

Studencki Projekt Wybranych Elementów Praktyki Projektowania Oprogramowania

Instytut Informatyki Uniwersytetu Wrocławskiego

Maksymilian Czudziak, Bartosz Jaśkiewicz, Andrzej Tkaczyk

Dokumentacja projektu Shopsio

Wrocław, 9 lutego 2021

Wersja 0.4

Tabela 0. Historia zmian w dokumencie

Data	Numer wersji	Opis	Autor
2021.01.29	0.1	Utworzenie dokumentu	Maksymilian Czudziak
2021.02.02	0.2	Dodanie przypadków użycia	Maksymilian Czudziak
2021.02.07	0.3	Dodanie opisu projektu	Bartosz Jaśkiewicz
2021.02.08	0.4	Dodanie konfiguracji	Bartosz Jaśkiewicz

Spis treści

1	Wprowadzenie	3
1.1	Cel dokumentu	3
2	Informacje o autorach	3
3	Opis projektu	3
4	Przypadki użycia	3
4.1	Przeglądanie zawartości sklepu	3
4.2	Wyszukiwanie po nazwie/opisie towaru	4
4.3	Rejestracja nowego konta	4
4.4	Logowanie się na konto	4
4.5	Kupowanie przedmiotu przez użytkownika zalogowanego	5
4.6	Dodanie/modyfikacja/usunięcie towaru	5
4.7	Przeglądanie listy złożonych/otwartych zamówień	6
5	Konfiguracja aplikacji	6
5.1	Instalacja	6
5.2	Uruchomienie	7

1. Wprowadzenie

1.1. Cel dokumentu

Niniejszy dokument ma na celu zebranie informacji o autorach, krótko opisać aplikację oraz jej przypadków użycia. Znajdą się tutaj także informacje o instalacji i uruchomieniu aplikacji. Dokument stanowi rozwiązanie siódmej listy zadań realizowanej w ramach pracowni z przedmiotu Wybrane Elementy Praktyki Projektowania Oprogramowania.

2. Informacje o autorach

Autorzy projektu:

- Maksymilian Czudziak 309498
- Bartosz Jaśkiewicz 307893
- Andrzej Tkaczyk 309181

3. Opis projektu

Niniejszy projekt jest stroną, spełniającą podstawowe funkcje sklepu internetowego. Niektóre ze zrealizowanych funkcji to: przeglądanie produktów, filtrowanie produktów, dodawanie/usuwanie produktów do/z koszyka, zarejestrowanie zamówienia, możliwość posiadania konta przez użytkowników oraz podstawowe funkcje administratorskie takie jak zarządzanie produktami i przeglądanie zamówień, kategorii oraz użytkowników. Strona posiada stronę główną, a także widoki do przeglądania listy produktów, konta użytkownika, koszyka, podsumowania i panelu administratora.

Aplikacja jest dostępna online w serwisie heroku pod adresem:
<https://shopsio.herokuapp.com/>

4. Przypadki użycia

4.1. Przeglądanie zawartości sklepu

- Aktor: Użytkownik niezalogowany
- Warunki początkowe: Brak znaczących warunków
- Zdarzenie inicjujące: Wejście na stronę sklepu
- Scenariusz:
 - System wyświetla stronę domową sklepu
 - Użytkownik Wybiera jedną z trzech kategorii na pasku nawigacyjnym na górze ekranu lub "Wszystkie przedmioty"
 - System wyświetla odpowiedni listing przedmiotów
- Skutek końcowy: Użytkownik może przeglądać odpowiedni fragment zawartości sklepu

4.2. Wyszukiwanie po nazwie/opisie towaru

- Aktor: Użytkownik niezalogowany
- Warunki początkowe: Wyszukiwany przedmiot jest w asortymencie sklepu
- Zdarzenie inicjujące: Wejście stronę główną sklepu lub jeden z listingów
- Scenariusz:
 - System wyświetla stronę domową sklepu lub jeden z listingów produktów
 - Użytkownik klika w pole wyszukiwania produktów
 - System podświetla pole tekstowe wyszukiwania
 - Użytkownik wpisuje całą lub fragment nazwy produktu, który chciałby wyszukać
 - System wyświetla odpowiedni listing z wynikami wyszukiwania
- Skutek końcowy: Użytkownik znalazł szukany przedmiot lub ustalił, że go nie ma w sprzedaży.

4.3. Rejestracja nowego konta

- Aktor: Użytkownik
- Warunki początkowe: Brak znaczących warunków
- Zdarzenie inicjujące: Użytkownik wybrał opcję "Zarejestruj"
- Scenariusz:
 - System wyświetla podstronę odpowiadającą za rejestrację nowych użytkowników
 - Użytkownik uzupełnia pola "nazwa użytkownika" i "hasło" klikając guzik zarejestruj
 - System umieszcza te dane w bazie danych rejestrując użytkownika oraz przekierowuje na stronę główną z dodatkową informacją o pomyślnym przebiegu rejestracji
- Sytuacje wyjątkowe: Rejestracja nie powiedzie się jeśli konto o takiej samej nazwie użytkownika już istnieje
- Skutek końcowy: Użytkownik założył konto

4.4. Logowanie się na konto

- Aktor: Użytkownik
- Warunki początkowe: Brak znaczących warunków
- Zdarzenie inicjujące: Użytkownik wybrał opcję "Zaloguj"
- Scenariusz:
 - System wyświetla podstronę odpowiadającą za logowanie użytkowników
 - Użytkownik wpisuje w odpowiednie pola swoją nazwę użytkownika i hasło
 - System pomyślnie loguje użytkownika przekierowując na stronę główną z dodatkową informacją o pomyślnym logowaniu

- Sytuacje wyjątkowe: Jeśli użytkownik podał błędną nazwę użytkownika lub/i hasło to nie zostanie zalogowany, pozostanie na witrynie logowania
- Skutek końcowy: Użytkownik został zalogowany

4.5. Kupowanie przedmiotu przez użytkownika zalogowanego

- Aktor: Użytkownik zalogowany
- Warunki początkowe: Przedmiot, który zostanie kupiony jest w asortymencie sklepu
- Zdarzenie inicjujące: Wejście na stronę naszego sklepu
- Scenariusz:
 - System wyświetla stronę główną sklepu
 - Użytkownik klika w interesującą dla niego kategorię lub "Wszystkie produkty"
 - System wyświetla listę przedmiotów oferowanych do kupienia
 - Użytkownik wybiera na interesujących go przedmiotach "Dodaj do koszyka"
 - System dodaje do jego koszyka wybrane przedmioty
 - Użytkownik klika opcję "Przejdź do płatności" obok ikonki koszyka.
 - System wyświetla koszyk użytkownika z sumą do zapłacenia
 - Użytkownik wybiera opcję "Przejdź do podsumowania"
 - System wyświetla podstowę podsumowania
 - Użytkownik wybiera w opcję dostawy i metodę płatności oraz uzupełnia dane do dostawy, a następnie klika "Kupuję i płacę"
 - System wyświetla podstronę podziękowania za zakup
- Sytuacje wyjątkowe: Jeśli użytkownik nie będzie zalogowany to guzik "przejdź do płatności" przeniesie użytkownika na podstronę informującą o wymogu założenia konta w celu zakupu.
- Skutek końcowy: Przedmiot został kupiony, w bazie danych zostało to zapisane dla tego użytkownika

4.6. Dodanie/modyfikacja/usunięcie towaru

- Aktor: Administrator
- Warunki początkowe: Brak znaczących warunków
- Zdarzenie inicjujące: Administrator się zalogował
- Scenariusz:
 - System wyświetla panel administratora
 - Administrator wybiera opcję "Produkty"
 - System wyświetla listę wszystkich produktów, które są w bazie danych sklepu
 - Administrator wybiera opcję "Dodaj produkt"

- System wyświetla formularz edycji produktu z pustymi polami informacji o produkcie
 - Administrator uzupełnia nazwę, cenę, wielkość, kolor, liczbę sztuk, status, opis i zdjęcie produktu, a następnie klika "Zapisz"
 - System zapisuje produkt w bazie danych, a następnie prezentuje listę wszystkich produktów w asortymencie z nowym, dodanym produktem
 - Administrator wybiera jeden z produktów
 - System prezentuje podstronę "Edycji produktu"
 - Administrator zmienia interesujące go pola i klika opcję "Zapisz"
 - System wyświetla listę wszystkich produktów, które są w bazie danych sklepu z nowymi danymi o zmienionym przedmiocie
 - Administrator wybiera opcję "usuń" przy wybranym towarze
 - System usuwa wybrany towar z listy oraz bazy danych
- Sytuacje wyjątkowe: Lista jest pusta, jeśli nie ma żadnego produktu
 - Skutek końcowy: Administrator wyświetlił, dodał, zmodyfikował i usunął produkt

4.7. Przeglądanie listy złożonych/otwartych zamówień

- Aktor: Administrator
- Warunki początkowe: Brak znaczących warunków
- Zdarzenie inicjujące: Administrator się loguje
- Scenariusz:
 - System wyświetla panel administratora
 - Administrator wybiera opcję "Zamówienia"
 - System prezentuje listę wszystkich zamówień złożonych w sklepie, przy każdym zamówieniu jest informacja o jego statusie
 - Administrator może wybrać dowolną z pozycji, aby uzyskać więcej informacji
- Sytuacje wyjątkowe: Lista zamówień jest pusta, jeśli żadne zamówienie nie zostało złożone w systemie.
- Skutek końcowy: Administrator przegląda listę zamówień i ich statusów w systemie

5. Konfiguracja aplikacji

5.1. Instalacja

Do poprawnego działania projektu, potrzebne są następujące programy (w nawiasach podano linki do stron producentów, gdzie znajdują się pełne instrukcje instalacji):

- node.js (<https://nodejs.org/>)
- npm (<https://www.npmjs.com/>)

- PostgreSQL (<https://www.postgresql.org/>)

Po instalacji powyższych należy skonfigurować bazę danych w programie PostgreSQL. Można to zrobić wykorzystując program **psql**. Przykładową bazę danych i użytkownika można skonfigurować następująco:

1. **psql -U postgres**
2. Wpisz hasło użytkownika postgres, zwykle zdefiniowane w trakcie instalacji.
3. **CREATE USER weppo WITH PASSWORD 'admin';**
4. **CREATE DATABASE shopsio;**
5. **GRANT ALL PRIVILEGES ON DATABASE shopsio TO weppo;**
6. **\q**

Następnie należy zmienić nazwę pliku **.env.default** na **.env**. W tym pliku w odpowiednich polach, tuż po znakach równości (bez spacji), trzeba wpisać dane potrzebne do połączenia się ze skonfigurowaną bazą danych. Przykładowy plik wygląda następująco:

```
1 DB_USER=weppo
2 DB_PASSWORD=admin
3 DB_HOST=localhost
4 DB_PORT=5432
5 DB_DATABASE=shopsio
```

Po poprawnym wpisaniu danych i zapisaniu pliku, w lokalizacji głównej projektu należy uruchomić polecenie: **npm install**, a następnie **npm run db-reset**.

5.2. Uruchomienie

By uruchomić projekt należy wywołać polecenie: **npm start**. Serwis będzie dostępny pod adresem **localhost:8080**.

W bazie zostały zainicjowane przykładowe konta dla użytkowników i dla administratora. By zalogować się na dane konto, należy przejść pod adres **/login** (przejdzie przez przycisk "Zaloguj" w pasku nawigacyjnym) i wpisać dane jednego z poniższych kont:

- konto administratora - LOGIN: admin1 HASŁO: adminpassword1
- LOGIN: user1 HASŁO: userpassword1
- LOGIN: user2 HASŁO: userpassword2
- LOGIN: user3 HASŁO: userpassword3