

Analiza tematu

Projekt: TechZone

Spis treści

1.	PIERWSZY ETAP PROJEKTOWANIA.....	3
1.1	Analiza wymagań.....	3
1.1.1	Identyfikacja wymagań i celów projektu.....	3
1.2	Analiza rynku i konkurencji.....	3
1.2.1	Analiza konkurencji	3
1.2.2	Badanie rynku (grupy docelowe)	4
1.3	Wymagania funkcjonalne	4
1.3.1	Wymagania podstawowe.....	5
1.3.2	Wymagania dodatkowe	5
1.3.3	Priorytetyzacja wymagań funkcjonalnych.....	6
1.3.4	Podsumowanie wymagań funkcjonalnych.....	6
1.4	Modelowanie procesów biznesowych.....	6
1.4.1	Rejestracja i logowanie użytkownika	7
1.4.2	Przeglądanie i wybór produktów	7
1.4.3	Proces zamówienia	7
1.4.4	Obsługa posprzedażowa	8
1.4.5	Proces zarządzania kontem użytkownika	8
1.4.6	Wizualizacja procesów (schemat BPMN)	8
1.4.7	Podsumowanie modelowania procesów biznesowych	8
1.5	Analiza ryzyka	8
1.5.1	Główne kategorie ryzyk w projekcie TechZone.....	9
1.5.2	Ocena ryzyk – skala zagrożeń	9
1.6	Plan zarządzania ryzykiem.....	9
1.6.1	Strategie zarządzania ryzykiem dla TechZone	9
1.6.2	Plan reakcji na ryzyko – matryca ryzyk	10
1.6.3	Monitorowanie ryzyk	11
2.	HARMONOGRAM.....	11
3.	FRONTEND	13
3.1	Prototypowanie UI i UX.....	13
3.1.1	Kluczowe zasady projektowania UI i UX dla TechZone.....	16
3.1.2	Główne ekrany i ich funkcjonalności	16
3.1.3	Przeprowadzanie testów użyteczności (UX)	17
3.1.4	Podsumowanie etapu prototypowania UI i UX	17
4.	BACKEND	18
4.1.1	Architektura.....	18
4.1.2	Warstwy aplikacji.....	18

4.1.3	Obsługa danych i komunikacja.....	18
4.1.4	Obsługa błędów.....	19
4.1.5	Bezpieczeństwo	19
4.1.6	Najważniejsze fragmenty kodu	20
4.1.7	Testy aplikacji	21
5.	Wnioski.....	22
5.1	Praca zespołowa.....	22
5.2	Podsumowanie pracy zespołowej	23

1. PIERWSZY ETAP PROJEKTOWANIA

1.1 Analiza wymagań

1.1.1 Identyfikacja wymagań i celów projektu

TechZone to nowoczesny sklep internetowy, którego celem jest umożliwienie łatwego i bezpiecznego zakupu sprzętu komputerowego, oferując dodatkowo funkcje, takie jak rejestracja, logowanie, aktualna lista produktów czy utworzenie zamówień. W związku z tym, główne wymagania i cele sklepu TechZone można rozłożyć na następujące punkty:

- **Intuicyjny i nowoczesny interfejs:** Dla zadowolenia użytkowników kluczowe jest stworzenie prostego i przyjaznego interfejsu, w którym użytkownicy będą mogli w łatwy sposób przeglądać produkty i dokonywać zakupu. Intuicyjność jest kluczowa zwłaszcza w przypadku nowoczesnych sklepów komputerowych, gdzie użytkownicy mogą mieć różny poziom wiedzy technicznej.
- **Bezpieczeństwo transakcji i danych użytkowników:** Sklep musi być zgodny z najlepszymi praktykami dotyczącymi ochrony danych użytkowników oraz zabezpieczeń transakcji online. Jest to kluczowe z punktu widzenia zarówno prawa, jak i zaufania klientów, którzy często podają dane osobowe i dane dotyczące płatności.
- **Łatwość przeglądania i nawigacji:** W sklepie z szeroką ofertą komponentów komputerowych i akcesoriów nawigacja produktów musi być bardzo dobrze przemyślana, aby użytkownik szybko znalazł interesujący go sprzęt.
- **Funkcja koszyka i historia zamówień:** Kluczowe elementy platformy e-commerce, które powinny działać intuicyjnie i bez problemów technicznych, oferując klientowi szybki podgląd zakupów i możliwość śledzenia historii swoich transakcji.

1.2 Analiza rynku i konkurencji

1.2.1 Analiza konkurencji

Wartość rynku e-commerce w zakresie sprzedaży sprzętu komputerowego stale rośnie, a TechZone jako sklep z nowoczesną technologią ma szansę wejść na rynek z potencjałem na wzrost. W tym celu należy jednak przeanalizować konkurencję oraz potrzeby klientów, aby zapewnić platformę wyróżniającą się funkcjonalnością i dostępnością. Na podstawie analizy konkurencyjnych sklepów internetowych, takich jak x-kom.pl, Morele.net i Komputronik.pl, można zidentyfikować mocne i słabe strony konkurentów oraz zdefiniować, jakie funkcje warto wprowadzić w TechZone.

Mocne strony konkurencji

- **Zaawansowane narzędzia filtrowania i wyszukiwania:** x-kom oraz Morele.net oferują zaawansowane filtry (według marki, specyfikacji, ceny, dostępności), które pozwalają szybko dotrzeć do poszukiwanych produktów.
- **Rozbudowane recenzje i oceny:** Umożliwiają użytkownikom wgląd w doświadczenia innych klientów, co podnosi poziom zaufania do sklepu i produktów.
- **Opcje rat i płatności odroczonych:** Funkcja popularna wśród klientów, szczególnie dla produktów o wyższej cenie (np. zestawy komputerowe i karty graficzne).
- **Promocje i rabaty:** Konkurencja przyciąga klientów licznymi promocjami i programami lojalnościowymi, co zwiększa atrakcyjność zakupów.

Słabe strony konkurencji

- **Niekonsekwencja w designie interfejsu użytkownika:** Wiele sklepów ma rozbudowane interfejsy, które mogą być trudne w nawigacji dla nowych użytkowników lub osób z mniejszym doświadczeniem technicznym.

- **Złożony proces zakupu:** Niektóre sklepy wymagają przejścia przez wiele kroków do realizacji zakupu, co może zniechęcać klientów.
- **Problemy z personalizacją:** Chociaż sklepy oferują szeroką gamę produktów, często brakuje im intuicyjnego narzędzia do składania zestawów komputerowych, co jest przewagą TechZone.

Wnioski i zalecenia

- TechZone powinien wyróżnić się prostym i przejrzystym interfejsem, który będzie zoptymalizowany pod kątem użyteczności zarówno na komputerach, jak i urządzeniach mobilnych.
- Należy rozważyć wdrożenie programu lojalnościowego lub systemu zniżek dla powracających klientów, aby zwiększyć ich zaangażowanie i przywiązanie do marki.
- Kluczowe będzie zapewnienie wygodnego i szybkiego procesu zakupu, z możliwością szybkiego logowania, finalizacji zamówienia oraz różnych opcji płatności.

1.2.2 Badanie rynku (grupy docelowe)

Persona 1: Gracz komputerowy

- **Charakterystyka:** Osoba młoda, zazwyczaj dobrze znająca nowinki technologiczne, zainteresowana gamingiem. Poszukuje sprzętu o wysokiej wydajności, jak karty graficzne, szybkie procesory i pamięci RAM.
- **Cele:** Ulepszanie swojego sprzętu do gier, możliwość dostosowania komponentów w zestawie gamingowym, oceny innych użytkowników sprzętu.
- **Funkcje kluczowe:** Personalizacja zestawów, recenzje produktów, szybki podgląd specyfikacji technicznej.

Persona 2: Entuzjasta IT i profesjonalista

- **Charakterystyka:** Dorosła osoba pracująca w branży technologicznej lub IT, zazwyczaj dobrze znająca specyfikacje techniczne i wymagająca sprzętu do zastosowań profesjonalnych, np. grafik komputerowych lub montażu wideo.
- **Cele:** Zakup sprzętu wysokiej jakości, który sprawdzi się przy dużym obciążeniu, takich jak procesory z wieloma rdzeniami czy szybkie dyski SSD.
- **Funkcje kluczowe:** Szczegółowe filtry wyszukiwania, zaawansowane opisy techniczne, system oceny jakości produktów.

Persona 3: Przeciętny użytkownik

- **Charakterystyka:** Osoba potrzebująca komputera lub akcesoriów do użytku codziennego, np. do pracy biurowej lub nauki, o ograniczonym budżecie i bez szczególnej znajomości technologii.
- **Cele:** Zakup prostego sprzętu w korzystnej cenie, możliwość łatwego przeglądania i szybkiego zakupu.
- **Funkcje kluczowe:** Intuicyjny interfejs, rekomendacje produktów na podstawie budżetu, wygodny koszyk i historia zamówień.

1.3 Wymagania funkcjonalne

Wymagania funkcjonalne TechZone można podzielić na dwie główne kategorie: **wymagania podstawowe** i **wymagania dodatkowe**. Wymagania podstawowe dotyczą kluczowych funkcji sklepu e-commerce, a dodatkowe mają na celu zwiększenie atrakcyjności i wygody korzystania z platformy.

1.3.1 Wymagania podstawowe

A. Przeglądanie i filtrowanie produktów

- **Wyszukiwanie produktów:** Użytkownicy muszą mieć możliwość wyszukiwania produktów za pomocą fraz (np. nazwa produktu, marka) oraz kodów produktów.
- **Filtrowanie i sortowanie:** Strona powinna umożliwiać zaawansowane filtrowanie produktów według kategorii, marki, ceny, popularności, parametrów technicznych (np. procesory, karty graficzne), dostępności oraz recenzji.
- **Sortowanie wyników:** Możliwość sortowania według cen (rosnąco i malejąco), najnowszych produktów, najlepiej ocenianych oraz najczęściej kupowanych.

B. Strona produktu

- **Opis produktu:** Każdy produkt powinien posiadać szczegółowy opis (parametry techniczne, marka, typ), zdjęcia oraz specyfikacje, co umożliwi użytkownikom świadomy wybór.
- **Cena oraz dostępność:** Wyświetlanie ceny, ewentualnych rabatów, oraz informacji o dostępności w magazynie.
- **Opcje dodania do koszyka i listy życzeń:** Każdy produkt powinien posiadać możliwość dodania do koszyka oraz listy życzeń, aby użytkownicy mogli wrócić do niego w przyszłości.
- **Rekomendacje produktów:** Wyświetlanie rekomendacji powiązanych produktów lub akcesoriów, co może zwiększyć sprzedaż oraz wygodę użytkowników.

C. Koszyk i finalizacja zakupu

- **Koszyk:** Koszyk powinien umożliwiać dodawanie, usuwanie i aktualizowanie liczby produktów oraz wyświetlać sumaryczny koszt zamówienia.
- **Szczegóły zamówienia:** Przed finalizacją zamówienia użytkownik powinien mieć możliwość sprawdzenia listy produktów oraz sumy zamówienia, w tym kosztów wysyłki i podatku.
- **Proces zakupu:** Intuicyjny proces realizacji zakupu, który poprowadzi użytkownika przez etapy: podanie adresu, wybór metody płatności, podsumowanie zamówienia i potwierdzenie.
- **Płatności:** Integracja z popularnymi metodami płatności (karta kredytowa, przelew bankowy, BLIK) oraz ewentualne opcje płatności ratowej.

D. Tworzenie i zarządzanie kontem użytkownika

- **Rejestracja i logowanie:** Możliwość rejestracji konta (email, hasło) oraz logowania przez e-mail lub zewnętrzne platformy (np. Google, Facebook).
- **Resetowanie hasła:** Opcja przypomnienia i resetowania hasła w przypadku jego utraty.
- **Panel użytkownika:** Każdy użytkownik po zalogowaniu powinien mieć dostęp do panelu, w którym zobaczy historię zamówień, szczegóły konta (adres, metoda płatności) oraz ustawienia konta.
- **Historia zamówień:** Możliwość przeglądania historii zakupów i podglądu szczegółów każdego zamówienia (data, produkty, koszt, status).

1.3.2 Wymagania dodatkowe

A. Lista życzeń

- **Tworzenie listy życzeń:** Użytkownicy powinni mieć możliwość dodawania produktów do listy życzeń, aby móc wrócić do nich w przyszłości.
- **Udostępnianie listy:** Możliwość udostępnienia swojej listy życzeń (np. w formie linku) znajomym lub rodzinie.

B. System powiadomień

- **Powiadomienia o stanie zamówienia:** Automatyczne powiadomienia mailowe lub SMS informujące o aktualnym statusie zamówienia (złożone, przetwarzane, wysłane, dostarczone).
- **Powiadomienia o produktach:** Powiadomienia o dostępności wybranych produktów (np. gdy produkt jest chwilowo niedostępny) lub o promocjach na produkty z listy życzeń.

C. System zarządzania promocjami i rabatami

- **Kody rabatowe:** Możliwość generowania i stosowania kodów rabatowych przez użytkowników.
- **Program lojalnościowy:** Opcja wprowadzenia systemu punktowego lub zniżek dla powracających klientów, co pozwala zwiększyć zaangażowanie i przywiązanie użytkowników do sklepu.

1.3.3 Priorytetyzacja wymagań funkcjonalnych

Podział funkcji na priorytety pomoże zespołowi skoncentrować się na kluczowych funkcjach w pierwszej fazie projektu, a także pozwoli dostosować harmonogram prac:

- **Wysoki priorytet (Must-have):**
 - Rejestracja/logowanie i zarządzanie kontem użytkownika.
 - Koszyk i finalizacja zakupu.
 - Podstawowe wyszukiwanie, filtrowanie i sortowanie produktów.
 - Strona produktu (opis, zdjęcia, cena, dostępność).
 - Historia zamówień i szczegóły zamówienia.
- **Średni priorytet (Should-have):**
 - Lista życzeń.
 - Powiadomienia o stanie zamówienia.
 - Personalizacja zestawów komputerowych z podstawowym konfiguracją kompatybilności.
- **Niski priorytet (Nice-to-have):**
 - Program lojalnościowy i system promocji.
 - System powiadomień o dostępności produktów.

1.3.4 Podsumowanie wymagań funkcjonalnych

Na podstawie opisanych wymagań funkcjonalnych TechZone będzie oferował platformę e-commerce z szeroką gamą funkcji, która wyróżni się możliwością personalizacji zestawów komputerowych oraz bogatymi opcjami oceny produktów przez użytkowników. Funkcje te zapewnią TechZone nie tylko pozycję atrakcyjnego sklepu z nowoczesną technologią, ale również wysoką użyteczność i bezpieczeństwo platformy, co może przełożyć się na większe zaangażowanie klientów i przewagę na rynku.

1.4 Modelowanie procesów biznesowych

Poniżej opisane zostaną główne procesy biznesowe w TechZone z zastosowaniem standardu BPMN (Business Process Model and Notation), który pozwala na przejrzyste przedstawienie procesów i przepływu działań między różnymi uczestnikami. Skupimy się na kluczowych procesach takich jak rejestracja i logowanie użytkownika, przeglądanie i wybór produktów, personalizacja zestawów, proces zamówienia oraz obsługa posprzedażowa.

1.4.1 Rejestracja i logowanie użytkownika

1. **Wejście użytkownika na stronę TechZone:**
 - Użytkownik ma możliwość przeglądania produktów bez logowania lub może zdecydować się na rejestrację/zalogowanie.
2. **Proces rejestracji:**
 - Użytkownik wybiera opcję rejestracji, wprowadza dane (email, hasło), akceptuje regulamin i potwierdza.
 - System sprawdza poprawność danych i tworzy konto w bazie użytkowników.
 - Po zakończeniu rejestracji, użytkownik otrzymuje mail z potwierdzeniem założenia konta.
3. **Proces logowania:**
 - Użytkownik wpisuje login i hasło (lub loguje się przez zewnętrzne platformy, np. Google).
 - System weryfikuje dane logowania i umożliwia dostęp do panelu użytkownika, gdzie klient może zarządzać kontem, zamówieniami oraz listą życzeń.

1.4.2 Przeglądanie i wybór produktów

1. **Przeglądanie kategorii:**
 - Użytkownik wybiera interesującą go kategorię (np. laptopy, procesory) lub wyszukuje produkt według frazy.
2. **Filtrowanie i sortowanie:**
 - Użytkownik stosuje filtry (np. cena, marka, parametry techniczne) i sortuje produkty według wybranego kryterium, np. ceny lub popularności.
3. **Strona produktu:**
 - Po kliknięciu na produkt użytkownik przechodzi do szczegółowego widoku produktu, gdzie ma dostęp do opisu, specyfikacji technicznej, ocen i recenzji.
 - Może dodać produkt do koszyka, listy życzeń lub wybrać opcję personalizacji (jeśli jest dostępna).

1.4.3 Proces zamówienia

1. **Dodanie produktów do koszyka:**
 - Użytkownik dodaje wybrane produkty lub skonfigurowany zestaw do koszyka.
2. **Przegląd koszyka:**
 - Użytkownik przegląda zawartość koszyka, ma możliwość edytowania liczby produktów, usunięcia wybranych pozycji oraz przeglądu kosztów (cena produktów, koszt wysyłki, rabaty).
3. **Podanie danych do wysyłki i wybór metody płatności:**
 - Użytkownik podaje dane do wysyłki i wybiera metodę płatności (np. przelew, karta kredytowa, płatność ratalna).
 - Możliwość wprowadzenia kodu rabatowego.
4. **Potwierdzenie zamówienia:**
 - Użytkownik dokonuje ostatecznego potwierdzenia zamówienia, po czym następuje przekierowanie do wybranego operatora płatności lub bezpośrednia finalizacja (np. BLIK).
 - Po potwierdzeniu płatności system generuje zamówienie i wysyła potwierdzenie na e-mail użytkownika.
5. **Realizacja zamówienia przez dział logistyki:**
 - Dział logistyki przetwarza zamówienie, sprawdza dostępność produktów i przygotowuje przesyłkę.

- Po wysłaniu zamówienia system generuje automatyczne powiadomienie z numerem śledzenia przesyłki.

1.4.4 Obsługa posprzedażowa

1. **Historia zamówień:**
 - Użytkownik w panelu konta ma dostęp do historii zamówień, gdzie widzi szczegóły wszystkich transakcji, statusy zamówień oraz możliwość śledzenia przesyłek.
2. **Obsługa zwrotów i reklamacji:**
 - Użytkownik może zgłosić zwrot lub reklamację poprzez panel użytkownika, wybierając odpowiednią opcję przy zamówieniu.
 - System umożliwia automatyczne generowanie formularza zwrotu, a dział obsługi klienta zarządza reklamacjami, aktualizując użytkownika o statusie zgłoszenia.

1.4.5 Proces zarządzania kontem użytkownika

1. **Aktualizacja danych konta:**
 - Użytkownik może aktualizować swoje dane, takie jak adres dostawy, hasło, metody płatności i preferencje dotyczące powiadomień.
2. **Zarządzanie listą życzeń:**
 - Użytkownik może dodawać lub usuwać produkty z listy życzeń, którą może także udostępnić.
3. **Powiadomienia i subskrypcje:**
 - Użytkownik może zarządzać powiadomieniami (np. o stanie zamówienia, promocjach), zapisując się lub rezygnując z subskrypcji newslettera lub powiadomień SMS.

1.4.6 Wizualizacja procesów (schemat BPMN)

Dla pełnej wizualizacji procesów warto stworzyć schematy BPMN dla poszczególnych procesów, które obrazują przepływ działań i decyzji w każdym z procesów. Przykładowe schematy BPMN, które można wykonać, to:

- **Proces logowania i rejestracji użytkownika.**
- **Proces wyboru i personalizacji zestawów komputerowych.**
- **Proces zakupowy (dodanie do koszyka, zamówienie i płatność).**
- **Proces obsługi posprzedażowej (zwroty, oceny produktów, zarządzanie kontem).**

1.4.7 Podsumowanie modelowania procesów biznesowych

Procesy biznesowe w TechZone są zaprojektowane w sposób zapewniający płynne i intuicyjne doświadczenie użytkownika, a ich dokładne modelowanie pomoże zespołowi zrozumieć każdy etap interakcji klienta ze sklepem. Procesy te obejmują nie tylko przeglądanie i zakup produktów, ale także ich personalizację, ocenę oraz kompleksową obsługę posprzedażową, co przyczyni się do wysokiego poziomu zadowolenia klientów oraz efektywności operacyjnej platformy TechZone.

1.5 Analiza ryzyka

Analiza ryzyka polega na identyfikacji, ocenie oraz priorytetyzacji zagrożeń, które mogą wpłynąć na realizację projektu oraz na jego funkcjonowanie po wdrożeniu. Dla TechZone zidentyfikowano najważniejsze rodzaje ryzyk, które mogą negatywnie wpłynąć na użytkowników oraz funkcjonowanie sklepu.

1.5.1 Główne kategorie ryzyk w projekcie TechZone

1. **Ryzyka technologiczne**
 - **Niestabilność systemu:** Możliwość awarii serwera lub błędów systemowych może prowadzić do niedostępności strony, co zniechęca użytkowników i prowadzi do utraty sprzedaży.
 - **Błędy w oprogramowaniu:** Nieprawidłowe działanie funkcji sklepu (np. koszyka, płatności, personalizacji zestawów) może negatywnie wpływać na zadowolenie klientów i wizerunek marki.
2. **Ryzyka bezpieczeństwa**
 - **Ataki cybernetyczne:** Ryzyko włamań, ataków typu DDoS czy kradzieży danych użytkowników.
 - **Utrata danych:** Przypadkowa utrata lub wyciek danych (np. danych użytkowników, historii zamówień) mogą naruszyć przepisy RODO oraz wpłynąć na zaufanie klientów.
3. **Ryzyka operacyjne**
 - **Problemy logistyczne:** Błędy lub opóźnienia w realizacji zamówień, wynikające z problemów po stronie dostawców lub w magazynach.
 - **Brak dostępności produktów:** Niedostateczna liczba produktów na stanie lub nieaktualne informacje o dostępności mogą prowadzić do niezadowolenia klientów.
 - **Błędy w procesie zwrotów:** Problemy w systemie obsługi zwrotów lub reklamacji mogą skutkować negatywnymi opiniami klientów.
4. **Ryzyka związane z doświadczeniem użytkownika**
 - **Niska intuicyjność interfejsu:** Złożony lub mało intuicyjny interfejs może prowadzić do rezygnacji użytkowników i niskiej konwersji.
 - **Wydłużony czas ładowania strony:** Wolne działanie strony, zwłaszcza w trakcie promocji, może wpłynąć na frustrację klientów i zmniejszyć sprzedaż.
5. **Ryzyka finansowe**
 - **Niedoszacowanie kosztów:** Ryzyko, że budżet okaże się niewystarczający na realizację i utrzymanie platformy.
 - **Problemy z płynnością finansową:** Niezaplanowane koszty lub opóźnienia w płatnościach (np. od partnerów) mogą wpłynąć na budżet.

1.5.2 Ocena ryzyk – skala zagrożeń

Każde ryzyko może być ocenione pod kątem:

- **Prawdopodobieństwa** (niskie, średnie, wysokie)
- **Skutków** (niski, średni, wysoki wpływ na projekt)

1.6 Plan zarządzania ryzykiem

Plan zarządzania ryzykiem to zestaw działań mających na celu minimalizację oraz efektywne reagowanie na zidentyfikowane ryzyka. Obejmuje zarówno działania prewencyjne, jak i konkretne plany na wypadek ich wystąpienia.

1.6.1 Strategie zarządzania ryzykiem dla TechZone

1. **Ryzyka technologiczne**
 - **Monitoring i backup systemu:** Wdrożenie monitoringu serwerów oraz regularnych kopii zapasowych minimalizuje ryzyko utraty danych i umożliwia szybkie przywrócenie systemu w przypadku awarii.

- **Testowanie i wersjonowanie oprogramowania:** Regularne testy automatyczne i manualne pozwalają na szybkie wykrycie błędów, a system wersjonowania umożliwia powrót do stabilnych wersji w przypadku awarii.
- 2. **Ryzyka bezpieczeństwa**
 - **Zabezpieczenia antywłamaniowe i szyfrowanie danych:** Wdrożenie technologii szyfrujących (np. SSL) oraz narzędzi do ochrony przed atakami, takich jak firewalle czy systemy anty-DDoS.
 - **Szkolenia z bezpieczeństwa i polityka RODO:** Szkolenie zespołu na temat bezpieczeństwa oraz ścisłe przestrzeganie polityki RODO w zakresie przetwarzania i przechowywania danych.
- 3. **Ryzyka operacyjne**
 - **Zarządzanie magazynem i integracja z systemami dostawców:** Integracja systemów ERP lub magazynowych z systemami dostawców pozwala na aktualizację stanów magazynowych oraz przyspieszenie procesów zamówień.
 - **Automatyzacja obsługi klienta i zwrotów:** Wdrożenie systemu zarządzania zgłoszeniami oraz automatyczne generowanie formularzy zwrotów skracają czas obsługi reklamacji i minimalizują błędy.
- 4. **Ryzyka związane z doświadczeniem użytkownika**
 - **Testy UX i optymalizacja strony:** Przeprowadzanie regularnych testów użyteczności oraz optymalizacja wydajności (np. CDN, optymalizacja grafik) pozwalają na poprawę jakości doświadczenia użytkowników i skrócenie czasu ładowania strony.
 - **Zbieranie opinii użytkowników:** Systematyczne zbieranie informacji zwrotnych od klientów oraz analiza ich preferencji pozwala na bieżącą poprawę interfejsu.
- 5. **Ryzyka finansowe**
 - **Rezerwa budżetowa:** Zaplanowanie rezerwy finansowej na nieprzewidziane wydatki lub kosztowne poprawki po fazie testowania.
 - **Planowanie budżetu i harmonogramu:** Opracowanie szczegółowego budżetu i harmonogramu projektu, w którym uwzględnione będą wszystkie planowane wydatki na rozwój, testy i utrzymanie platformy.

1.6.2 Plan reakcji na ryzyko – matryca ryzyk

1. **Zagrożenie:** Awaria serwera / przestój
 - **Reakcja:** Natychmiastowe przełączenie na zapasowy serwer i przywrócenie danych z kopii zapasowej.
 - **Priorytet:** Wysoki.
2. **Zagrożenie:** Atak hakerski / kradzież danych
 - **Reakcja:** Zablokowanie dostępu i identyfikacja źródła zagrożenia, poinformowanie użytkowników oraz wzmocnienie zabezpieczeń.
 - **Priorytet:** Wysoki.
3. **Zagrożenie:** Problemy z realizacją zamówień
 - **Reakcja:** Wdrożenie alternatywnych dostawców lub uruchomienie dodatkowych zasobów magazynowych.
 - **Priorytet:** Średni.
4. **Zagrożenie:** Wolne działanie strony
 - **Reakcja:** Optymalizacja wydajności poprzez caching i zmniejszenie liczby zewnętrznych zasobów.
 - **Priorytet:** Średni.
5. **Zagrożenie:** Niedobór budżetu na kluczowym etapie projektu
 - **Reakcja:** Przeanalizowanie priorytetów projektowych oraz pozyskanie dodatkowego finansowania (np. od inwestora lub banku).
 - **Priorytet:** Wysoki.

1.6.3 Monitorowanie ryzyk

Aby zminimalizować wpływ ryzyk, należy wdrożyć regularny proces monitorowania:

- **Regularne przeglądy systemów:** Monitoring systemów serwerowych oraz okresowe testy penetracyjne w celu wykrycia potencjalnych luk w zabezpieczeniach.
- **Analiza wskaźników KPI:** Śledzenie wskaźników wydajności, takich jak czas ładowania strony, liczba zgłoszeń klientów oraz poziom dostępności produktów.
- **Przeglądy budżetowe:** Regularna analiza stanu budżetu i kosztów utrzymania platformy.

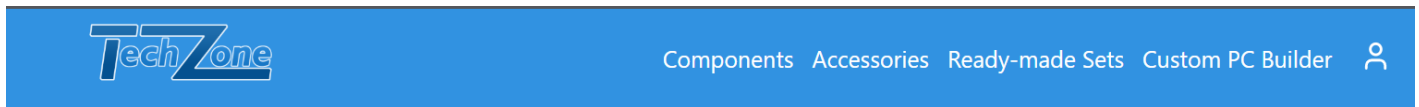
2. HARMONOGRAM

Data	Zjazd	Zakres Prac	Zadania do realizacji	Status
13.10.24	1	Zarządzanie projektem	- Spotkanie inicjujące projekt - Ustalenie zakresu projektu i podział ról w zespole.	😊
		Analityka biznesowa	- Przeprowadzenie analizy wstępnej i badania rynku.	😊
20.10.24	2	Analityka biznesowa	- Zbieranie wymagań od interesariuszy. - Określenie wymagań funkcjonalnych i niefunkcjonalnych.	😊
27.10.24	3	Analityka biznesowa	- Przygotowanie specyfikacji funkcjonalnej.	😊
		UI/UX Design	- Tworzenie prototypów.	😊
3.11.24		UI/UX Design	- Tworzenie szkiców poszczególnych elementów strony.	😊
10.11.24		Analityka biznesowa	- Konsultacje i weryfikacja wymagań.	😊
17.11.24	4	Frontend Development	- Tworzenie strony głównej i okien popup	😊
		Zarządzanie projektem	- Spotkanie określające co zostało zrobione i ustalenie kolejnych zadań i priorytetów	😊
24.11.24	5	Backend Development	- Rozpoczęcie implementacji backendu (np. baza danych, logowanie użytkownika).	😊
01.12.24		Backend Development	- Kontynuacja implementacji poszczególnych funkcji takich jak koszyk.	
		Frontend Development	- Utworzenie kolejnych podstron takich jak koszyk	😊
07.12.24	6	Frontend Development	- Utworzenie ekranów logowania i rejestracji	😊
		Frontend Development	- Implementacja frontendowych funkcji podstawowych (strona główna, kategorie produktów).	😊
15.12.24	7	Backend Development	- Kontynuacja implementacji: <ul style="list-style-type: none">• listy produktów	😊

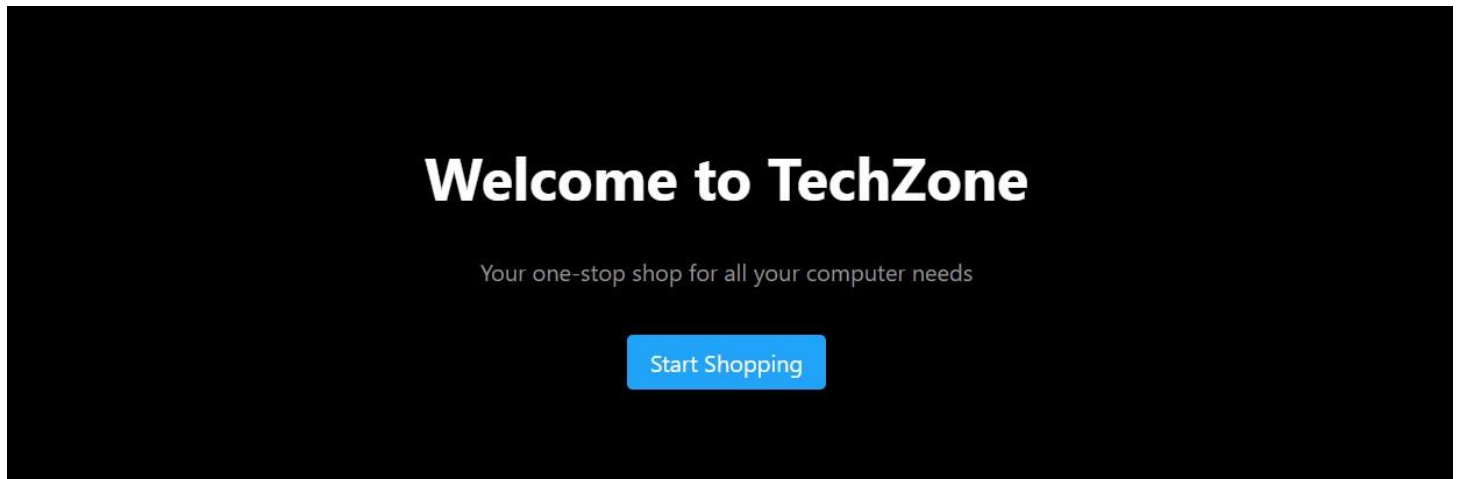
			<ul style="list-style-type: none">• dodawanie do koszyka• utworzenie zamówień• szczegółowe informacje o produktach	
		Frontend Development	- Kontynuacja tworzenia podstron takich jak profil użytkownika	😊
PRZERWA ŚWIĄTECZNA				
12.01.25	8	QA (Quality Assurance)	- Testowanie wstępnych funkcji platformy.	😊
		QA / Backend/Frontend Development	- Wprowadzanie poprawek po testach.	😊
		Backend/Frontend Development	- Integracja backendu z frontendem (np. zarządzanie użytkownikami, historia zamówień).	😊
19.01.25	9	Backend/Frontend Development / DevOps	- Optymalizacja działania platformy.	😊
		QA (Quality Assurance)	- Ostateczne testy funkcjonalne.	😊
		QA / UI/UX Design	- Przeprowadzenie testów UX z udziałem użytkowników.	😊
26.01.25	10	Analityka biznesowa	- Analiza ryzyka i przygotowanie dokumentacji dotyczącej zarządzania ryzykiem.	😊
		QA / Backend/Frontend Development	- Wprowadzenie ostatnich poprawek w oparciu o feedback z testów.	😊
		Zarządzanie projektem	- Przygotowanie do prezentacji projektu. - Prezentacja gotowego projektu na zjeździe.	😊

3. FRONTEND

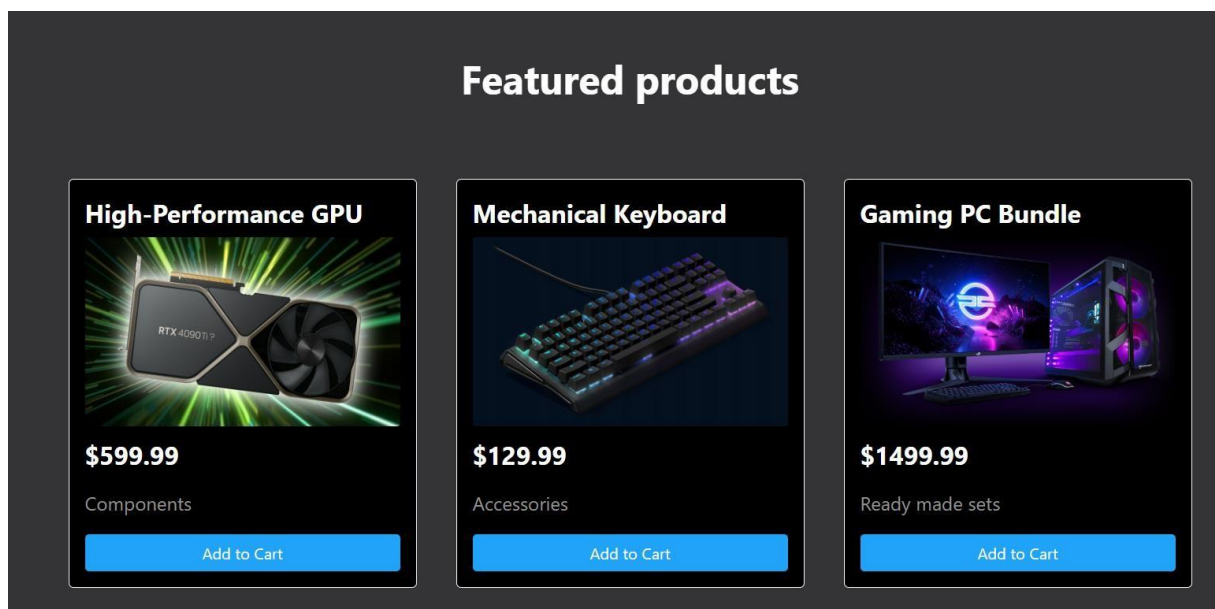
3.1 Prototypowanie UI i UX



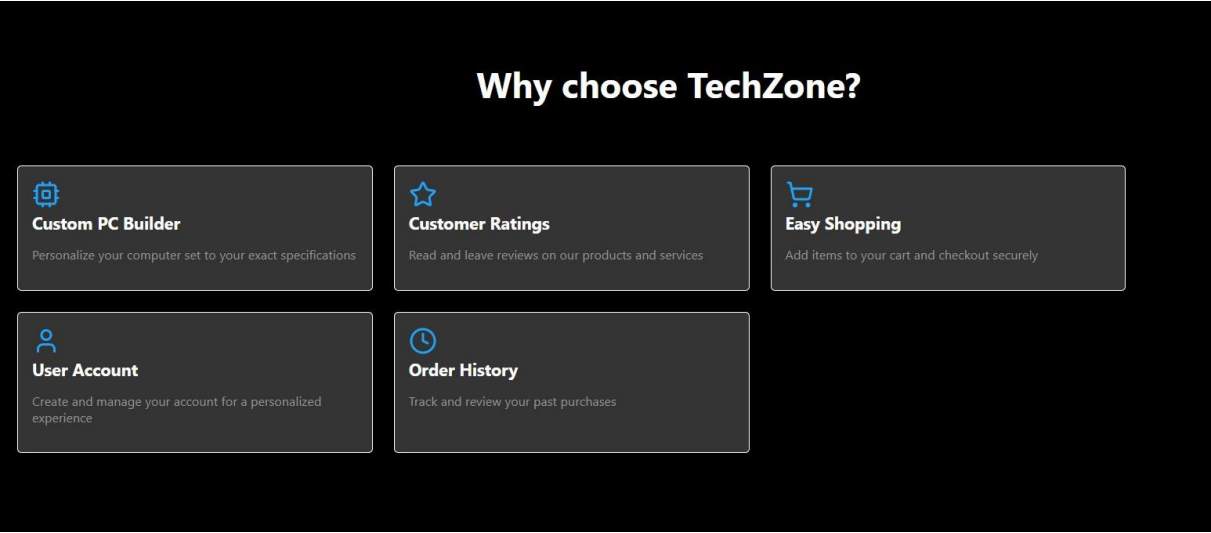
Rys. 1 Nagłówek strony z logo oraz sekcją podstron, a także ikoną ludzika, służącego do zarządzania kontem użytkownika.



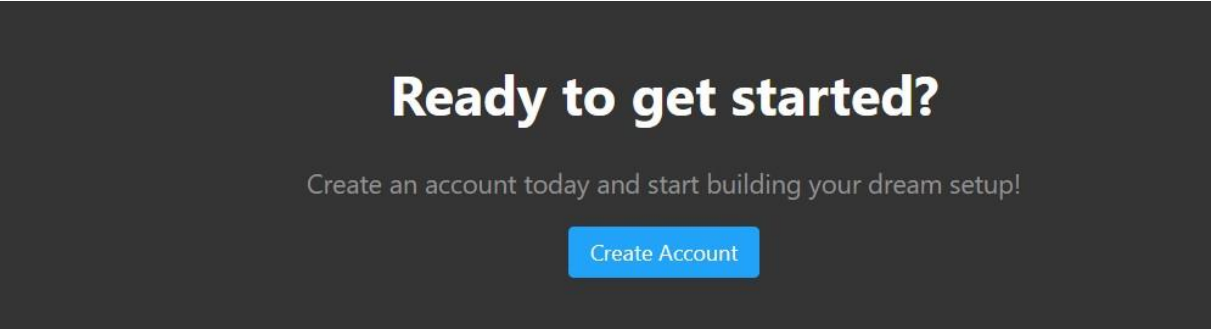
Rys. 2. Blok witający użytkownika i zachęcający go do rozpoczęcia zakupów



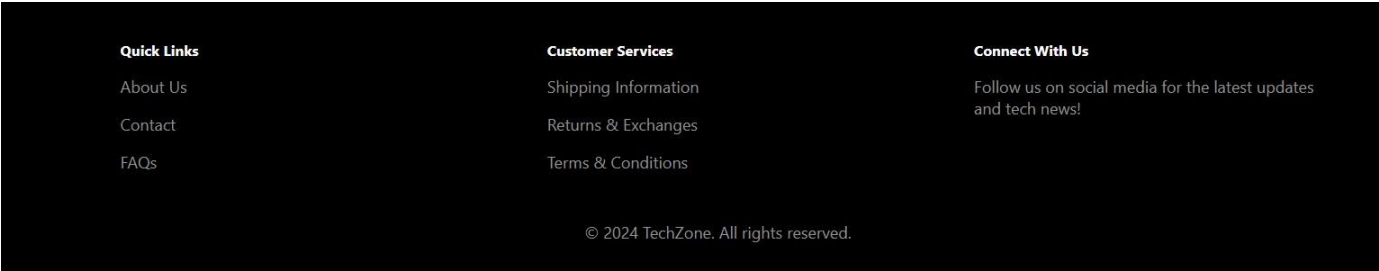
Rys. 3. Panel z polecanymi produktami



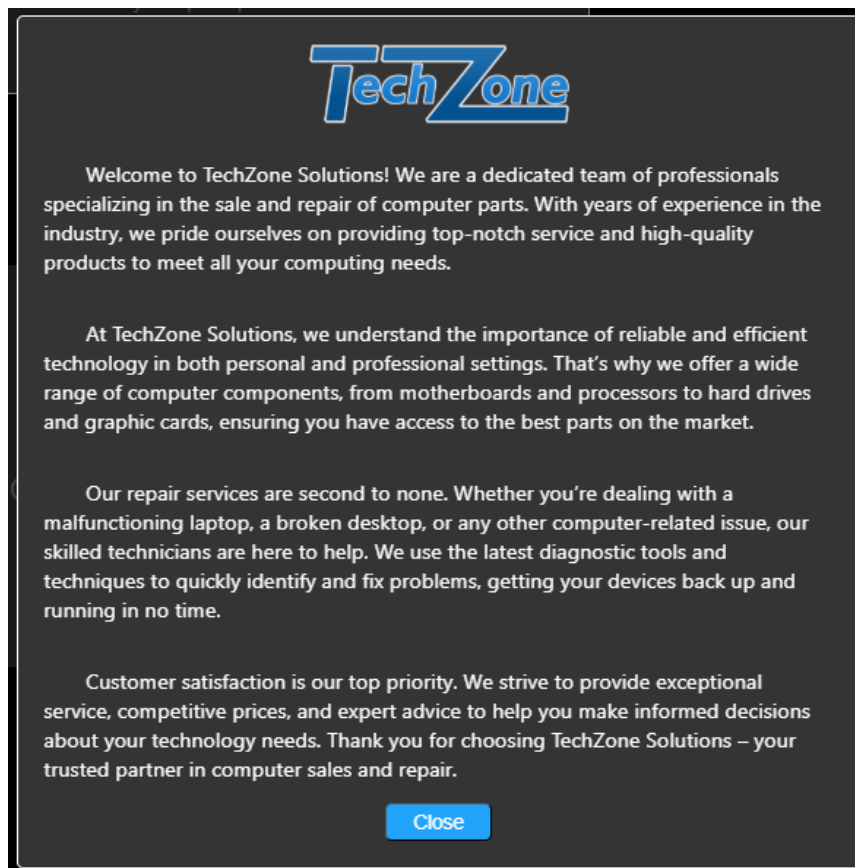
Rys. 4. Panel przekonujący użytkownika dlaczego powinien wybrać nasze usługi.



Rys. 5. Panel zachęcający użytkownika do założenia konta na stronie



Rys. 6. Panel z linkami do podstawowych informacji o sklepie



Rys. 7. Okno modalne „About us”

Create Your Account

Name

Email

Password

Provide password again

Phone Number

Create Account

Already have an account? [Sign in!](#)

Rys 8. Okno rejestracji

Log In to Your Account

Email

Password

Log In!

Don't have an account yet? [Create one!](#)

Rys 9. Okno logowania

3.1.1 Kluczowe zasady projektowania UI i UX dla TechZone

1. **Intuicyjność interfejsu:** Prosty układ i jasne oznaczenia mają zapewnić, że użytkownicy szybko zrozumieją, jak korzystać ze strony.
2. **Szybki dostęp do informacji:** Główne kategorie produktów, wyszukiwarka oraz koszyk powinny być łatwo dostępne z każdej strony.
3. **Responsywność:** Interfejs musi być dostosowany do urządzeń mobilnych, tabletów i komputerów, aby użytkownicy mogli swobodnie korzystać ze sklepu na każdym urządzeniu.
4. **Spójna kolorystyka i elementy graficzne:** Kolory, typografia i ikony powinny być spójne z nowoczesnym, technologicznym charakterem sklepu.

3.1.2 Główne ekrany i ich funkcjonalności

A. Strona główna

- **Nagłówek:** Umieszczenie logo, wyszukiwarki produktów, koszyka i ikony konta użytkownika.
- **Główna nawigacja:** Lista kategorii (np. Laptopy, Komponenty, Akcesoria), co pozwala użytkownikom szybko przejść do interesującej ich sekcji.
- **Slider promocyjny:** Obszar z grafikami promującymi nowości, oferty specjalne lub popularne produkty.
- **Polecane produkty:** Sekcja z rekomendacjami oraz najlepiej sprzedającymi się produktami w różnych kategoriach.

B. Strona kategorii produktów

- **Opcje filtrowania i sortowania:** Umożliwienie filtrowania wyników według kategorii, ceny, popularności i specyfikacji (np. RAM, procesor).
- **Kafelkowy widok produktów:** Zdjęcie produktu, cena, krótki opis, przycisk „Dodaj do koszyka” i możliwość przejścia do szczegółowego widoku produktu.
- **Pasek nawigacyjny:** W każdej chwili użytkownik może przejść do innej kategorii lub wrócić na stronę główną.

C. Strona produktu

- **Zdjęcia i opis:** Wysokiej jakości zdjęcia, szczegółowy opis produktu, specyfikacje techniczne oraz ceny.
- **Opcje zakupu:** Przyciski „Dodaj do koszyka” i „Dodaj do listy życzeń”.
- **Rekomendacje produktów:** Produkty powiązane, takie jak akcesoria czy alternatywne modele.

D. Koszyk

- **Podgląd produktów:** Lista produktów w koszyku, ich ilość, cena jednostkowa oraz opcja usunięcia lub zmiany liczby sztuk.
- **Podsumowanie zamówienia:** Całkowity koszt zamówienia z opcją przejścia do finalizacji zakupów.
- **Pole kodu rabatowego:** Możliwość wpisania kodu promocyjnego i natychmiastowe naliczenie rabatu.
- **Opcje dostawy i koszt wysyłki:** Użytkownik może wybrać sposób dostawy, a koszty są wyświetlane w czasie rzeczywistym.

E. Strona finalizacji zamówienia

- **Formularz danych adresowych:** Pola na dane kontaktowe i adres dostawy.
- **Wybór metody płatności:** Integracja z płatnościami kartą, przelewem lub płatnościami online (np. BLIK).
- **Potwierdzenie zamówienia:** Podsumowanie zakupów i kosztów przed ostatecznym złożeniem zamówienia.

F. Panel użytkownika

- **Informacje o koncie:** Opcje zarządzania danymi użytkownika (adresy, płatności, hasło).
- **Historia zamówień:** Lista wcześniejszych zakupów z możliwością ponownego zamówienia.
- **Lista życzeń:** Przegląd zapisanych produktów, które użytkownik chce kupić w przyszłości.
- **Opcje subskrypcji i powiadomień:** Zarządzanie preferencjami, takimi jak zgoda na newsletter.

3.1.3 Przeprowadzanie testów użyteczności (UX)

Oto przykłady testów, które po utworzeniu głównych funkcjonalności planujemy wykonać po stronie backendu.

1. **Testy A/B:**
 - Przeprowadzenie testów A/B pozwoli zrozumieć, które warianty interfejsu są bardziej intuicyjne. Na przykład można przetestować dwie wersje układu koszyka lub różne kolory przycisków dodawania do koszyka.
2. **Testy z użytkownikami:**
 - Można zaprosić potencjalnych użytkowników do przetestowania prototypu i poprosić o feedback dotyczący wygody użytkowania, czytelności oraz szybkości znajdowania produktów.
 - Testy pozwalają na identyfikację problemów, takich jak niezrozumiałe ikony czy zbyt długi proces rejestracji.
3. **Analiza ścieżki użytkownika:**
 - W trakcie testów ścieżek użytkowników (np. od strony głównej po finalizację zakupu) można sprawdzić, czy nawigacja na stronie jest płynna i czy nie pojawiają się żadne błędy lub nieporozumienia.

3.1.4 Podsumowanie etapu prototypowania UI i UX

W procesie prototypowania UI i UX skupiono się na stworzeniu przejrzystego, nowoczesnego i łatwego w obsłudze interfejsu, który zapewni użytkownikom przyjemne doświadczenia podczas korzystania ze sklepu TechZone. Przemyślany interfejs z przejrzystą nawigacją, wyraźnymi przyciskami akcji oraz intuicyjnymi funkcjonalnościami (np. personalizacja zestawów) sprawi, że użytkownicy będą mogli łatwo znaleźć interesujące ich produkty i szybko sfinalizować zakup.

Testy użyteczności pozwolą dodatkowo zoptymalizować interfejs i zapewnią, że TechZone spełni wysokie standardy w zakresie doświadczenia użytkownika, co przyczyni się do wzrostu lojalności klientów i wyróżnienia sklepu na tle konkurencji.

4. BACKEND

4.1.1 Architektura

Backend sklepu internetowego został zaprojektowany z wykorzystaniem frameworka ASP.NET Core. Architektura aplikacji została zbudowana w sposób modułowy i zgodnie z zasadami czystej architektury, co zapewnia skalowalność, czytelność kodu oraz łatwość wprowadzania zmian.

4.1.2 Warstwy aplikacji

Aplikacja została podzielona na następujące warstwy:

1. Warstwa API

- Odpowiada za obsługę żądań HTTP i zwracanie odpowiedzi w formacie JSON.
- Zawiera kontrolery, które obsługują różne zasoby, np. produkty, użytkowników, zamówienia.
- Przykładowe endpointy:
 - GET /api/products/list - pobieranie listy produktów.
 - POST /api/cart /add - tworzenie zamówienia.

2. Warstwa usług (Service Layer)

- Odpowiada za logikę biznesową aplikacji.
- Zawiera klasy odpowiedzialne za operacje na danych, np. obsługa zamówień czy przeliczanie wartości koszyka.
- Przykłady metod:
 - GenerateJwtToken (): Generuje token JWT
 - HashPassword (): Haszuje hasło.

3. Warstwa dostępu do danych (Data Access Layer)

- Implementacja logiki dostępu do bazy danych z wykorzystaniem Entity Framework Core.
- Odpowiedzialna za zapisywanie, odczytywanie i aktualizowanie danych w bazie.
- Wzorzec Repository Pattern został zastosowany do oddzielenia logiki bazy danych od logiki biznesowej.

4.1.3 Obsługa danych i komunikacja

Do obsługi danych w backendzie sklepu internetowego wykorzystano Entity Framework Core jako ORM, który umożliwia łatwą i wydajną komunikację z relacyjną bazą danych. Baza danych została zaprojektowana i wdrożona w Microsoft SQL Server (MSSQL), co zapewniło wysoką wydajność, skalowalność oraz zgodność z wymaganiami projektu. Struktura bazy danych została stworzona zgodnie z zasadami normalizacji, co pozwoliło zminimalizować redundancję danych oraz zoptymalizować ich przechowywanie. Do zarządzania zmianami w schemacie bazy danych wykorzystano narzędzie EF Core Migrations, które pozwala na kontrolowane i bezpieczne wdrażanie

modyfikacji w środowisku produkcyjnym. W razie potrzeby baza danych mogła zostać rozszerzona o procedury składowane w celu optymalizacji operacji na dużych zestawach danych.

4.1.4 Obsługa błędów

W backendzie sklepu internetowego zastosowano kompleksowy mechanizm obsługi błędów, który zapewnia zarówno przejrzystość dla użytkowników, jak i możliwość diagnozowania problemów przez programistów. Globalna obsługa błędów została zaimplementowana przy użyciu middleware w ASP.NET Core, co pozwala na przechwytywanie wszystkich wyjątków występujących w aplikacji.

Błędy walidacji danych wejściowych są automatycznie zwracane z odpowiednim statusem HTTP, takim jak 400 Bad Request, wraz z jasnym komunikatem wyjaśniającym problem. W przypadku niespodziewanych błędów system zwraca status 500 Internal Server Error.

4.1.5 Bezpieczeństwo

Backend sklepu został zaprojektowany z uwzględnieniem wysokich standardów bezpieczeństwa, aby chronić dane użytkowników i zapobiegać potencjalnym atakom. W celu uwierzytelniania użytkowników wykorzystano tokeny JWT (JSON Web Token), które umożliwiają bezpieczne zarządzanie sesjami użytkowników. Endpointy API wymagające uwierzytelnienia zostały zabezpieczone za pomocą atrybutu [Authorize], co uniemożliwia dostęp nieautoryzowanym użytkownikom.

Hasła użytkowników są przechowywane w bazie danych w postaci haszowanej, co minimalizuje ryzyko ich ujawnienia w przypadku nieautoryzowanego dostępu do bazy. Do haszowania haseł zastosowano bibliotekę bcrypt, która zapewnia wysoki poziom bezpieczeństwa dzięki odporności na ataki brute-force. W procesie rejestracji i logowania weryfikacja hasła odbywa się poprzez porównanie jego haszowanej wersji z wartością przechowywaną w bazie danych.

4.1.6 Najważniejsze fragmenty kodu

```
3 references
public string GenerateJwtToken(string email, int userId)
{
    var issuer:string? = _configuration["Jwt:Issuer"];
    var audience:string? = _configuration["Jwt:Audience"];
    var key:byte[] = Encoding.UTF8.GetBytes(_configuration["Jwt:Key"] ?? string.Empty);
    var signin = new SigningCredentials(new SymmetricSecurityKey(key), algorithm: SecurityAlgorithms.HmacSha256);
    var subject = new ClaimsIdentity(new[]
    {
        new Claim( type:JwtRegisteredClaimNames.Name, value:email),
        new Claim(type:ClaimTypes.NameIdentifier, value:userId.ToString())
    });

    var tokenDescriptor = new SecurityTokenDescriptor
    {
        Issuer = issuer,
        Audience = audience,
        Subject = subject,
        Expires = DateTime.UtcNow.AddDays(1),
        SigningCredentials = signin
    };
    var tokenHandler = new JwtSecurityTokenHandler();
    var token = tokenHandler.CreateToken(tokenDescriptor);
    var tokenString = tokenHandler.WriteToken(token);

    return tokenString;
}
```

Rysunek 1 Generowanie tokenu JWT

```
[AllowAnonymous]
[HttpPost(template: "Register")]
0 references
public async Task<ActionResult<string>> Register([FromBody] RegisterModel model)
{
    try
    {
        var jwt:string = await _userService.Register(model.Email, model.Password, model.Name, model.PhoneNumber);
        return Ok(jwt);
    }
    catch (UserAlreadyExistsException e)
    {
        return BadRequest(e.Message);
    }
    catch (Exception e)
    {
        return StatusCode(StatusCode.Status500InternalServerError);
    }
}
```

Rysunek 2 Endpoint do rejestracji użytkownika

```

1 reference
public string HashPassword(string password)
{
    var hashedPassword:string? = BCrypt.Net.BCrypt.EnhancedHashPassword(password, HashType.SHA384, workFactor: 13);
    return hashedPassword;
}

1 reference
public bool VerifyPassword(string password, string hashedPassword)
{
    return BCrypt.Net.BCrypt.EnhancedVerify(text: password, hash: hashedPassword);
}

```

Rysunek 3 Hashowanie i walidacja hasła

```

8 references
public class Product
{
    7 references
    public int Id { get; set; }
    1 reference
    public string Name { get; set; }
    1 reference
    public double Price { get; set; }
    1 reference
    public bool InStock { get; set; }
    1 reference
    public string ImageUrl { get; set; }
    0 references
    public Category Category { get; set; }
}

```

Rysunek 4 Przykładowy obiekt EF Core

4.1.7 Testy aplikacji

Backend sklepu internetowego został gruntownie przetestowany zarówno pod kątem poprawności działania, jak i zgodności z wymaganiami funkcjonalnymi. Testowanie odbywało się na kilku poziomach, w tym poprzez manualne testy API oraz testy jednostkowe.

Do manualnego testowania endpointów API wykorzystano swagger, który został zintegrowany z aplikacją w celu automatycznego generowania dokumentacji interfejsu API. Swagger umożliwił łatwe testowanie metod HTTP, bez potrzeby korzystania z zewnętrznych narzędzi. Testerzy i programiści mogli szybko sprawdzać działanie poszczególnych endpointów, przysyłając przykładowe dane i analizując odpowiedzi serwera. Dzięki tej integracji możliwe było także wczesne wykrywanie błędów w strukturze danych wejściowych i odpowiedziach API.

Dodatkowo aplikacja została przetestowana za pomocą testów jednostkowych, które zweryfikowały poprawność działania kluczowych funkcji backendu. Do testów jednostkowych użyto NUnit, co pozwoliło na symulowanie zależności i testowanie logiki biznesowej w izolacji od innych komponentów.

Przykłady przetestowanych scenariuszy:

- poprawność obliczania wartości zamówienia,
- obsługę wyjątków i błędów,
- rejestrowanie użytkownika z takim samym mailem

5. Wnioski

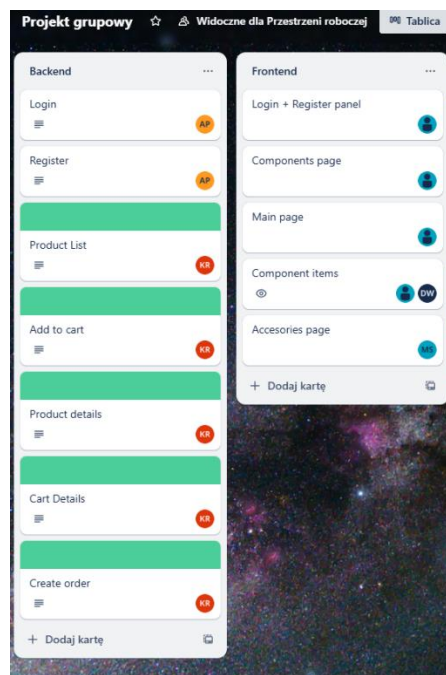
5.1 Praca zespołowa

Praca nad projektem przebiegała z wykorzystaniem platformy Trello, która umożliwiała efektywne przydzielanie zadań i monitorowanie postępów. Dzięki temu każdy członek zespołu wiedział, za co jest odpowiedzialny, co usprawniło realizację większości zadań. Niestety, testy nie zostały przeprowadzone z powodu braku kontaktu z jednym z członków zespołu, co wpłynęło na finalizację tego elementu projektu.

Rozmowy zespołowe prowadzone były na platformie Discord, gdzie stworzyliśmy dedykowany kanał dla projektu. Korzystaliśmy zarówno z kanału tekstowego, jak i głosowego. Jednakże kanał głosowy nie przechowuje logów rozmów, dlatego brak jest zapisów z prowadzonych dyskusji.

Spotkania zespołu odbywały się regularnie w soboty podczas weekendów zjazdowych, co pozwoliło na omówienie bieżących spraw i efektywne podejmowanie decyzji dotyczących dalszych działań. Wspólne rozmowy i ustalenia miały kluczowe znaczenie dla realizacji projektu, mimo pewnych wyzwań związanych z komunikacją.

Podsumowując, praca zespołowa przebiegała sprawnie dzięki wykorzystaniu nowoczesnych narzędzi, choć problemy z kontaktem z jednym z członków zespołu wpłynęły na niedokończenie niektórych elementów projektu.



Rys. 4. Ostateczny wygląd tablicy Trello

5.2 Podsumowanie pracy zespołowej

Co działało dobrze?

1. **Podział zadań:**
Dzięki wykorzystaniu platformy Trello praca była dobrze zorganizowana. Każdy członek zespołu miał jasno określone zadania, co pozwoliło na efektywne wykorzystanie czasu i zasobów.
2. **Regularne spotkania:**
Spotkania zespołu odbywały się w soboty podczas weekendów zjazdowych. Pozwoliło to na bieżące omawianie postępów i szybkie podejmowanie decyzji.
3. **Technologie:**
Praca nad aplikacją webową z wykorzystaniem Reacta przebiegała płynnie. Projekt – internetowy sklep o nazwie **TechZone**, specjalizujący się w sprzedaży sprzętu komputerowego – został zrealizowany z dużym zaangażowaniem zespołu.
4. **Komunikacja:**
Discord okazał się wygodną platformą do prowadzenia rozmów głosowych i tekstowych. Dedykowany kanał głosowy pozwolił na sprawne omawianie problemów w czasie rzeczywistym.

Jakie były wyzwania?

1. **Brak kontaktu z jednym członkiem zespołu:**
Jednym z największych wyzwań był brak kontaktu z jednym z członków zespołu, co uniemożliwiło przeprowadzenie testów aplikacji. To wpłynęło na finalną wersję projektu, która nie mogła zostać w pełni zweryfikowana.
2. **Brak zapisów rozmów:**
Ze względu na wykorzystanie kanału głosowego na Discordzie, nie mamy logów z rozmów. Może to utrudniać w przyszłości dokumentację procesu decyzyjnego.
3. **Koordinacja zadań:**
Chociaż Trello pomogło w organizacji pracy, sporadyczne opóźnienia w realizacji zadań wynikały z braku stałej dostępności wszystkich członków zespołu.

Jakie wnioski można wyciągnąć na przyszłość?

1. **Lepsze zarządzanie zespołem:**
Ważne jest, aby od początku projektu ustalić zasady odpowiedzialności za kontakt i postępy prac. W przyszłości należy rozważyć wyznaczenie osoby odpowiedzialnej za monitorowanie komunikacji w zespole.
2. **Automatyzacja zapisów komunikacji:**
Warto korzystać z narzędzi, które umożliwiają zapis rozmów, takich jak kanały tekstowe lub dedykowane systemy do notatek z zebrań (np. Google Docs, Notion).
3. **Testy jako priorytet:**
W przyszłych projektach testowanie powinno być etapem zaplanowanym z wyprzedzeniem, z odpowiedzialnością przypisaną całemu zespołowi, aby uniknąć zależności od jednej osoby.

4. **Usprawnienie komunikacji:**

Discord sprawdził się jako platforma do rozmów, ale w przyszłości warto rozważyć dodatkowe narzędzia, takie jak Slack, które lepiej integrują różne formy komunikacji (np. przypomnienia, listy zadań).

Podział zadań i metody komunikacji

Zadania były podzielone w Trello, co zapewniło przejrzystość i możliwość śledzenia postępów. Główne etapy realizacji projektu obejmowały:

- **Frontend:** Tworzenie komponentów aplikacji w React, projektowanie interfejsu użytkownika, w tym widoku sklepu.
- **Backend:** Obsługa danych produktów, w tym ich kategoryzacja i zarządzanie stanem magazynowym.
- **Spotkania:** Sobota była głównym dniem zespołowych spotkań, podczas których omawialiśmy postępy i trudności.

Pomimo wyzwań, projekt został zrealizowany z dużym zaangażowaniem zespołu, a uzyskane doświadczenia będą cenne przy kolejnych projektach.