

## **Dokumentacja techniczna projektu „Kafejka”**

**Autorzy: Drużyna Ognia (Klaudia Gawrysiak, Aleksander Chocaj)**

---

### **Tytuł projektu:**

**Strona internetowa z programem lojalnościowym klientów kawiarni Kafejka**

---

### **Opis projektu**

Jako projekt stworzyliśmy stronę internetową dla kawiarni Kafejka. Strona zawiera informacje o kawiarni (historia kawiarni, lokalizacja, kontakt, pozycja dnia, menu itp.). Dodatkowo na niej klienci mogą dołączać do programu lojalnościowego przez założenie konta. Na koncie uczestnika programu można zapisać swój pobyt w kawiarni oraz zakupione pozycje z menu, a potem można je przeglądać. Każda wizyta (jeden kod z paragonu podawany przy zapisywaniu pobytu) daje punkty (1 punkt = 1 zł), następnie punkty są zamieniane na “pieczątki” (100 punktów = 1 pieczątką), a po zdobyciu odpowiedniej ilości pieczątek (5 pieczątek = 1 darmowy produkt) użytkownik ma prawo do ubiegania się o bonus lojalnościowy. Gdy użytkownik zbiera pieczątki może podejrzeć o jaką nagrodę może się ubiegać (ulubiona pozycja z menu, która była najwięcej razy kupowana). Strona posiada jednego administratora, który może zatwierdzić lub dodać za użytkownika wizytę (sprawdza manualnie poprawność informacji pod względem zgodności z danymi z kawiarni). Administrator może też dodawać nowe pozycje i rodzaje do menu kawiarni.

---

### **Specyfikacja wykorzystanych technologii**

- **Język programowania:** C#, HTML, CSS
  - **Framework:** ASP.NET MVC
  - **Wersja .NET:** .NET 8.0
  - **Baza danych:** MS SQL Server
  - **Narzędzie ORM:** Entity Framework Core, ASP.NET Core Identity
  - **Inne biblioteki/narzędzia:** Bootstrap, jQuery,
  - **Środowisko IDE:** Visual Studio 2022
- 

### **Instrukcja pierwszego uruchomienia projektu:**

1. Otwórz Visual Studio 2022
2. Wybierz opcję “Klonuj repozytorium”
3. Zapoznaj się z plikiem README.md
4. W lokalizacji repozytorium wpisz: <https://github.com/vKlaudyGaw/Kafejka.git>
5. Przenieś plik KafejkaDatabase.mdf z folderu głównego do folderu  
\\Kafejka\\bin\\Debug\\net8.0

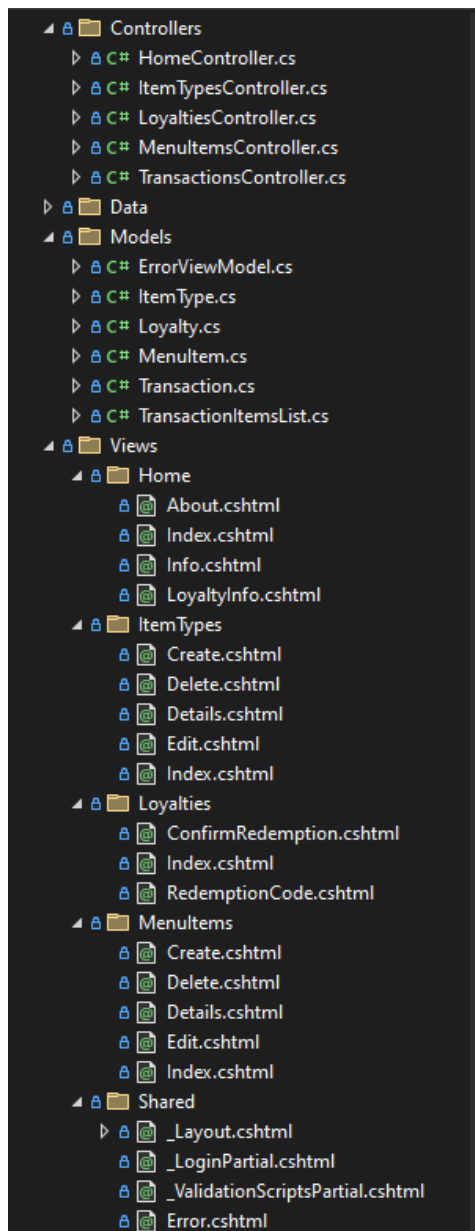
## 6. Skompiluj projekt

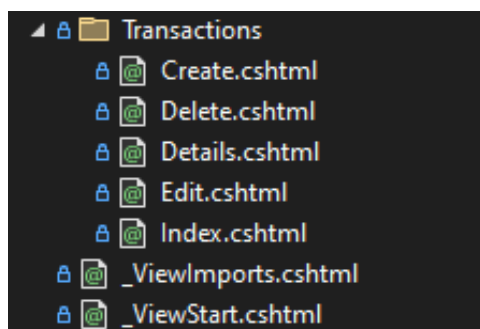
---

### Struktura projektu

#### Model-View-Controller (MVC)

Projekt jest zorganizowany zgodnie z architekturą MVC.





## Opis komponentów:

### 1. Modele:

Model	Opis	Pola	Ograniczenia
ItemType	model przechowujący informacje o rodzaju pozycji	Id, Name	Id: klucz główny Name: wymagane, 1-150 znaków
MenuItem	model przechowujący informacje o pozycji w menu	Id, Name, Description, Price, ItemTypeId, Type	Id: klucz główny Name: wymagane 1-150 znaków Description: wymagane 1-1000 znaków Price: wymagane, 0-10000 ItemTypeId: wymagane
Transaction	model przechowujący informacje o wizycie użytkownika	Id, Amount, Code, PurchaseTime, Approved, TransactionItemList, UserId, User	Id: klucz główny Amount: wymagane, liczba całkowita od 1 do 999999 Code: Wymagane. Dokładnie 10 znaków alfanumerycznych. Musi być unikalny (zapewnione przez [Index(nameof(Code), IsUnique = true)]). PurchaseTime: wymagane TransactionItemList: powiązanie z modelem TransactionItemList
TransactionItemList	model tabeli pomocniczej przy	Id, TransactionId, Transaction,	Id: klucz główny TransactionId:

	tworzeniu połączenia MenuItem i Transaction	MenuItemId, MenuItem, Quantity	Wymagane. Klucz obcy do modelu Transaction Transaction: Obiekt nawigacyjny powiązany z modelem Transaction MenuItemId: Wymagane. Klucz obcy do modelu MenuItem MenuItem: Obiekt nawigacyjny powiązany z modelem MenuItem. Quantity: wymagane
Loyalty	model przechowujący informacje o programie lojalnościowym	Id, TotalPoints, NumberOfStampsUses, Transactions, CurrentPoints, CurrentStamps, FreeProductsAvailable, CurrentReward, CalculateCurrentReward, UserId, User	Id: Klucz główny TotalPoints: Wymagane NumberOfStampsUses: wymagane Transactions: powiązane z użytkownikiem

## 2. Kontrolery:

- HomeController – zarządza publicznymi podstronami

Index	GET	brak	Główna strona aplikacji. Pobiera pozycję menu dnia na podstawie bieżącego dnia tygodnia. Jeśli pozycja nie istnieje, widok zostaje załadowany bez tej informacji.	Widok Index z opcjonalną informacją o pozycji menu dnia (DayMenuItem).
About	GET	brak	Wyświetla stronę z	Widok About.

			informacjami o kawiarni	
Info	GET	brak	Wyświetla stronę z informacjami o kontakcie z kawiarnią.	Widok Info.
LoyaltyInfo	GET	brak	Wyświetla stronę z informacjami o programie lojalnościowym.	Widok LoyaltyInfo.
Error	GET	brak	Wyświetla stronę błędu aplikacji z informacją o identyfikatorze bieżącego żądania. Metoda jest opatrzona atrybutem ResponseCache z parametrami, które zapobiegają buforowaniu odpowiedzi.	Widok Error z modelem zawierającym RequestId

- ItemTypesController – zajmuje się zarządzaniem rodzajami pozycji

Metoda	HTTP	Parametry	Opis	Zwracane dane
Index	GET	brak	Wyświetla listę wszystkich rodzajów pozycji z menu	Widok z listą obiektów ItemType
Details	GET	int? id: Identyfikator typu rodzaju	Wyświetla szczegóły wybranego rodzaju pozycji. Jeśli id jest null lub obiekt nie istnieje, zwraca NotFound().	Widok z danymi szczegółowymi obiektu ItemType.
Create	GET	brak	Wyświetla formularz tworzenia nowego rodzaju pozycji.	Widok z formularzem tworzenia obiektu ItemType.
Create	POST	ItemType itemType: Dane nowego rodzaju pozycji przekazane w żądaniu.	Tworzy nowy rodzaj pozycji. Waliduje dane, zapisuje je w bazie danych i przekierowuje do metody Index.	Przekierowanie do Index lub widok formularza w przypadku błędów walidacji.
Edit	GET	int? id: Identyfikator typu pozycji do edycji.	Wyświetla formularz edycji istniejącego rodzaju pozycji. Jeśli id jest null lub obiekt nie istnieje, zwraca NotFound().	Widok z formularzem edycji obiektu ItemType.
Edit	POST	int id: Identyfikator edytowanego	Aktualizuje dane rodzaju pozycji w bazie danych.	Przekierowanie do Index lub widok z

		typu.  ItemType itemType: Zmodyfikowane dane rodzaju pozycji.	Sprawdza poprawność danych i obsługuje wyjątki związane z jednoczesnym dostępem do tych samych danych.	formularzem w przypadku błędów.
Delete	GET	int? id: Identyfikator typu pozycji do usunięcia.	Wyświetla szczegóły rodzaju pozycji przeznaczonej do usunięcia. Jeśli id jest null lub obiekt nie istnieje, zwraca NotFound().	Widok z potwierdzeniem usunięcia obiektu ItemType.
DeleteConfirmed	POST	int id: Identyfikator usuwanego typu.	Usuwa wskazany rodzaj pozycji z bazy danych i zapisuje zmiany.	Przekierowanie do Index.
ItemTypeExists	metoda pomocni cza	int id: Identyfikator sprawdzanego typu pozycji.	Sprawdza, czy obiekt o podanym id istnieje w bazie danych.	Wartość bool (true lub false).

- LoyaltiesController – zajmuje się zarządzaniem programem lojalnościowym

Metoda	HTTP	Parametry	Opis	Zwracane dane
Index	GET	brak	Pobiera dane programu lojalnościowego zalogowanego użytkownika oraz jego zatwierdzone transakcje. Oblicza punkty, dostępne nagrody oraz najczęściej kupowane produkty. Jeśli użytkownik nie jest zapisany w programie lojalnościowym, wyświetla domyślne wartości.	Widok z danymi programu lojalnościowego (ViewBag zawiera m.in. TotalPoints, CurrentStamps, FreeProductsAvaila ble, CurrentReward).
ConfirmRedempt ion	POST	Brak	Wyświetla widok potwierdzenia wykorzystania nagrody.	Widok ConfirmRedemption.
RedeemStamps	POST	brak	Wykorzystuje dostępne znaczkki lojalnościowe użytkownika w celu odebrania nagrody. Aktualizuje dane w programie lojalnościowym, generuje kod nagrody oraz ustawia timer ważności	Widok RedemptionCode (z wygenerowanym kodem i timerem).  Przekierowanie do Index w przypadku braku dostępnych

			kodu. Jeśli użytkownik nie ma wystarczających znaczków, przekierowuje do metody Index.	znaczków.
ConfirmUse	POST	Brak	Potwierdza wykorzystanie kodu nagrody.	Przekierowanie do metody Index.
LoyaltyExists	Brak (metoda pomocnicza).	int id: Identyfikator programu lojalnościowego.	Sprawdza, czy program lojalnościowy o podanym identyfikatorze istnieje w bazie danych.	Wartość bool (true lub false).

- MenuItemsController – zajmuje się zarządzaniem pozycjami w menu

Metoda	HTTP	Parametry	Opis	Zwracane dane
Create	GET	brak	Wyświetla formularz tworzenia nowego elementu menu. Dostępna tylko dla administratora.	Widok formularza tworzenia z listą typów elementów (ViewData["ItemType Id"]).
Create	POST	Obiekt MenuItem z właściwościami: Id, Name, Description, Price, ItemTypeId.	Tworzy nowy element menu na podstawie przesłanych danych. Waliduje dane i zapisuje je do bazy.	Przekierowanie do Index lub ponowny widok formularza w przypadku błędów.
Edit	GET	int? id: Opcjonalny identyfikator elementu menu.	Wyświetla formularz edycji istniejącego elementu menu. Dostępna tylko dla administratora.	Widok formularza edycji z listą typów elementów (ViewData["ItemType Id"]) lub NotFound, jeśli identyfikator jest nieprawidłowy.
Edit	POST	int id: Identyfikator elementu menu.  Obiekt MenuItem z właściwościami: Id, Name, Description, Price, ItemTypeId.	Aktualizuje dane elementu menu na podstawie przesłanych danych. Waliduje dane i zapisuje zmiany w bazie.	Przekierowanie do Index lub ponowny widok formularza w przypadku błędów.
Delete	GET	int? id: Opcjonalny identyfikator elementu menu.	Wyświetla widok potwierdzenia usunięcia elementu menu. Dostępna tylko dla administratora.	Widok z detalami elementu menu lub NotFound, jeśli identyfikator jest

				nieprawidłowy.
Delete	POST	int id: Identyfikator elementu menu.	Usuwa element menu na podstawie jego identyfikatora	Przekierowanie do Index.
MenuItemExists	Brak (metoda pomocnicza).	int id: Identyfikator elementu menu.	Sprawdza, czy element menu o podanym identyfikatorze istnieje w bazie danych.	Wartość bool (true lub false)

- TransactionController – zajmuje się zarządzaniem wizytami

Metoda	HTTP	Parametry	Opis	Zwracane dane
Index	GET	brak	Wyświetla listę wszystkich transakcji.  Administrator widzi wszystkie transakcje, użytkownicy standardowi tylko swoje.  Dodatkowo sprawdza, czy w systemie są użytkownicy niebędący administratorami.	Widok z listą transakcji i informacją w ViewBag.HasNonAdminUsers
Details	GET	int? id - ID transakcji.	Wyświetla szczegóły transakcji o podanym id.  Administrator widzi szczegóły każdej transakcji, użytkownicy standardowi tylko swoje.  Jeśli transakcja nie istnieje lub użytkownik nie ma do niej dostępu, zwraca błąd 404.	Widok z szczegółami transakcji.
Create	GET	brak	Wyświetla formularz do utworzenia nowej transakcji.  Administrator widzi listę użytkowników niebędących administratorami i wszystkie elementy menu.  Użytkownik standardowy	Widok formularza utworzenia transakcji z listą elementów menu w ViewData["MenuItems"].



			widzi tylko elementy menu.	
Create	POST	<p>Transaction transaction - Obiekt transakcji do zapisania.</p> <p>int[] quantities - Ilości wybranych pozycji z menu.</p> <p>string? selectedUserId - Opcjonalne ID użytkownika, używane przez administratora.</p>	<p>Waliduje dane wejściowe (np. unikalność kodu, datę zakupu, kwotę, ilości). Tworzy nową transakcję i zapisuje ją w bazie.</p> <p>Jeśli użytkownik jest administratorem, może przypisać transakcję innemu użytkownikowi i ustawić status "Approved".</p> <p>Jeśli to pierwsza transakcja użytkownika, tworzy powiązany obiekt lojalnościowy.</p> <p>Tworzy listę pozycji transakcji (TransactionItemsList).</p>	<p>Po sukcesie: Przekierowanie do akcji Index.</p> <p>W przypadku błędów: Widok formularza z komunikatami o błędach.</p>
Edit	GET	id (int?, opcjonalny) – identyfikator transakcji.	<p>Wyświetla widok do edycji transakcji. Pobiera szczegóły transakcji wraz z powiązanymi danymi (TransactionItemsList, MenuItem) na podstawie podanego id. Dodatkowo ładuje listę wszystkich dostępnych pozycji menu.</p> <p>Działa różnie w zależności od roli użytkownika (Administrator lub zwykły użytkownik).</p>	Widok z modelem transakcji i listą dostępnych pozycji menu.
Edit	POST	<p>id (int) – identyfikator transakcji.</p> <p>transaction (Transaction) – zaktualizowane dane transakcji.</p> <p>quantities (int[]) – ilości przedmiotów związanych z transakcją.</p>	<p>Aktualizuje istniejącą transakcję na podstawie danych przesłanych w formularzu. Weryfikuje poprawność kodu transakcji, daty zakupu, oraz innych pól. Dodaje nowe elementy do listy pozycji transakcji (TransactionItemsList).</p>	Przekierowanie do widoku Index w przypadku sukcesu lub ponowne wyświetlenie widoku edycji z komunikatami błędów.
Delete	GET	id (int?, opcjonalny) – identyfikator transakcji.	<p>Pobiera szczegóły transakcji do usunięcia, w tym powiązane pozycje (TransactionItemsList, MenuItem, Type). Działa różnie dla administratorów</p>	Widok potwierdzenia usunięcia z modelem transakcji.

			i zwykłych użytkowników, ograniczając widoczność danych.	
DeleteConfirmed	POST	id (int) – identyfikator transakcji.	Usuwa transakcję wraz z powiązanymi pozycjami z TransactionItemsList. Ogranicza dostęp do usuwania transakcji innych użytkowników (poza rolą Administrator).	Przekierowanie do widoku Index.
Approve	POST	id (int) – identyfikator transakcji.	Zatwierdza transakcję, zwiększając punkty lojalnościowe użytkownika (Loyalty.TotalPoints) o wartość transakcji. Ustawia pole Approved w transakcji na true.	Przekierowanie do widoku Index.
DissApprove	POST	id (int) – identyfikator transakcji.	Cofnięcie zatwierdzenia transakcji. Odejmuje punkty lojalnościowe użytkownika (Loyalty.TotalPoints) o wartość transakcji. Ustawia pole Approved w transakcji na false.	Przekierowanie do widoku Index.
TransactionExists	brak (metoda pomocnicza).	id (int) – identyfikator transakcji.	Sprawdza, czy transakcja istnieje w bazie danych.	Wartość logiczna (true/false).

### 3. Widoki:

- Publiczne (Administrator, użytkownicy zalogowani, goście):

Home/Index,About,Info,LoyaltyInfo

MenuItems/Index

Identity/Account/Register,Login

- Prywatne:

Wspólne (administrator, użytkownicy zalogowani):

Transactions/Index,Create,Details,Edit

Identity/Account/Manage/Index,ChangePassword

Tylko dla użytkowników:

Loyalties/Index,ConfirmRedemption,Redemption,Code

Tylko dla administratora:

MenuItems/Create,Delete,Details,Edit

---

## System użytkowników

- **Role w systemie:**
  - Administrator (z przypisaną rolą w bazie danych)
  - Użytkownik zwykły
- **Nadawanie ról:**

Użytkownik nie ma roli.  
Administrator jest jeden i została mu ta rola przydzielona ręcznie poprzez `AspNetUserRoles`.
- **Funkcje zależne od ról:**
  - Goście: Dostęp do stron publicznych
  - Zalogowani użytkownicy: Dostęp do dodatkowych funkcjonalności (wizyty, pieczątki)
  - Administratorzy: Dostęp do panelu zarządzania wizytami i menu.
- **Dane powiązane z użytkownikami:**
  - Konto użytkownika zawiera informacje o jego danych (mail, hasło).
  - Wizyty użytkowników są przypisywane do ich kont.

---

## Opis najciekawszych funkcjonalności:

Program liczy punkty zbierane przez użytkowników. Punkty naliczane są po zaakceptowaniu przez administratora. Program przelicza kwotę podaną przy dodawaniu wizyty na punkty, które mogą być wymienione na pieczątki. Użytkownik ma dostęp do wyświetlania ilości punktów, nagrody, ilości wykorzystanych pieczątek. Po uzbieraniu odpowiedniej ilości pieczątek, użytkownicy mogą ubiegać się o bonusy lojalnościowe w kawiarni. Bonusem jest najczęściej wybierana pozycja z menu przez danego użytkownika.

---