláciu s dátami. Rozhodli sme sa preň z dôvodu predchádzajúcich skúseností s týmto jazykom.

1.2. Spring Boot framework 3.1.4

Spring Boot je framework pre vývoj mikroslužieb a webových aplikácií v jazyku Java. Poskytuje jednoduchý spôsob na vytvorenie samostatnej, produkčne pripravenej aplikácie. V našom projekte sme sa rozhodli pre Spring Boot kvôli našim skúsenostiam s jeho používaním a rýchlosti vývoja (2).

1.3 JPA Repozitár 3.1.4

V repozitári, ktorý interaguje s databázou rozširujeme rozhranie JPA Repository, ktoré nám umožňuje používať množstvo preddefinovaných metód. V zadaní sme zatiaľ využili možnosť získania záznamu z databázy na základe primárneho kľúča.

1.4 PostgreSQL 42.6.0

PostgreSQL je open-source relačný databázový systém, ktorý sme zvolili pre náš projekt na ukladanie a manipuláciu s dátami. Táto voľba vychádza z jeho spoľahlivosti a výkonu. Využívame ho ako backendovú databázu pre náš JPA repozitár, čo nám umožňuje efektívne spravovať dátové operácie v našich mikroslužbách implementovaných pomocou Spring Boot frameworku (3).

Frontend

1.5 Javascript

JavaScript, je skriptovací programovací jazyk. Jazyk je používaný najmä pri tvorbe webových stránok(5). Rozhodli sme sa preň z dôvodu predchádzajúcich skúseností s týmto jazykom.

1.6 React.js

React je open-source frontendová knižnica JavaScriptu na vytváranie používateľských rozhraní alebo komponentov používateľského rozhrania(6). React.js sme si vybrali kvôli jeho popularite, jednoduchosti a z dôvodu predchádzajúcich skúseností s vývojom webovej aplikácie pomocou tohto nástroja.

2. Implementácia

V aplikácii je implementovaná registrácia, prihlásenie aj prenos šifrovaných údajov a následné dešifrovanie. Po spustení kontajnera v backendovej časti aplikácie pomocou príkazu **docker-compose up --build** a následnom spustení frontendu je možné tieto funkcionality pretestovať, ukážky funkcionality sú aj na screenshotoch v ďalšej časti dokumentácie.

2.1 Registrácia

Kontrola zložitosti hesla

Kontrola zložitosti hesla prebieha vo frontendovej aj backendovej časti aplikácie pomocou regexov a kontroly dĺžky hesla. V našej aplikácii sú minimálne požiadavky na dĺžka hesla 8 znakov a heslo musí obsahovať minimálne 1 malé písmeno, 1 veľké písmeno, jednu číslicu a jeden špeciálny znak zo zoznamu: @#\$%^&+=. Pokiaľ heslo nespĺňa požiadavky, používateľ je na to upozornený a registrácia nie je úspešná.

Heslo je v frontendovej časti kontrolované pomocou yup validačnej schémy.

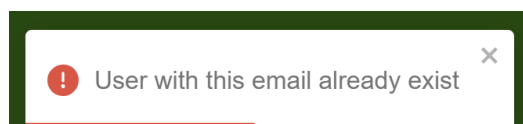
Kontrola platnosti emailu

Pri pokuse o registráciu prebehne vo frontendovej aj backendovej časti aplikácie kontrola, či má zadaný email platný tvar. Kontrola sa vykonáva pomocou regexov a očakáva sa, že email je v tvare name@domainsite.domain. Pokiaľ nie je email v platnom tvare, používateľ je na to upozornený a registrácia nie je úspešná. Email je v frontendovej časti kontrolovaný pomocou yup validačnej schémy.

Obrázok 1 Upozornenie na uniknuté heslo

Kontrola existujúceho používateľa

V backendovej časti aplikácie prebehne pred registrovaním kontrola, či sa v databáze nenachádza používateľ s rovnakým emailom. Pokiaľ je už email použitý, registrácia je neúspešná.



Obrázok 2 Upozornenie na existujúceho používateľa

Ukladanie hesla

Na zabezpečenie bezpečného uloženia hesla je použitý **BCryptPasswordEncoder**, pridaný do Spring Security. Bcrypt používa silný hashovací algoritmus spolu s pridaním „soli“ pre zvýšenie bezpečnosti hesiel. Tento algoritmus vytvára hash pomocou iteratívneho procesu založeného na Blowfish kryptografickom algoritme (4). Heslo je teda pred uložením do databázy bezpečne zahashované a je k nemu pridaný aj náhodne generovaný reťazec, teda „sol“.

JWT

Po úspešnom prihlásení do aplikácie sa v backendovej časti vygeneruje JWT (JSON Web Token). Z bezpečnostných dôvodov je jeho platnosť nastavená na 30 minút, teda aj platnosť prihlásenia používateľa je iba 30 minút. V JWT je bezpečne uložené ID prihláseného používateľa, čo umožňuje bezpečnú autorizáciu pri ďalších dopytoch. Pri dopyte na endpoint, pre ktorý je nutná autorizácia s neplatným alebo vymysleným JWT je takáto aktivita v backendovej časti odchytená a používateľ je na to upozornený a jeho prístup nie je povolený. Vo frontendovej časti je tento token uložený do Cookies. Tento token sa posielá spolu s dopytmi na backendovú časť aplikácie v hlavičke dopytu. Po uplynutí 30 minút je z cookies odstránený a používateľ je presmerovaný na stránku prihlásenia.

Name	Value	D...	Path	Expires / Max-Age
auth	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZWxzXWw...	lo...	/	2023-11-18T10:04:25.000Z

Obrázok 3 Uloženie tokenu

Validácie

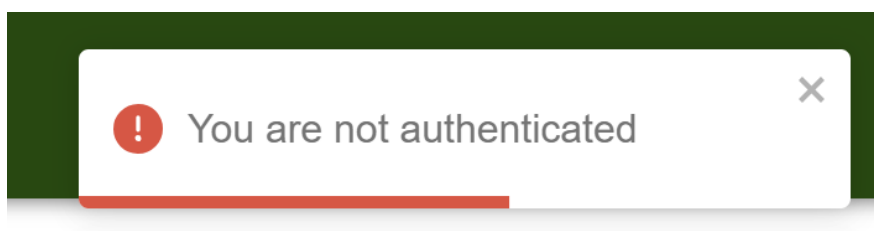
Pri prihlásení ja taktiež implementovaná validácia na vstupoch od používateľa. Taktiež je v backendovej časti implementovaná validácia na kontrolu správnosti emailu a hesla. V prípade, že jedno s týchto polí nie je správne, používateľ je upozornený.

Po úspešnom prihlásení je používateľovi prispôsobený navigačný panel, v ktorom sa následne.

2.5 Autorizácia dopytov

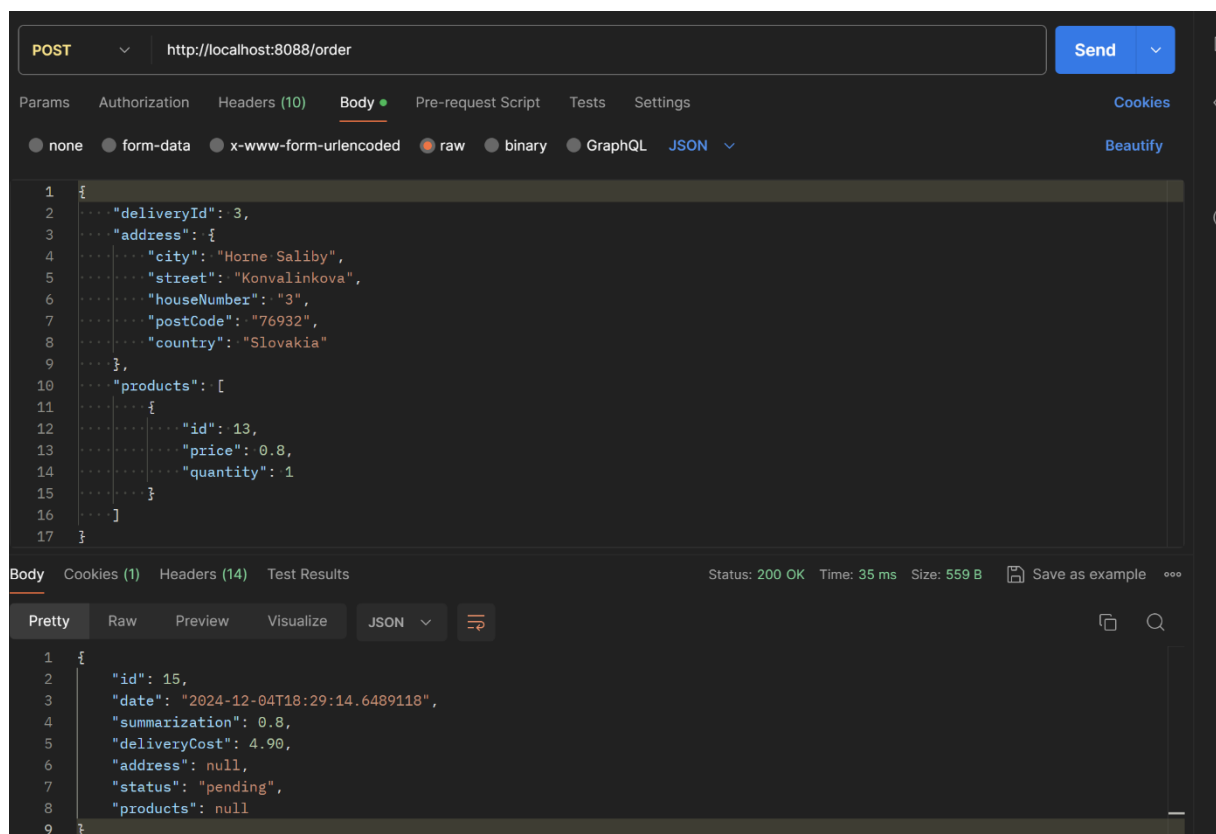
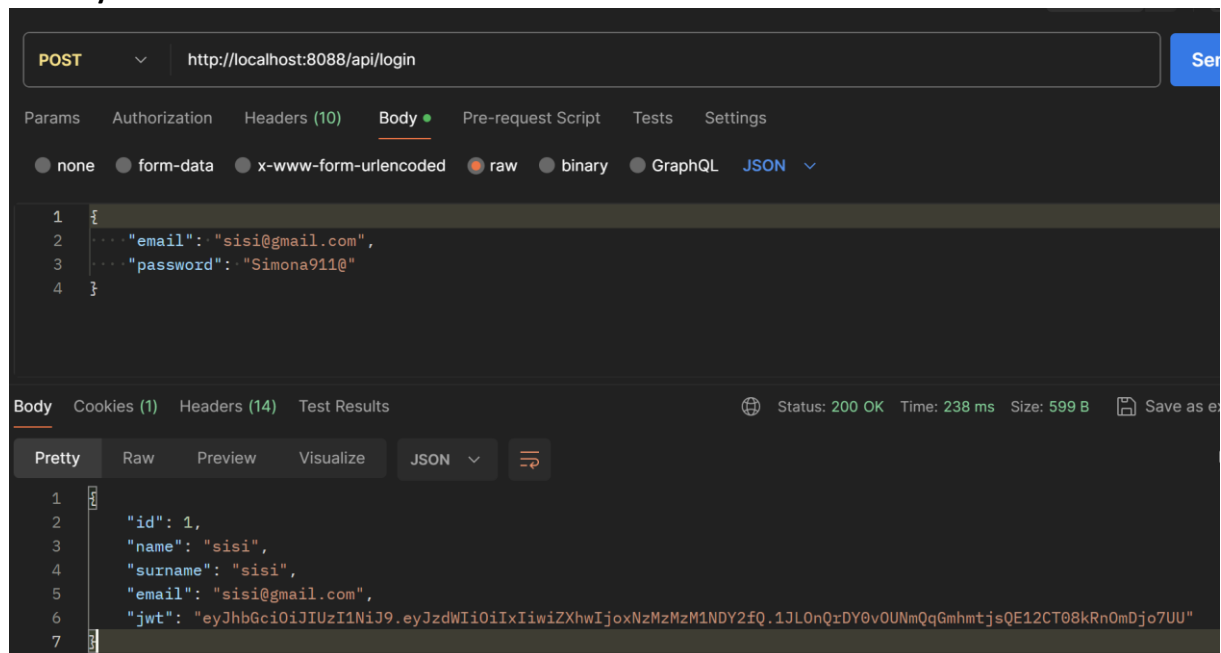
Každý dopyt vykonaný po prihlásení používateľa je autorizovaný pomocou autorizačnej hlavičky zasielanej s dopytom. Táto autorizačná hlavička obsahuje JWT token, ktorý je uložený v Cookies aplikácie. V prípade modifikácie alebo odstránenia tohto tokenu, je používateľovi odmietnutý prístup a je následne presmerovaný na stránku s prihlásením. Tento token má expiráciu 30 minút. Po tomto čase je používateľ presmerovaný na stránku s prihlásením.

V prípade že užívateľ sa bez prihlásenia snaží dostať na link ktorý vyžaduje autentifikáciu, je upozornený a presmerovaný na prihlásenie.



3.1 Backend

Príklady:



Databázový model




Swagger


V projekte sa nachádza .yaml súbor s popisom endpointov (swagger)


GET	/user/orders	Users orders	🔒	▼
GET	/user/profile	User info	🔒	▼
GET	/delivery	delivery info	🔒	▼
POST	/order	Order from user	🔒	▼
POST	/product/filter	Product filter	🔒	▼


Registrácia používateľa


**QUICK
HARVEST**












Prihlásenie používateľa



**QUICK
HARVEST**




sisi@gmail.com

Vyhľadanie produktov

Úspešne prihlásený užívateľ je automaticky presmerovaný na stránku s vyhľadani.


QUICK HARVEST

SEARCH




Milk Super
Full cream milk
Available 15 pc

- 1 +




0.80 €




Milk Rajo
Slovakian milk
Available 12 pc

- 1 +




0.90 €




Milk Coconut
Rich coconut milk
Available 10 pc

- 1 +




1-20 €

1.10 €





Milk Cacao
Chocolate milk drink
Available 8 pc

- 1 +



1.00 €

Košík

QUICK HARVEST

1

Cart

2


Address

3

Delivery

4

Order





Milk Super

0.80 € / pc

- 3 +

2.40 €






Milk Rajo

0.90 € / pc

- 1 +

0.90 €



Total: 3.30 €

TO ADDRESS >



Cart



Address



Delivery



Order

Please enter your shipping address.

Street

Hlavná

House Number

12/8

City

Bratislava

Postal Code

87654

Country

Slovakia

< BACK TO CART

CONFIRM ADDRESS >



Cart



Address



Delivery



Order



DPD delivery

1.2 €

Delivery tomorrow



SPS delivery

3.8 €

Delivery tomorrow



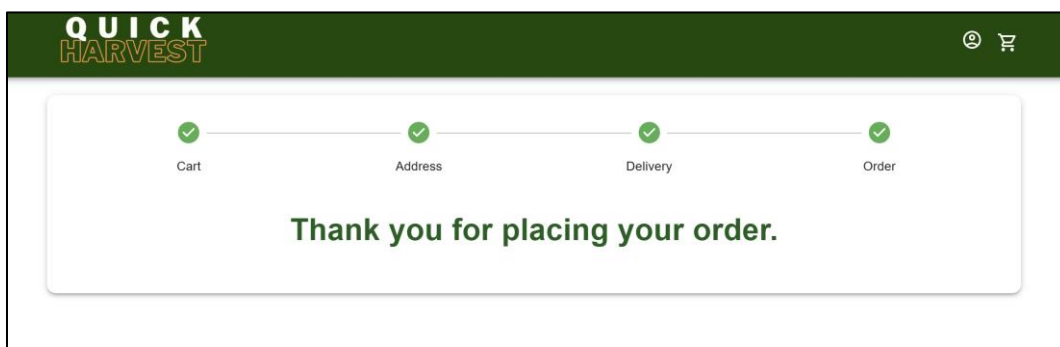
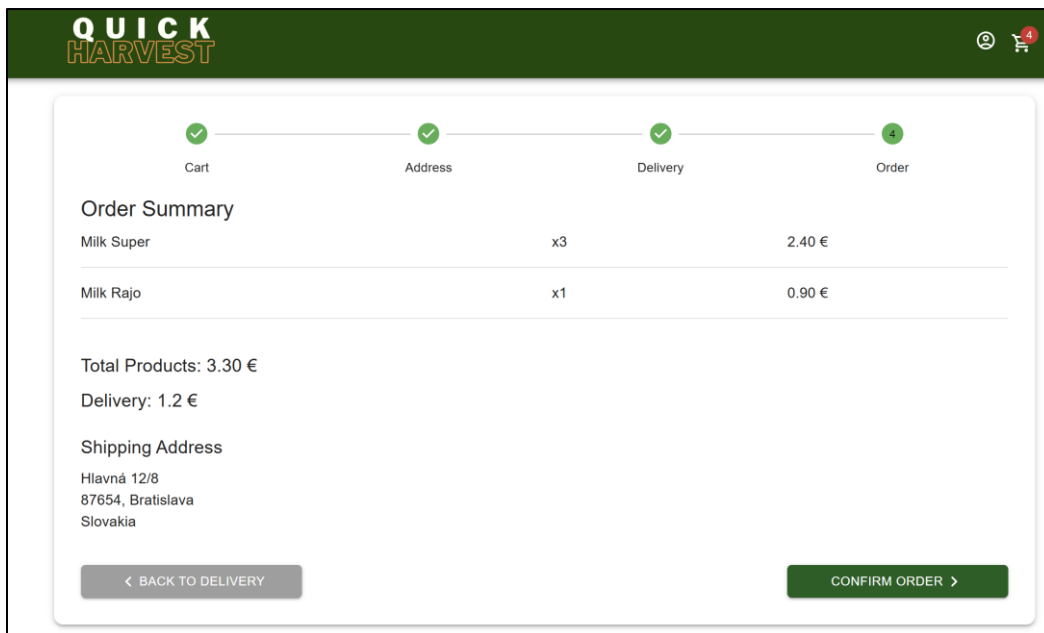
DHL delivery

4.9 €

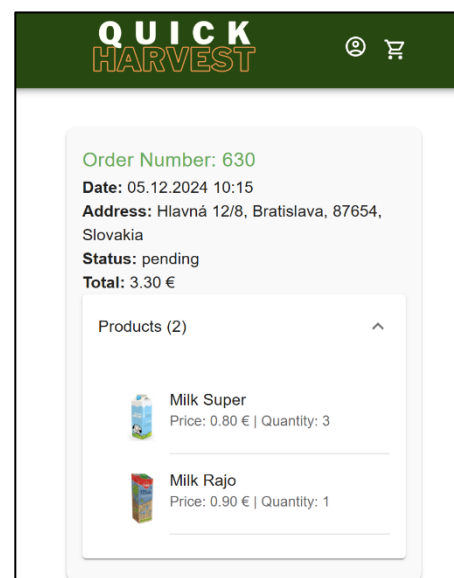
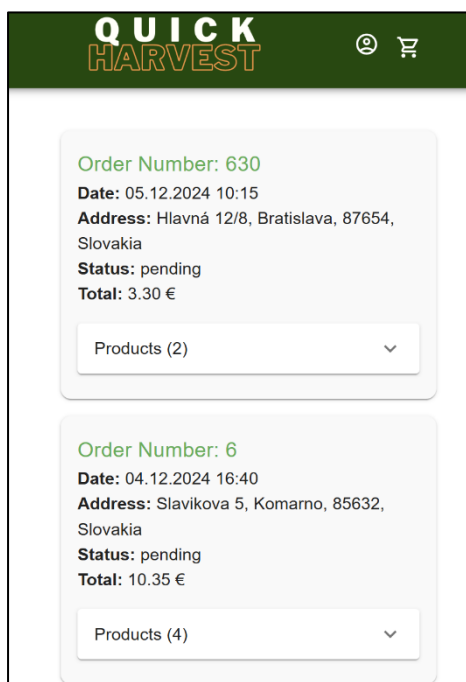
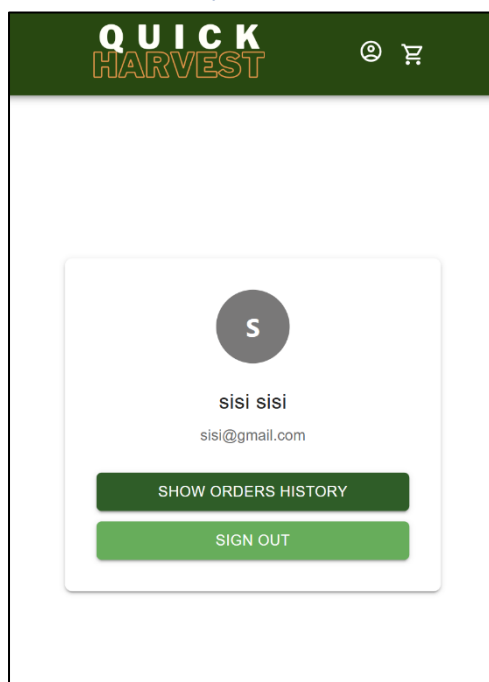
Delivery tomorrow

< BACK TO ADDRESS

ORDER SUMMARY >



Profil používateľa

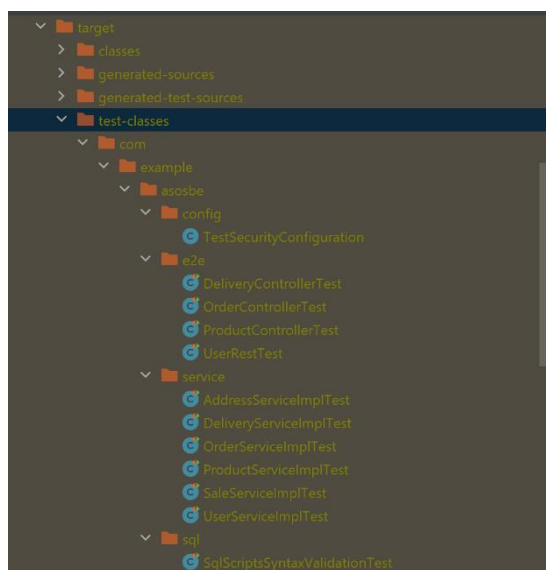


Odhlásenie používateľa

Používateľovi je taktiež umožnené odhlásenie pomocou tlačidla „Sign Out“ v profile . Ak klikne na toto tlačidlo, je automaticky presmerovaný na stránku s prihlásením a jeho token je odstránený.

Testovanie

V projekte sa nachádza adresár s testami. Taktiež je v PDF priložený report z Load testovania našej aplikácie.



Test setup

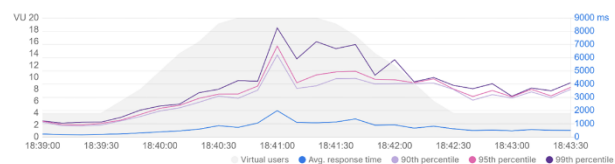
Virtual users	Start time	Load profile
20 VU	Dec 4, 18:38:33 (GMT+1)	Peak
Duration	End time	Environment
5 minutes	Dec 4, 18:43:39 (GMT+1)	-

1. Summary

Total requests sent	Throughput	Average response time	Error rate
3,279	10.72 requests/second	718 ms	0.00 %

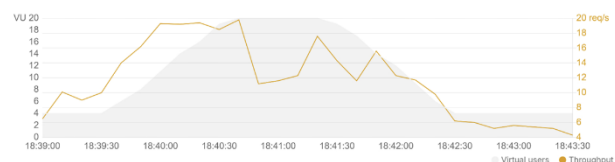
1.1 Response time

Response time trends during the test duration.



1.2 Throughput

Rate of requests sent per second during the test duration.



Zdroje:

- (1) https://www.w3schools.com/java/java_intro.asp
- (2) <https://www.ibm.com/topics/java-spring-boot>
- (3) <https://aws.amazon.com/rds/postgresql/what-is-postgresql/>
- (4) <https://www.baeldung.com/spring-security-registration-password-encoding-bcrypt>
- (5) <https://sk.wikipedia.org/wiki/JavaScript>
- (6) [https://sk.wikipedia.org/wiki/React_\(webov%C3%BD_framework\)](https://sk.wikipedia.org/wiki/React_(webov%C3%BD_framework))

Ďalšie zdroje použité pri implementácii:

- <https://www.baeldung.com/java-rsa>
- <https://www.devglan.com/java8/rsa-encryption-decryption-java>
- <https://docs.spring.io/spring-security/site/docs/current/api/org.springframework.security.crypto.bcrypt/BCryptPasswordEncoder.html>
- <https://www.baeldung.com/java-email-validation-regex>
- <https://www.geeksforgeeks.org/how-to-validate-a-password-using-regular-expressions-in-java/>
- <https://formik.org/docs/guides/validation>
- <https://mui.com/material-ui/getting-started/>
- <https://www.tabnine.com/code/javascript/functions/node-forge/decode64>
- <https://snyk.io/advisor/npm-package/node-forge/functions/node-forge.pki.privateKeyFromPem>
- <https://fkhadra.github.io/react-toastify/introduction>
-