UPB - Zadanie 4

Na obsiahnutie funkcionalít zadania 3 sme vytvorili dve separátne aplikácie: backend a frontend. V tomto zadaní 4 sme existujúce aplikácie rozšírili o ďalšie funkcionality. Backend bol implementovaný v programovacom jazyku Python, konkrétne v Django Frameworku. Aplikácia frontendu bola implementovaná pomocou webového frameworku React. Backend a frontend aplikácie medzi sebou komunikujú prostredníctvom API rozhrania. Funkcionality pridané v zadaní 4 do oboch aplikácii, ich implementácia a spustenie sú opísané v nasledujúcich kapitolách.

Spustenie

Spustenie pomocou Docker-u

Na spustenie aplikácií v docker kontajneroch je potrebné mať nainštalovaný Docker a Docker Compose.

V termináli (v priečinku UPB-eshop) spustíme nasledujúci príkaz:

```
docker-compose up -d -build
```

Po spustení tohto príkazu sú vytvorené a spustené tri kontajnery: ‘be‘ (backend), ’fe‘ (frontend) a ‘db‘ (PostgreSQL databáza). Django development server je dostupný na http://localhost:8000/admin/ s prihlasovacími údajmi admin@admin.com:admin. API endpointy poskytované backendom sú dostupné na http://localhost:8000/api/ na jednotlivých URI. Frontend aplikácia je dostupná na http://localhost:3000/.

Lokálne spustenie

Postup spustenia jednotlivých aplikácii (backend a frontend) na svojom lokálnom zariadení (mimo dockeru) je opísaný v súbore README.md v priloženom zdrojovom kóde v priečinku UPB-eshop.

Backend dokumentácia

Vyhľadávacie pole

Vyhľadávanie produktov na eshope sme riešili na backendovej strane full-textovým vyhľadávaním v názve produktu. Teda každý produkt, ktorého názov obsahuje vyhľadávaný substring je vrátený v odpovedi na frontend. O túto funkcionalitu sme rozšírili už existujúci endpoint na vrátenie zoznamu produktov. Ak požiadavka z frontendu obsahuje substring na

vyhľadávanie, vrátia sa len produkty, ktoré obsahujú substring, inak sú vrátené všetky produkty.

Ochrana pred SQL-injection

Django poskytuje vstavanú ochranu proti SQL injection útokom prostredníctvom použitia parametrizovaných queries a systému Object Relational Mapping (ORM).

Parametrizované queries: Databázové API Djanga používa parametrizované queries, čo znamená, že vstupy sa považujú za parametre a nie sú použité priamo ako SQL query. To pomáha zabrániť SQL injection útokom oddelením SQL kódu od vstupu používateľa.

Ochrana ORM: Systém Django ORM umožňuje interagovať s databázou pomocou Python objektov namiesto nespracovaných SQL dotazov. ORM automaticky validuje a čistí používateľské vstupy pri generovaní SQL dotazov.

Automatické čistenie vstupov: Django automaticky maže znaky v používateľských vstupoch, ktoré majú v SQL špeciálny význam, ako napríklad úvodzovky. To zaisťuje, že aj keď používateľský vstup obsahuje škodlivý SQL kód, bude sa s ním zaobchádzať ako s údajmi a nie ako spustiteľným kódom.

Vytvorenie objednávky, pridanie recenzie

Medzi funkcionalitami backendu pridanými v tomto zadaní je aj vytvorenie objednávky a pridávanie recenzie k jednotlivým produktom e-shopu. Tieto funkcionality zahŕňajú prenos citlivých údajov medzi frontendom a backendom cez Rest API. Citlivé údaje sa prenášajú napríklad pri požiadavke na vytvorenie objednávky s údajmi o objednávke v tele požiadavky poslanými z frontendu na backend, alebo pri odpovedi na túto požiadavku s údajmi faktúry zaslanými z backendu späť na frontend, alebo pri požiadavke na vytvorenie recenzie k produktu. Tieto údaje sú pred prenosom či už z frontendu alebo z backendu zašifrované, kvôli bezpečnosti, tak ako to bolo aj v predchádzajúcich zadaniach pri iných funkcionalitách.

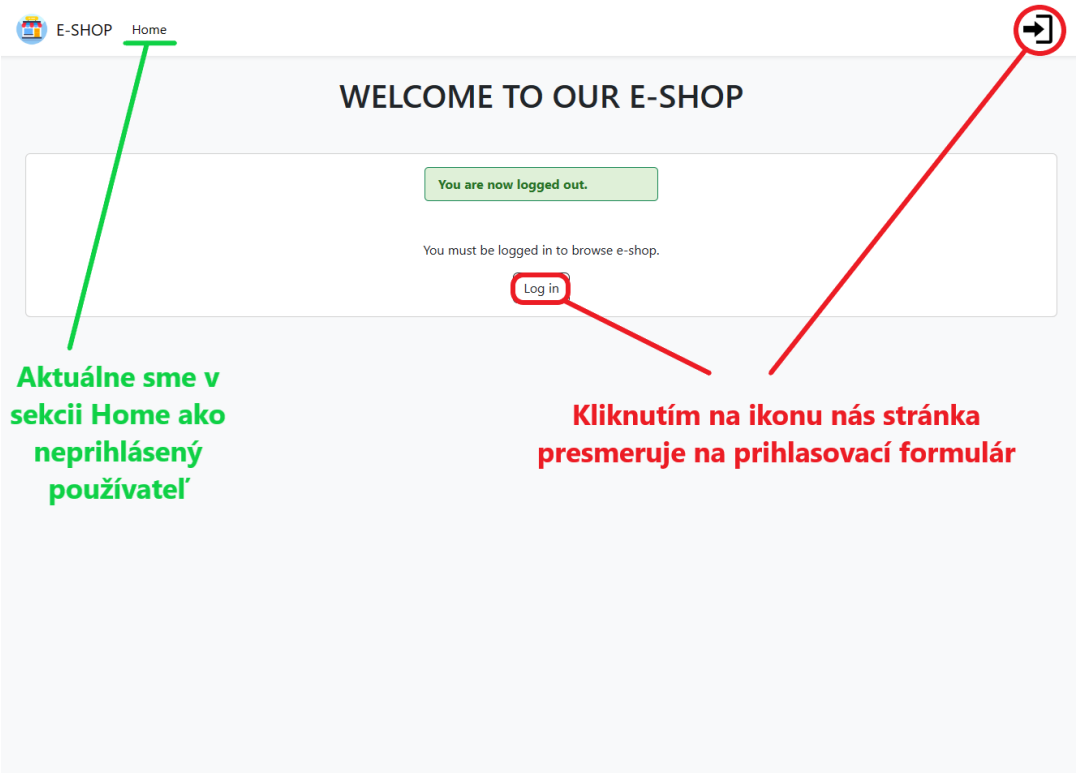
Dáta posielané v tele požiadavky z frontendu na backend sú zašifrované. Obsahujú symetrický kľúč asymetricky zašifrovaný verejným kľúčom backendu. Na získanie symetrického kľúča sa na backende najskôr odšifruje pomocou privátneho kľúča backendu. Následne sa symetricky odšifrujú ostatné dáta z tela požiadavky (údaje o objednávke, údaje o recenzii) pomocou získaného symetrického kľúča a spracujú sa (vytvorí sa objednávka, vytvorí sa recenzia).

Dáta odosielané z backendu ako odpoveď na spomínané požiadavky sú zašifrované symetrickým kľúčom, ktorý sa získal z tela požiadavky.

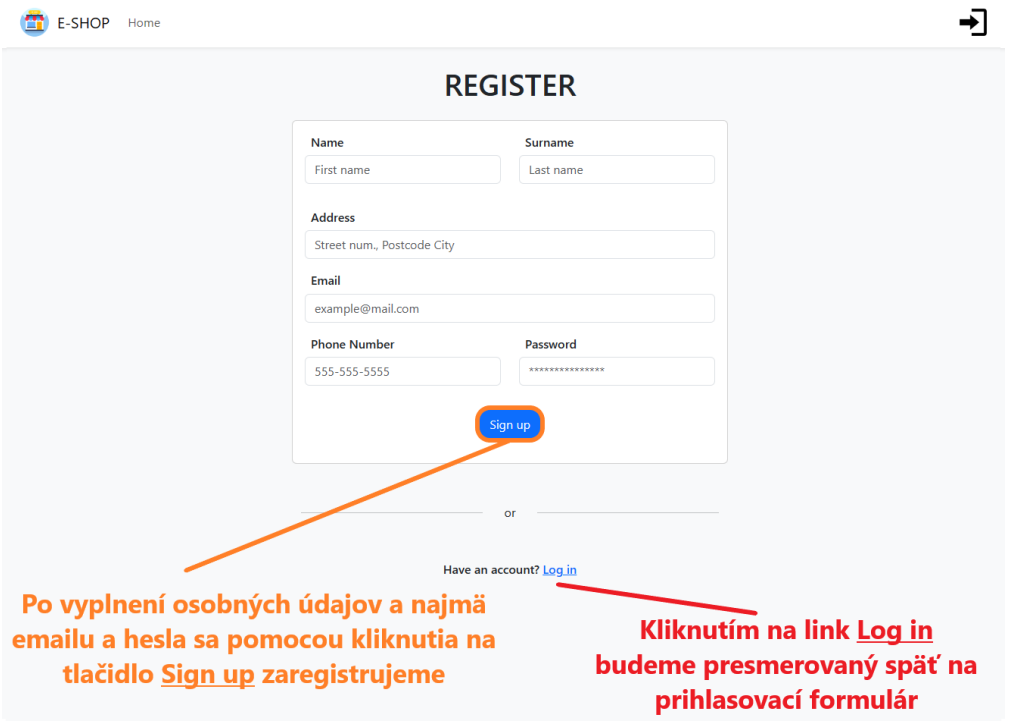
V dokumentáciách zadaní 2 a 3 bola konkrétna implementácia symetrického a asymetrického šifrovania a dešifrovania na backende podrobnejšie opísaná.

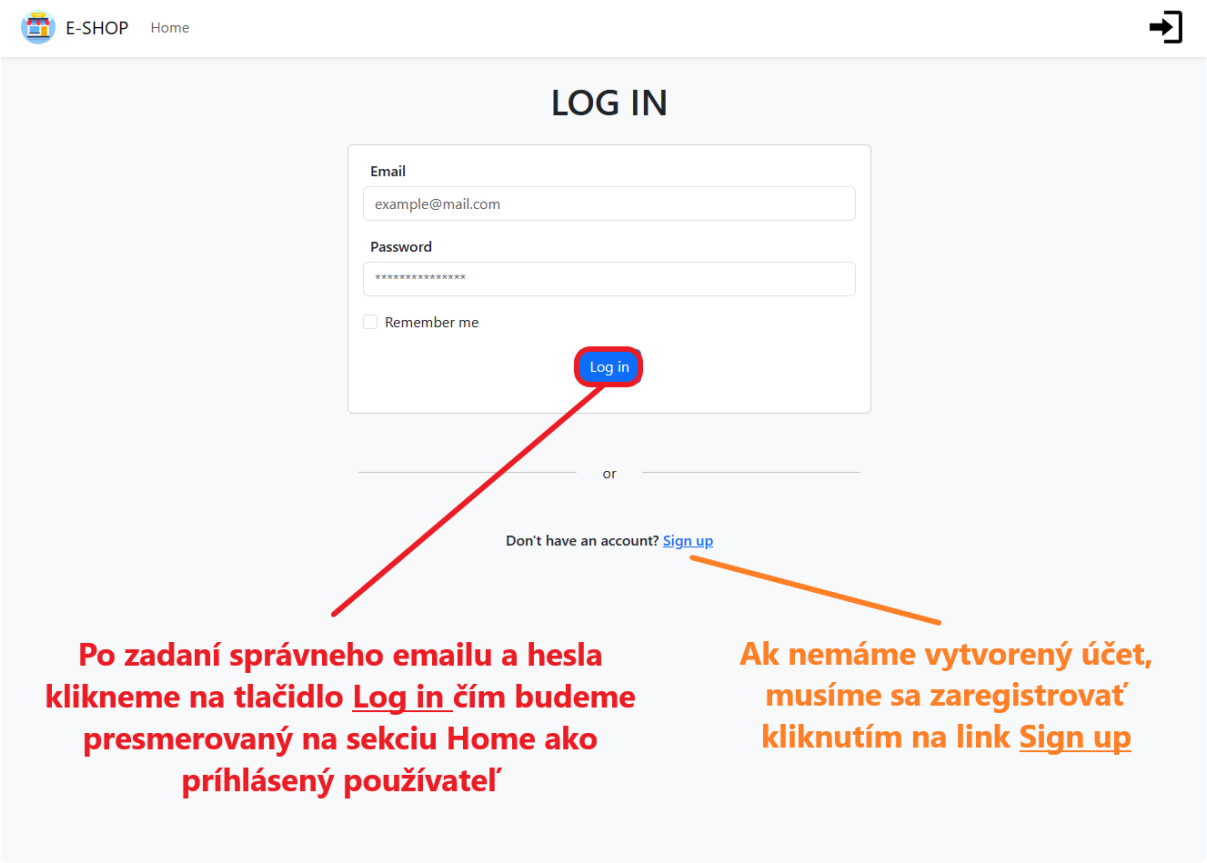
Frontend dokumentácia

Home



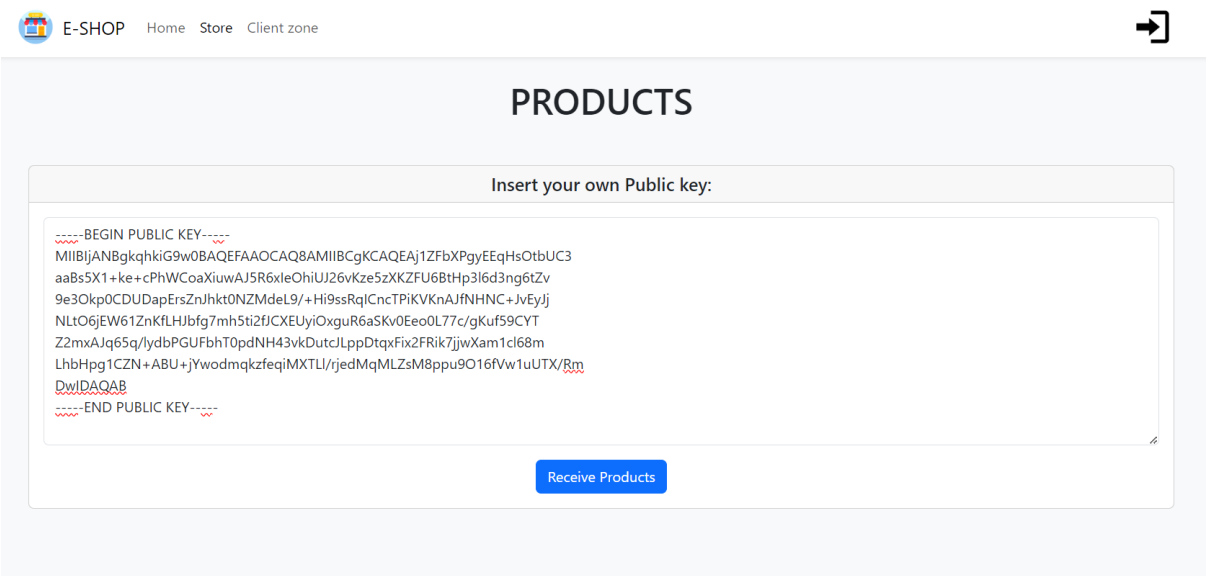
Login/register



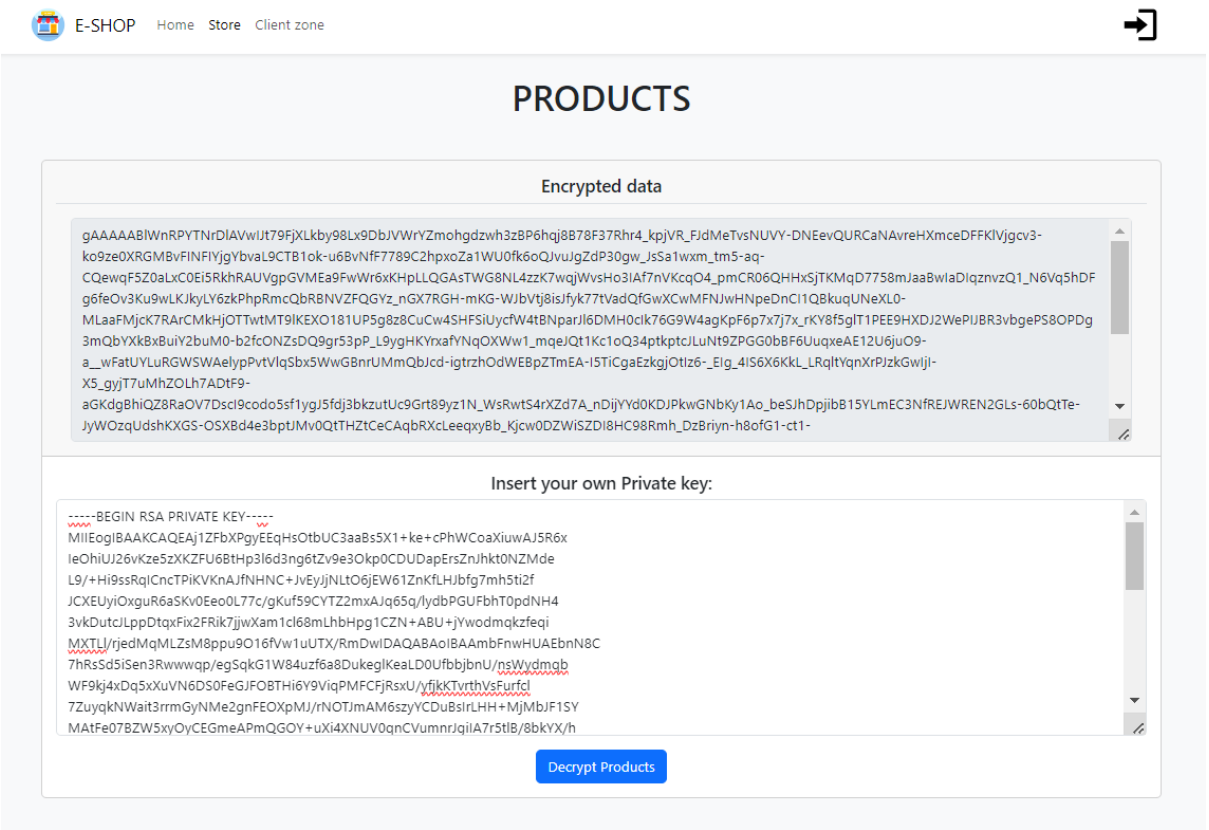


Zobrazenie produktov

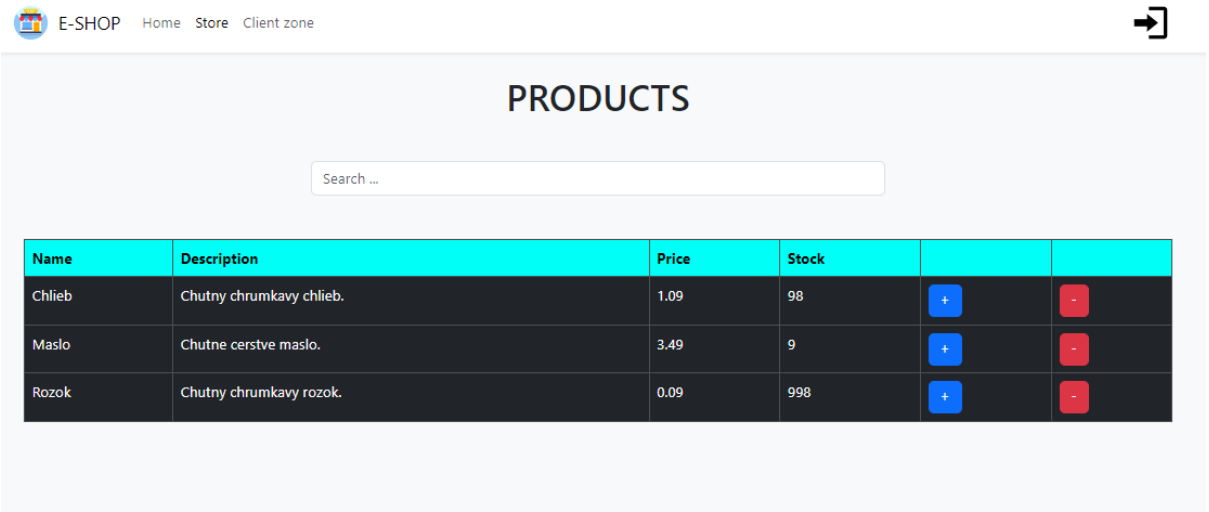
Táto funkcionalita je prístupná v záložke Store len prihlásenému používateľovi. Po kliknutí na túto podstránku v menu, bude používateľ vyzvaný na zadanie verejného kľúča (ktorým sa dáta o produktoch na backende zašifrujú kvôli bezpečnejšej komunikácii). Zadaný kľúč má byť v PEM formáte a mal by byť dlhý 2048 bitov. Kľúč v správnom formáte je možný vygenerovať na asledujúcej podstránke: https://cryptotools.net/rsagen. Podstránku Store so zadaným verejným kľúčom používateľa môžeme vidieť na nasledujúcom obrázku:



Pokiaľ bol zadaný kľúč v správnom formáte, po stlačení tlačidla Receive Products sa zobrazí výzva na zadanie privátneho kľúča používateľa (na odšifrovanie produktov na frontende). Zadaný kľúč má byť opäť vo formáte PEM a má byť dlhý 2048 bitov. Vzhľad postránky Store so zadaným privátnym kľúčom používateľa môžeme vidieť na nasledujúcom obrázku:

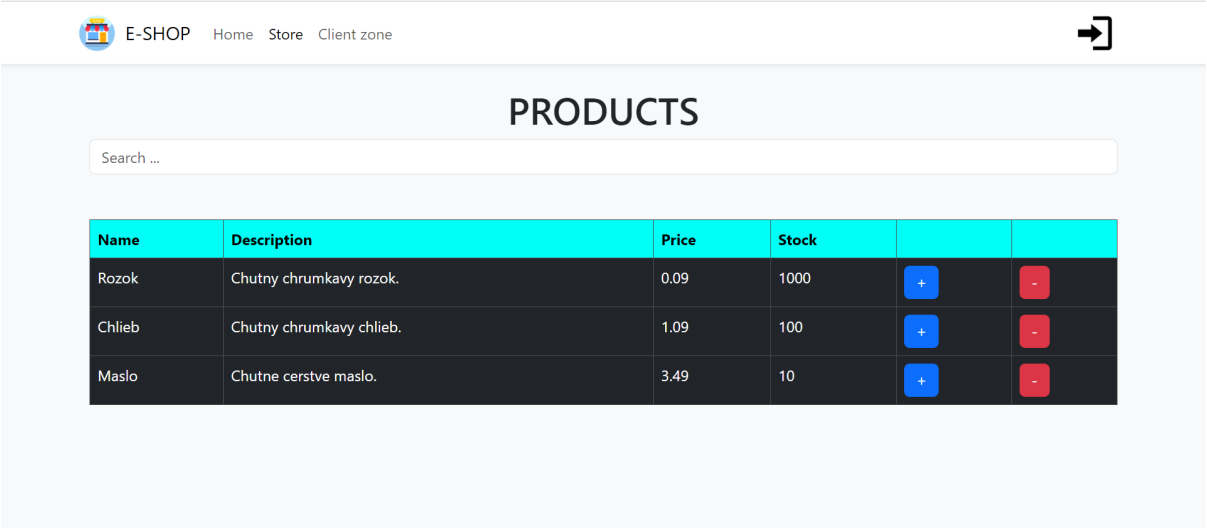


Pokiaľ bol zadaný privátny kľúč správny, po stlačení tlačidla Decrypt Products sa zobrazí tabuľka produktov nášho e-shopu, ktorá je zobrazená na nasledujúcom obrázku:

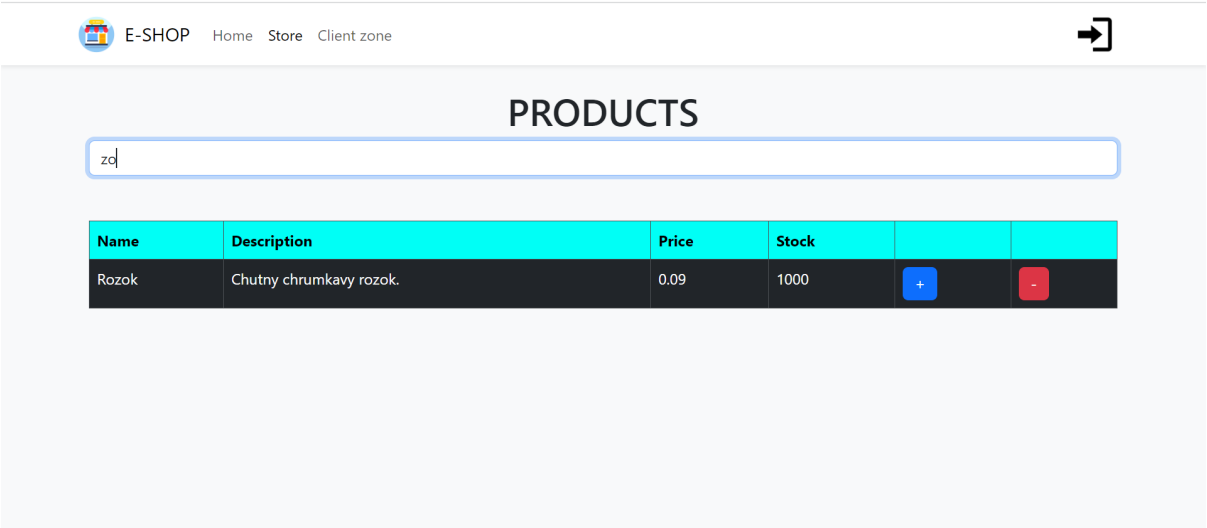


Searchbar a filtrovanie produktov

Táto funkcionalita je prístupná v záložke Store len prihlásenému používateľovi a po úspešnom odšifrovaní produktov, ktoré sa zobrazili v tabuľke. Filtrovanie funguje po každom zadanom znaku, pričom vždy sa posiela nový request na získanie produktov (filtrovaných). Neprebieha to teda na frontende. Zároveň už nie je nutné vždy zadať svoj verejný a privátny kľúč, nakoľko sa použijú tie, ktoré používateľ vložil pred prvotným zobrazením produktov.
Nefiltrovaný výber:

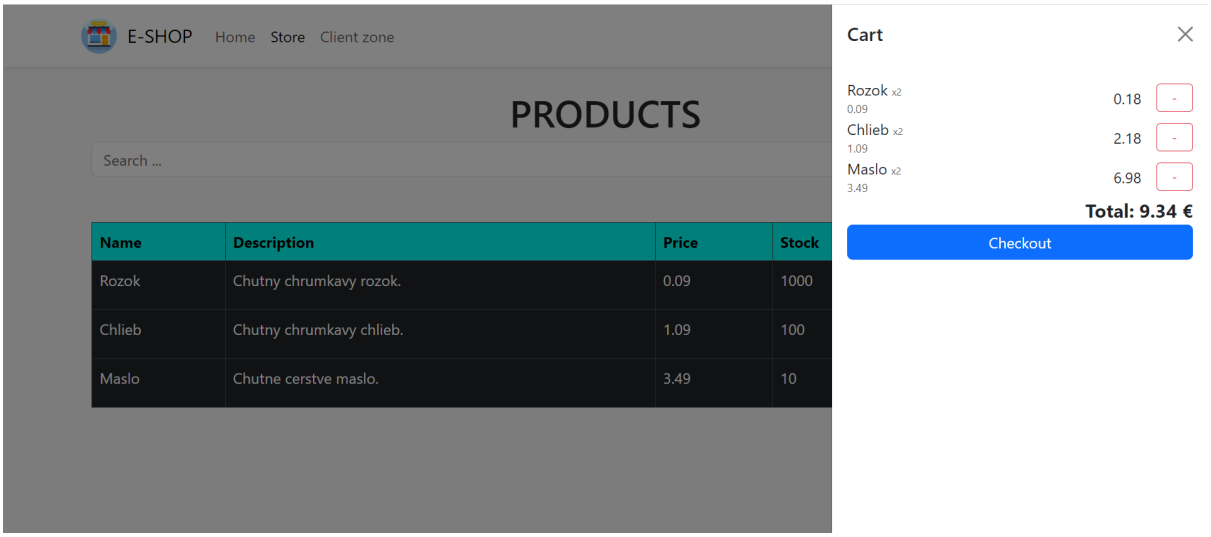


Filtrovaný výber:

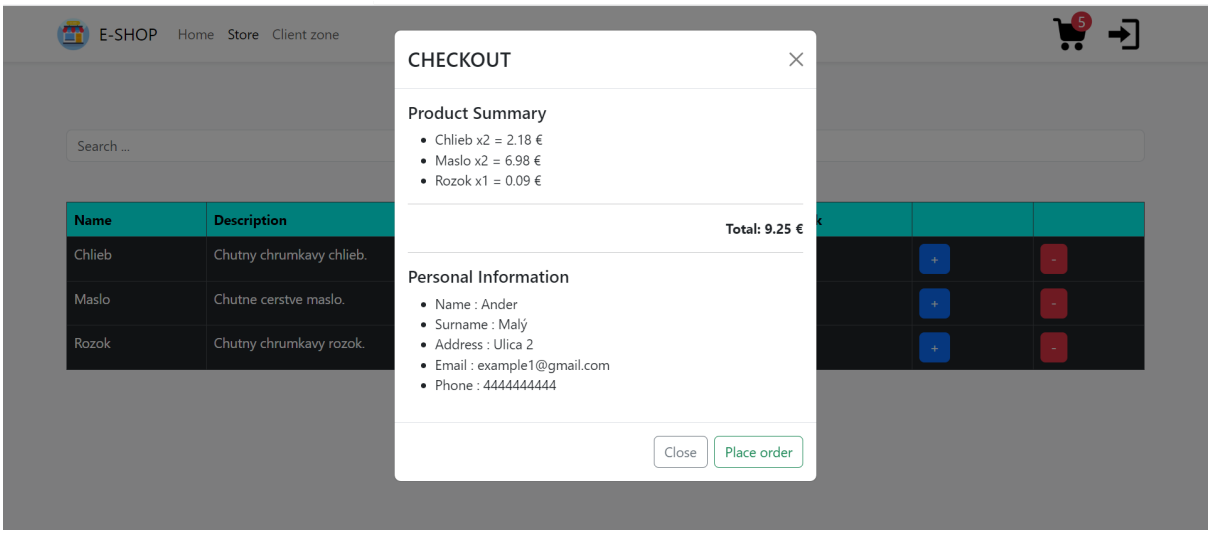


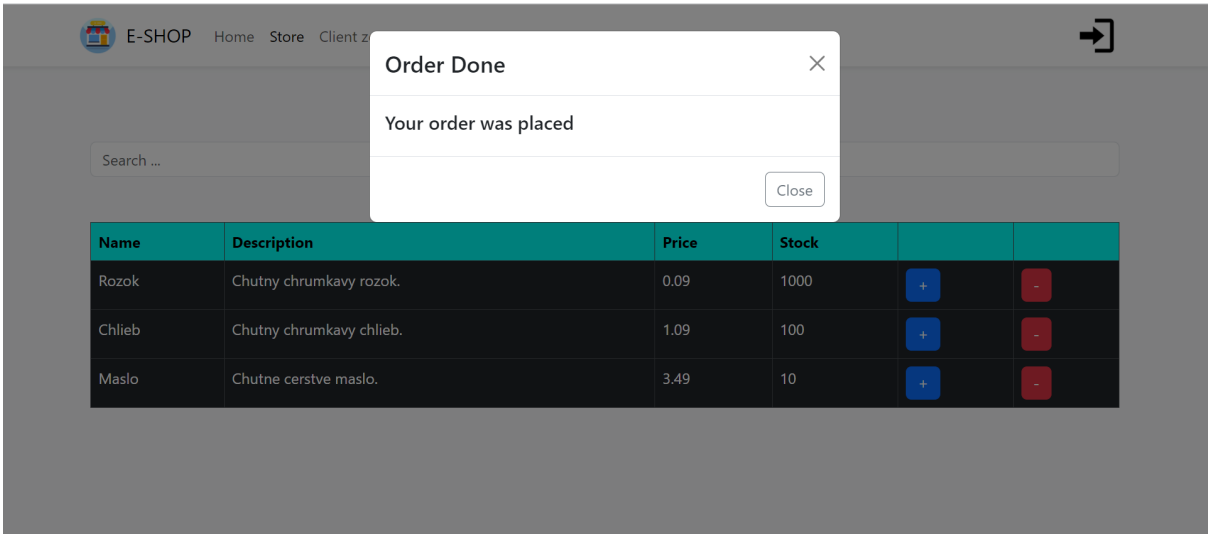
Košík

Do tabuľky boli vložené 2 tlačidlá na pridanie, resp. odobranie produktov do košíka. Počet kusov z jednotlivého produktu si používateľ vloží do košíka pomocou počtu kliknutí (pridanie/odobranie po jednom kuse). Ak košík nie je prázdny, tak sa v pravom hornom rohu zobrazí ikona košíka aj s počtom všetkých položiek v ňom. Po kliknutí na ikonu sa ešte dá počet jednotlivých produktov upraviť.



Následne po kliknutí na tlačidlo checkout sa zobrazí sumár košíka aj s údajmi používateľa. Svoje údaje používateľ nemôže meniť, automaticky sa použijú tie, ktoré vložil pri registrácii. Po odoslaní objednávky sa buď vypíše chyba, alebo sa otvorí nový modal s informáciou o vytvorení objednávky. Po zatvorení modalu sa automaticky presmeruje na Home.





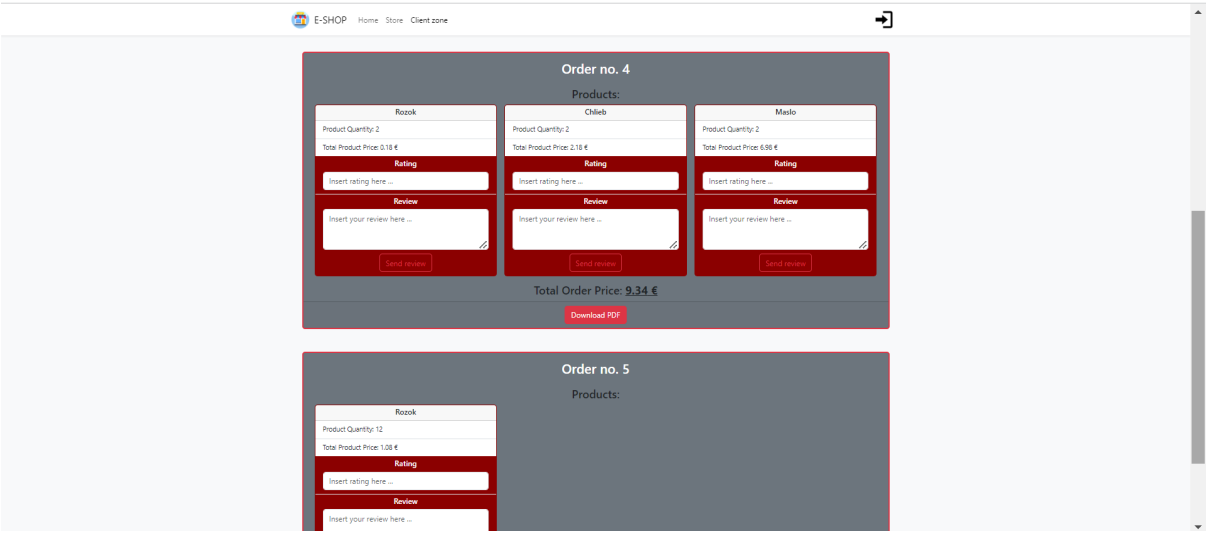
Samotný košík sa zobrazí len vtedy, ak sa používateľ dostane do časti, kde už vidí dešifrované dáta v tabuľke. Ak je v inej časti (napr. klientská zóna), tak ikona košíka nie je dostupná.

Vytvorenie objednávky

V košíku po odoslaní objednávky sa všetky potrebné dáta zašifrujú vytvoreným symetrickým kľúčom a ten sa následne ako zašifrovaný verejným kľúčom serveru pripojí k dátam a všetko sa odošle vo formáte json na server. Ak bola objednávka vytvorená, tak sa zároveň vrátia aj zašifrované dáta s informáciou o danej objednávke, aktuálne však s nimi nič nerobíme (vytvorené objednávky budú dostupné v klientskej zóne).

Klientská zóna

Ak je používateľ prihlásený, tak sa mu v hornom menu zobrazí záložka Client zone. Následne je nutné vykonať postup ako pri zobrazení produktov (vložiť verejný a privátny kľúč). Ak sa všetko vykonalo správne, tak používateľovi sa zobrazia všetky jeho objednávky.



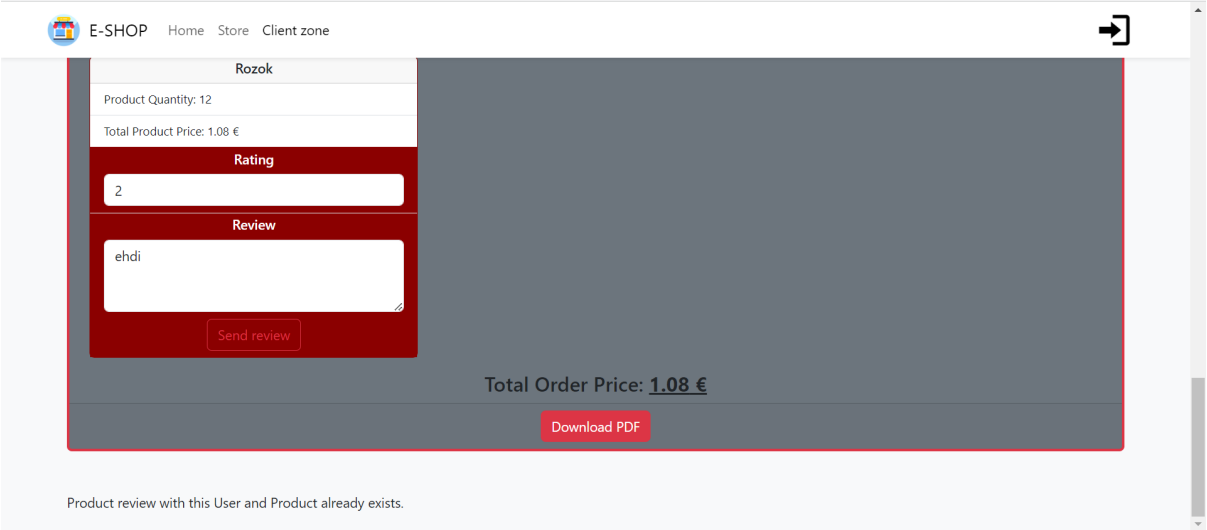
Ku každej objednávke sa zobrazia všetky produkty, ich počet, celková cena za každý produkt. Okrem toho tiež recenzia (číslo a text) s tlačidlom pridať recenziu (send review). Používateľ tu má tiež možnosť vygenerovať PDF so sumárom objednávky (tlačidlo Download PDF). Nakoniec je zobrazená aj celková cena nákupu.

Pridanie recenzie

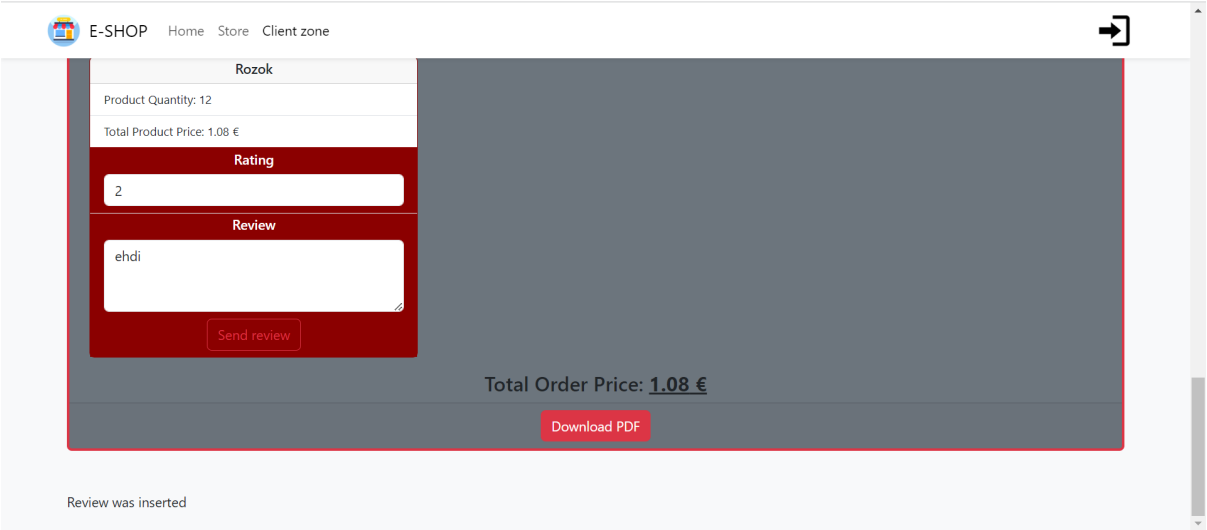
Ku každému produktu v každej objednávke sú zobrazené dva inputy: číselný na číselné hodnotenie produktu a textový na slovné hodnotenie. Ak sa vo viacerých objednávkach nachádza rovnaký produkt, tak tieto recenzie sú previazané, nakoľko sme zvolili politiku takú, že používateľ môže k jednému produktu mať iba jednu recenziu. Je to najmä kvôli zachovaniu konzistentnosti. Taktiež, ak už používateľ vložil recenziu, tak už ju nemôže upravovať.

Používateľ teda môže do inputov zadávať svoje vstupy. Po stlačení tlačidla na odoslanie (send review) sa načítajú dáta z inputov. Následne sa zašifrujú pomocou vytvoreného symetrického ten sa následne ako zašifrovaný verejným kľúčom serveru pripojí k dátam a všetko sa odošle vo formáte json na server. Ak používateľ chcel meniť recenziu k produktu,

ku ktorému už predtým recenziu vložil, tak to je zakázané a vráti sa chyba, ktorá sa vypíše v spodnej časti stránky.Sú možné aj ďalšie chyby, ktoré sa ošetrovali na backende.
Screenshot chybovej hlášky:



Ak sa ale recenzia správne vložila, tak sa používateľovi vypíše správa o úspešnom vložení.



Vygenerovanie PDF

V klientskej zóne má používateľ v každej objednávke možnosť zobraziť pdf so sumarizáciou danej objednávky. PDF sa generuje na frontende z dát, ktoré sme získali zo serveru (boli šifrované, dešifrovanie pomocou privátneho a symetrického kľúča) a ktoré tiež slúžia na zobrazenie všetkých dát v samotnej klientskej zóne. Takže pri generovaní PDF sa len použijú dáta asociované s danou objednávkou a z nich PDF vygeneruje.
Screenshot vygenerovaného PDF môžeme vidieť na nasledujúcej strane:

Invoice

Order ID: 6

Order Date: 19. 11. 2023 21:20:30

Bill To:

Ander Malý

Ulica 2

example1@gmail.com

4444444444

Products:

- Chlieb

  Description: Chutny chrumkavy chlieb.

  Quantity: 2

  Total Product Price: 2.18€

  Review: Good

- Maslo

  Description: Chutne cerstve maslo.

  Quantity: 2

  Total Product Price: 6.98€

- Rozok

  Description: Chutny chrumkavy rozok.

  Quantity: 1

  Total Product Price: 0.09€

  Review: ehdiehd

Total Order Price: 9.25€