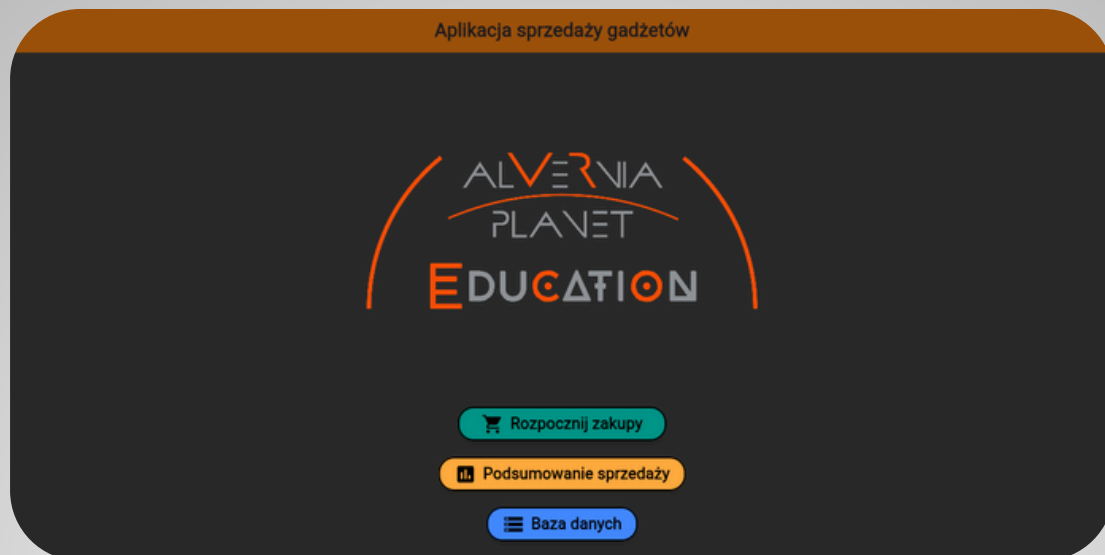




Aplikacja Sklepiku AP

Dokumentacja projektu

'AplikacjaSklepikuAP'



1. Cel projektu

Celem projektu jest stworzenie zaawansowanej aplikacji webowej, która umożliwia kompleksowe zarządzanie sprzedażą gadżetów w środowisku lokalnym. Aplikacja została zaprojektowana w celu ułatwienia śledzenia stanu magazynowego w czasie rzeczywistym, zapewniając użytkownikowi intuicyjny interfejs do obsługi koszyka zakupowego, gdzie można dokładnie określić ilość, cenę i rabaty dla wybranych produktów.

Dodatkowo, aplikacja umożliwia szybkie i bezproblemowe finalizowanie transakcji dzięki obsłudze różnych metod płatności, takich jak karta lub gotówka. Jedną z kluczowych funkcji jest także generowanie szczegółowych raportów sprzedaży, które mogą być eksportowane do formatu PDF. Raporty te są przejrzyste i czytelne, co ułatwia analizę sprzedaży oraz planowanie przyszłych działań biznesowych.

2. Główne funkcjonalności

2.1. Zarządzanie produktami

- Pobieranie danych o gadżetach z bazy Firebase Firestore.
- Wyświetlanie listy produktów dostępnych w magazynie.
- Możliwość edycji:
 - Ilości dostępnych produktów.
 - Ceny produktów.
 - Rabatów na poszczególne produkty.

2.2. Koszyk zakupowy

- Dodawanie produktów do koszyka.
- Uwzględnianie rabatów na pojedynczy produkt.
- Automatyczne obliczanie całkowitej ceny koszyka, z uwzględnieniem rabatów.
- Wyświetlanie podsumowania koszyka przed finalizacją zakupu.

2.3. Obsługa transakcji

- Wybór metody płatności (karta/gotówka).
- Aktualizacja danych o sprzedanych produktach oraz zrealizowanych transakcjach.

2.4. Podsumowanie sprzedaży

- Generowanie podsumowania transakcji z podziałem na płatności kartą i gotówką.
- Eksportowanie raportów do pliku PDF w celu archiwizacji lub analizy.

3. Technologie i narzędzia

- Flutter: do budowy aplikacji webowej.
- Firebase Firestore: baza danych w chmurze do przechowywania informacji o gadżetach i transakcjach.
- Pakiet pdf: do generowania raportów w formacie PDF.
- Dart: język programowania używany w Flutterze.
- Firebase Core: do integracji aplikacji z Firebase.

4. Architektura aplikacji

Aplikacja opiera się na trójwarstwowym podejściu:

1. Warstwa prezentacji:

- Wyświetlanie danych użytkownikowi.
- Interfejs użytkownika zbudowany w Flutterze.
- Ekrany takie jak: HomePage, GadzetSelectionScreen, DatabasePage.

2. Warstwa logiki biznesowej:

- Obliczenia, takie jak rabaty, podsumowania koszyka i sprzedaży.
- Aktualizacje w bazie Firebase.

3. Warstwa danych:

- Firebase Firestore jako główne źródło danych o produktach.
- Dane o transakcjach przechowywane lokalnie w stanie globalnym aplikacji.

5. Struktura katalogów

lib/

-- models/	
-- gadzet.dart	# Model gadżetu
-- screens/	
-- home_page.dart	# Ekran główny aplikacji
-- gadzet_selection_screen.dart	# Ekran wyboru gadżetów
-- koszyk_screen.dart	# Koszyk zakupowy
-- payment_screen.dart	# Finalizacja płatności
-- database_page.dart	# Zarządzanie bazą danych
-- summary_page.dart	# Podsumowanie transakcji
-- utils/	
-- global_state.dart	# Stan globalny aplikacji

6. Modele danych

6.1. ****Model** ******Gadzet

- Reprezentuje produkt w aplikacji.
- Atrybuty:
 - nazwa – nazwa gadżetu.
 - cena – cena jednostkowa.
 - rabat – rabat na pojedynczy produkt.
 - obrazek – ścieżka do obrazu produktu.

Przykładowa implementacja:

```
class Gadzet {  
  String nazwa;  
  double cena;  
  double rabat;  
  String obrazek;  
  
  Gadzet({  
    required this.nazwa,  
    required this.cena,  
    required this.rabat,  
    required this.obrazek,  
  });  
  
  Gadzet.fromFirestore(Map<String, dynamic> data)  
  // Metoda  
  factory Gadzet.fromFirestore(Map<String, dynamic> data) {  
    return Gadzet(  
      nazwa: data['Nazwa'] ?? 'Nieznany',  
      cena: (data['Cena'] is num) ? (data['Cena'] as num).toDouble() : 0.0,  
      rabat: (data['Rabat'] is num) ? (data['Rabat'] as num).toDouble() : 0,  
      obrazek: 'assets/gadzet/${data['Nazwa']}.png',  
    );  
  }  
}
```

7. Opis funkcjonalności

7.1. Ekran główny (HomePage*****)

- Umożliwia użytkownikowi przejście do trzech głównych funkcji aplikacji:
 - a. Wybór gadżetów.
 - b. Zarządzanie bazą danych.
 - c. Podsumowanie sprzedaży.

7.2. Zarządzanie bazą danych (DatabasePage*****)

- Wyświetla wszystkie produkty z bazy Firebase.
- Umożliwia edycję:
 - Ilości produktów w magazynie.
 - Ceny produktów.
 - Rabatów.

7.3. Koszyk zakupowy (KoszykWidget*****)

- Umożliwia dodawanie i usuwanie produktów.
- Naliczanie rabatów na jedną sztukę produktu.
- Wyświetlanie podsumowania koszyka z rozbiciem na:
 - Liczbę produktów.
 - Całkowitą cenę koszyka.
 - Zaoszczędzoną kwotę dzięki rabatom.

7.4. Finalizacja transakcji (PaymentScreen*****)

- Obsługa wyboru metody płatności.
- Finalizacja zakupu:
 - Aktualizacja stanu globalnego o transakcje gotówkowe i kartą.
 - Reset koszyka.

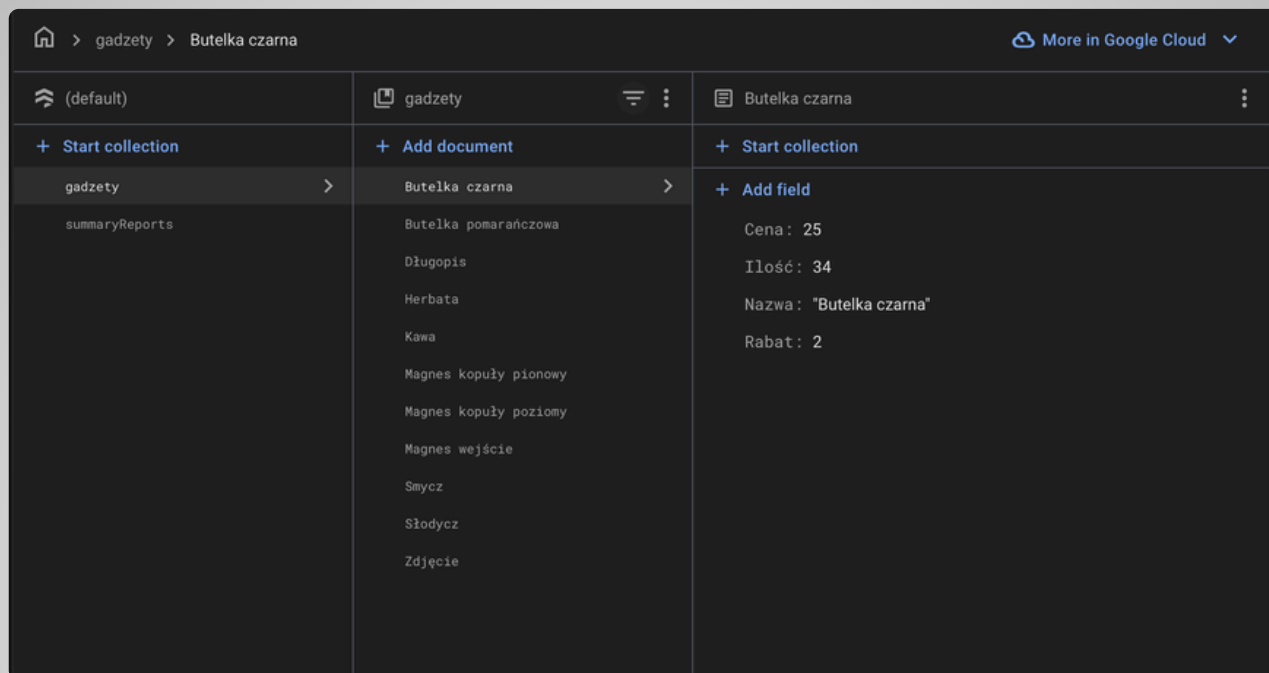
7.5. Generowanie raportów sprzedaży (TransactionsSummaryScreen*****)

- Grupowanie transakcji według:
 - Płatności kartą.
 - Płatności gotówką.
 - Ogólnych transakcji.
- Eksport raportu w formacie PDF.

8. Struktura kolekcji w Firebase Firestore

Kolekcja: *gadzet*

Przykładowy dokument:



9. Instalacja i uruchomienie

1.) Sklonuj repozytorium:

```
git clone <URL-repozytorium>
cd sklepik
```

2.) Zainstaluj zależności:

```
flutter pub get
```

3.) Uruchom aplikację:

```
flutter run
```

Autorzy oraz podział obowiązków

Autorzy

SzymonCzermak – Scrum Master, Initiator, Developer.

Patryk Kostecki – Developer.

Oscar Czempiel – Developer.