**Spis treści**

1. **Wstęp**
   1. Wprowadzenie
   2. Cel pracy
2. **Założenia projektowe**
   1. Opis wzorca architektonicznego
   2. Wymagania od strony użytkownika
   3. Sposób realizacji
3. **Projekt**
   1. Diagram przypadków użycia
   2. Scenariusze przypadku użycia
   3. Diagram przebiegu(sekwencji)
   4. Diagram ERD
4. **Implementacja**
   1. Wykorzystane środowisko i narzędzia programistyczne
   2. Problemy implementacyjne
5. **Instrukcja obsługi dla użytkownika**
   1. Instrukcja panelu głównego użytkownika
   2. Instrukcja podłączenia bazy danych
   3. Opis zabezpieczenia połączenia z bazą
   4. Instrukcja tworzenia testu
   5. System weryfikacji użytkownika
   6. Weryfikacja odpowiedzi
   7. Instrukcja wykorzystywania narzędzi SQL
   8. Instrukcja pisania testu SQL
6. **Podsumowanie**
   1. Wykonane prace
   2. Dalszy rozwój aplikacji
7. **Wykaz literatury**
8. **Załączniki**

# 1. Wstęp

## 1.1.Wprowadzenie

Dzisiejszy świat nie mógłby się obejść bez technologii. Człowiek spędza większość dnia używając Internetu w przeróżny sposób. Czy to w celach zarobkowych czy rozrywkowych. Aktualnie Internet jest nieodłączną częścią naszego życia. Informatyka zdominowała prawie każdy aspekt handlu jak i rozrywki. Zapotrzebowanie na nowe lepsze i bardziej rozwinięte aplikacje internetowe jest coraz większa. Każdy poważny przedsiębiorca posiada swoją stronę/aplikacje internetową w celach promocyjnych oraz informacji. Ale także ważnym aspektem jest kto takie usługi tworzy i jak dobre mogą się stać. To programiści tworzą i rozwijają strony/aplikacje internetowe. Szukają nowych rozwiązań na poszczególne problemy w danej branży i starają się je rozwiązać. Dlatego tak ważna jest wiedza i doświadczenie programistów. Programista programuje w wielu językach i wykorzystuje je w swojej pracy. Ale ten którego używa się najczęściej to SQL. Ten język operuje na bazach relacyjnych które przechowują dane. Ale tą bazą trzeba zarządzać i wykorzystywać ją tworząc przeróżne wykresy analityczne. Aktualnie transakcje czy przelewy robione są przez Internet i wykorzystuje się w nich język SQL dlatego tak ważne jest kształcenie programistów w tym języku i wykorzystywanie go.

Dlatego w pracy podjęto temat stworzenia aplikacji internetowej która będzie w stanie zweryfikować wiedze programisty na temat języka SQL. Aplikacja będzie przeprowadzać testy wcześniej przygotowane przez osobę egzaminującą. Do już napisanych testów egzaminator będzie miał wgląd i będzie mógł ocenić wiedze studenta w zakresie języka SQL. Piszący egzamin będzie miał też do dyspozycji narzędzie pozwalające wykorzystywać zapytania SQL podczas pisania testu w celu weryfikacji poprawności zapytania.

**1.2.Cel pracy**

Celem pracy jest stworzenie aplikacji internetowej umożliwiającej weryfikacje wiedzy studentów z języka SQL na podstawie testu stworzonego przez egzaminatora. Aplikacja będzie spełniała określone wymagania, takie jak:

* Możliwość dodania własnej bazy danych jako podstawę do stworzenia testu.
* Możliwość stworzenia testu wykorzystującą wcześniej przygotowaną bazę danych, dodają własne zadania do rozwiązania.
* Weryfikacja wcześniej napisanych testów przez egzaminowanych.
* Możliwość używania narzędzia zapytań SQL podczas pisania testu.
* System zabezpieczeń blokujących wejście do testu użytkownikom nieupoważnionym .
* System aktywacyjny określający w jakim przedziale czasu test jest aktywny.
* System zabezpieczeń nie pozwalający wejść do testu drugi raz temu samemu użytkownikowi gdy już zakończył test.

1. **Założenia projektowe**

**2.1 Opis wzorca architektonicznego**

Aplikacja będzie tworzona na wzorcu architektonicznym MVC. Model ten posiada trzy podstawowe warstwy:

* Model (reprezentuje problem lub logikę aplikacji )
* Widok ( jest odpowiedzialny za wyświetlanie interfejsów użytkowników oraz posiada możliwość posiadania podwidoków)
* Kontroler (przyjmuje dane od użytkowników i reaguje na nie w określony sposób zaprogramowany przez progamistów)

Logiką całej aplikacji zarządzają kontrolery to one reagują na zapytania użytkownika, które pobierane są z widoków. Aktualnie wzorzec ten jest bardzo popularny i używany do tworzenia projektów na dużą skale.

**Zalety wzorca MVC:**

* Porządek w kodzie poprzez podzielenie warstw
* Aplikacje można łatwo rozbudowywać
* Struktura jest przejrzysta co pomaga w zrozumieniu aplikacji
* Dobra do pracy zespołowej
* Brak zależności pomiędzy modelem a widokiem

**2.2 Wymagania od strony użytkownika**

* System umożliwia odwiedzającemu zarejestrować się i zalogować na konto
* System umożliwia użytkownikowi dodanie połączenia z przygotowaną bazą danych
* System umożliwia użytkownikowi dodanie nowego testu razem z określeniem liczby pytań i ich zawartości.
* System umożliwia użytkownikowi dołączenie do wybranego testu za pomocą klucza udostępnionego przez osobę która utworzyła dany test
* System daje użytkownikowi narzędzie zapytań SQL, które może używać podczas pisania strony bez jej ponownego przeładowywania
* System umożliwia użytkownikowi tworzącego test dostęp do odpowiedzi osób które dany test uzupełniły

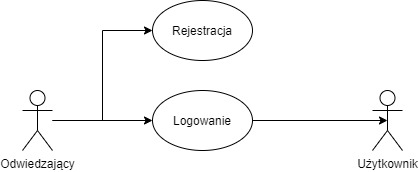
2.3

**2.3 Sposób realizacji**

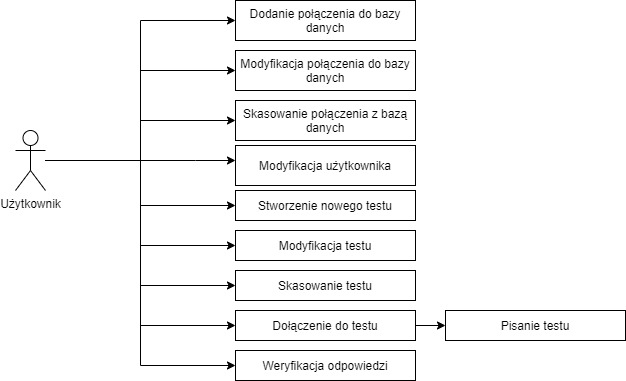
Aplikacja będzie udostępniana na przeglądarce. Do stworzenia aplikacji niezbędne będzie użycie takich narzędzi jak ASP .NET, bazy danych (Microsoft SQL Server, Oracle Database, PostgreSQL). Diagramy zostaną wykonane dzięki przeglądarkowej aplikacji do tworzenia diagramów (Draw.io). Natomiast projekt interfejsu zostanie stworzony przy użyciu Bootstrap oraz JavaScript.

# 3. Projekt

## 3.1 Diagram przypadków użycia



*3.1 Diagram dostępu do aplikacji*



*3.1 Diagram przypadków użycia dla użytkownika*

## 3.2. Scenariusze przypadków użycia

Tabela 3.1: PU-1 –Rejestracja

|  |  |
| --- | --- |
| **Lp.** | PU-1 |
| **Nazwa** | Rejestracja |
| **Warunki początkowe** |  |
| **Opis** | Odwiedzający rejestruje się na aplikacje |
| **Ścieżka główna** | 1. Odwiedzający naciska przycisk „Rejestracja”  2. Aplikacja wyświetla formularz rejestracyjny  3. Odwiedzający uzupełnia dane:   1. Imię 2. Nazwisko 3. Numer indeksu 4. Email 5. Login 6. Hasło   4. Odwiedzający wciska przycisk „Zarejestruj”  5. Użytkownik został dodany do bazy danych |
| **Ścieżka alternatywna** | 3a. Odwiedzający źle uzupełnił dane  4a. System otwiera formularz z powiadomieniem |
| **Warunki końcowe** | Dodano nowego użytkownika |

Tabela 3.2: PU-2 –Logowanie

|  |  |
| --- | --- |
| **Lp.** | PU-2 |
| **Nazwa** | Logowanie |
| **Warunki początkowe** |  |
| **Opis** | Odwiedzający loguje się na aplikacje |
| **Ścieżka główna** | 1. Odwiedzający uzupełnia pola logowania  4. Odwiedzający wciska przycisk „Zaloguj”  5. System otwiera formularz użytkownika |
| **Ścieżka alternatywna** | 3a. Odwiedzający wprowadził błędne dane logowania  4a. System otwiera formularz z powiadomieniem „Błędny login lub hasło” |
| **Warunki końcowe** | Zalogowanie na formularz użytkownika |

Tabela 3.3: PU-3 –Dodanie połączenia do bazy danych

|  |  |
| --- | --- |
| **Lp.** | PU-3 |
| **Nazwa** | Dodanie połączenia do bazy danych |
| **Warunki początkowe** | Użytkownik zalogowany w systemie |
| **Opis** | Użytkownik chce dodać połączenie do bazy danych |
| **Ścieżka główna** | 1. Użytkownik naciska przycisk „Dodaj” w zakładce „Bazy danych”  2. Aplikacja wyświetla formularz połączenia z bazą.  3. Użytkownik uzupełnia dane:   1. Typ bazy 2. Nazwa bazy 3. Server bazy 4. Login bazy 5. Hasło bazy   4. Użytkownik wciska przycisk „Dodaj”  5. Baza zostaje zapisana |
| **Ścieżka alternatywna** | 3a. Użytkownik źle uzupełnił dane  4b. Sprawdzanie połączenia z bazą dało wynik negatywny  5b. Błąd Systemu. Połączenie nie zostało dodane. |
| **Warunki końcowe** | Dodano połączenie z bazą |

Tabela 3.4: PU-4 –Zmodyfikowanie połączenia z bazą danych

|  |  |
| --- | --- |
| **Lp.** | PU-4 |
| **Nazwa** | Modyfikacja połączenia z bazą danych |
| **Warunki początkowe** | Połączenie zapisane w bazie danych |
| **Opis** | Użytkownik modyfikuje połączenie z bazą |
| **Ścieżka główna** | 1. Użytkownik naciska przycisk „Edytuj” przy połączeniu które chce zmienić  2. Aplikacja wyświetla formularz połączenia z bazą.  3. Użytkownika modyfikuje dane:   1. Typ bazy 2. Nazwa bazy 3. Server bazy 4. Login bazy 5. Hasło bazy   4. Użytkownik wciska przycisk „Zapisz bazę”  5. Baza zostaje zmodyfikowana. |
| **Ścieżka alternatywna** | 3a. Użytkownik źle uzupełnił dane  4b. Sprawdzanie połączenia z bazą dało wynik negatywny  5b. Błąd Systemu. Połączenie nie zostało dodane. |
| **Warunki końcowe** | Zmodyfikowano połączenie z bazą |

Tabela 3.5: PU-5 –Usunięcie połączenia z bazą danych

|  |  |
| --- | --- |
| **Lp.** | PU-5 |
| **Nazwa** | Usunięcie połączenia z bazą danych |
| **Warunki początkowe** | Połączenie zapisane w bazie danych |
| **Opis** | Użytkownik usuwa połączenie z bazą |
| **Ścieżka główna** | 1. Użytkownik naciska przycisk „Usuń” przy połączeniu które chce usunąć  2. System sprawdza czy do połączenia są przypisane jakieś testy i odpowiedzi.  3. Jeśli tak wyświetlany jest komunikat „Usuwając bazę usuniesz też wszystkie testy z nią połączoną. Czy chcesz usunąć bazę ?”  4. Użytkownik naciska tak  5. Usuwane są wszystkie testy i odpowiedzi połączone z tą bazą. |
| **Ścieżka alternatywna** | 4a. Użytkownik naciska nie  5a. Połączenie nie zostaje usunięte |
| **Warunki końcowe** | Usunięto połączenie z bazą |

Tabela 3.6: PU-6 –Modyfikacja użytkownika

|  |  |
| --- | --- |
| **Lp.** | PU-6 |
| **Nazwa** | Modyfikacja Użytkownika |
| **Warunki początkowe** | Użytkownik zalogowany w systemie |
| **Opis** | Użytkownik modyfikuje swoje dane |
| **Ścieżka główna** | 1. Użytkownik naciska przycisk „Edytuj” w panelu użytkownika  2. Aplikacja wyświetla formularz danych użytkownika.  3. Użytkownik zmienia wartości  4. Użytkownik naciska „Zapisz”  5. Dane zostają zmodyfikowane i zapisane |
| **Ścieżka alternatywna** | 3a. Użytkownik błędnie wypełni pola  4a. Aplikacja wyświetla błąd walidacyjny i wraca do formularza  3b. Użytkownik nie nacisnął przycisku „Zapisz”  4b. Dane nie zostały zmodyfikowane |
| **Warunki końcowe** | Zmodyfikowano dane użytkownika |

Tabela 3.7: PU-7 –Stworzenie Testu

|  |  |
| --- | --- |
| **Lp.** | PU-7 |
| **Nazwa** | Stworzenie testu |
| **Warunki początkowe** | Użytkownik zalogowany w systemie oraz dodał wcześniej połączenie z bazą |
| **Opis** | Użytkownik tworzy nowy test |
| **Ścieżka główna** | 1. Użytkownik naciska przycisk „Dodaj” w zakładce „Testy”  2. Aplikacja wyświetla formularz tworzenia testu.  3. Użytkownik uzupełnia dane:   1. Nazwa testu 2. Wybiera połączenie z bazą 3. Datę początku 4. Datę końca   4. Użytkownik naciska „Dodaj test”  5. Aplikacja wyświetla formularz tworzenia pytań do stworzonego testu  6. Użytkownik dodaje pytania przyciskiem „Dodaj”  7. Użytkownik uzupełnia wygenerowane pole na pytanie  8. Użytkownik naciska przycisk „Zapisz”  9. Test razem z odpowiedziami zostaje zapisany do bazy danych |
| **Ścieżka alternatywna** | 3a. Użytkownik błędnie wypełni pola  4a. Aplikacja wyświetla błąd walidacyjny i wraca do formularza tworzenia testu  8a. Użytkownik nie dodał ani jednego pytania  9a. Aplikacja wyświetla błąd walidacyjny i wraca do formularza tworzenia pytań  3b. Data końca została źle ustawiona  4b. Aplikacja wyświetla błąd walidacyjny i wraca do formularza tworzenia testu  8b. Użytkownik nie wypełnił wszystkich pól które stworzył  9b. Aplikacja wyświetla błąd walidacyjny i wraca do formularza tworzenia pytań |
| **Warunki końcowe** | Dodano nowy test |

Tabela 3.8: PU-8 –Modyfikowanie testu

|  |  |
| --- | --- |
| **Lp.** | PU-8 |
| **Nazwa** | Modyfikowanie testu |
| **Warunki początkowe** | Test zapisany w bazie danych |
| **Opis** | Użytkownik modyfikuje test |
| **Ścieżka główna** | 1. Użytkownik naciska przycisk „Edytuj” przy wybranym teście  2. Aplikacja wyświetla formularz edycji testu.  3. Użytkownik modyfikuje dane:   1. Nazwa testu 2. Wybiera połączenie z bazą 3. Datę początku 4. Datę końca   4. Użytkownik naciska „Edytuj”  5. Aplikacja wyświetla formularz edycji pytań  6. Użytkownik dodaje pytania przyciskiem „Dodaj”  7. Użytkownik uzupełnia wygenerowane pola na pytanie  8. Użytkownik naciska przycisk „Zapisz”  9. Test razem z odpowiedziami zostaje zmodyfikowany i zapisany w bazie danych |
| **Ścieżka alternatywna** | 3a. Użytkownik błędnie wypełni pola  4a. Aplikacja wyświetla błąd walidacyjny i wraca do formularza edycji testu  8a. Użytkownik modyfikując nie ma ani jednego pytania do danego testu  9a. Aplikacja wyświetla błąd walidacyjny i wraca do formularza edycji pytań  3b. Data końca została źle ustawiona  4b. Aplikacja wyświetla błąd walidacyjny i wraca do formularza edycji testu  8b. Użytkownik nie wypełnił wszystkich pól które stworzył  9b. Aplikacja wyświetla błąd walidacyjny i wraca do formularza edycji pytań |
| **Warunki końcowe** | Zmodyfikowano wybrany test |

Tabela 3.9: PU-9 –Skasowanie testu

|  |  |
| --- | --- |
| **Lp.** | PU-9 |
| **Nazwa** | Skasowanie testu |
| **Warunki początkowe** | Test zapisany w bazie danych |
| **Opis** | Użytkownik usuwa test |
| **Ścieżka główna** | 1. Użytkownik naciska przycisk „Usuń” przy wybranym teście  2. System sprawdza czy do połączenia są przypisane jakieś pytania i odpowiedzi.  3. System wyświetla komunikat „Czy chcesz usunąć test ?”  4. Użytkownik naciska tak  5. Usuwane są wszystkie pytania i odpowiedzi należące do tego testu |
| **Ścieżka alternatywna** | 4a. Użytkownik naciska nie  5a. Test nie zostaje usunięty |
| **Warunki końcowe** | Skasowanie testu |

Tabela 3.10: PU-10 –Dołączenie do testu

|  |  |
| --- | --- |
| **Lp.** | PU-10 |
| **Nazwa** | Dołączenie do testu |
| **Warunki początkowe** | Użytkownik zalogowany oraz testu utworzony |
| **Opis** | Użytkownik dołącza do testu poprzez klucz |
| **Ścieżka główna** | 1. Użytkownik wpisuje klucz dostępu w zakładce nawigacji  2. Użytkownik naciska przycisk „Napisz”  3. System sprawdza klucz dostępu  4. Otwiera test |
| **Ścieżka alternatywna** | 4a. Podany klucz jest błędny  5a. System wyświetla komunikat „Błędny klucz”  4b. Podany klucz jest poprawny lecz okres aktywacji testu przeminął  5b. System wyświetla komunikat „Test jest nieaktywny” |
| **Warunki końcowe** | Otwarcie testu |

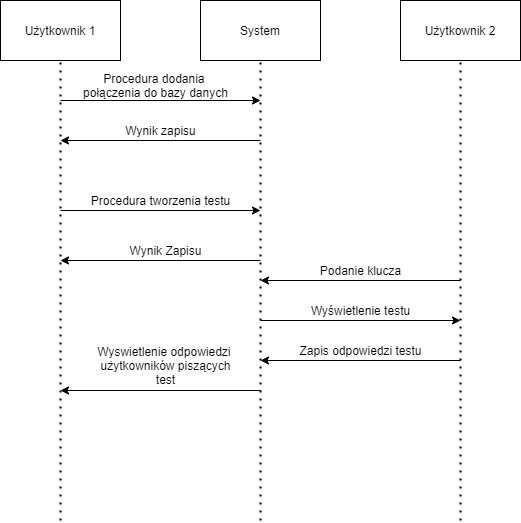
Tabela 3.11: PU-11 –Pisanie testu

|  |  |
| --- | --- |
| **Lp.** | PU-11 |
| **Nazwa** | Pisanie testu |
| **Warunki początkowe** | Otwarcie testu(Tabela 3.10) |
| **Opis** | Użytkownik odpowiada na pytania testu |
| **Ścieżka główna** | 1. Użytkownik używa narzędzia SQL połączonej z bazą danych  2. Zapytania użytkownika są wysyłane do bazy  3. System wypisuje je w formularzu obok pól odpowiedzi  4. Użytkownik uzupełnia pola odpowiedzi  5. System sprawdza klucz dostępu  6.Użytkownik naciska przycisk „Zakończ”  7.System zapisuje odpowiedzi i wyświetla komunikat „Zapisano test pomyślnie” |
| **Ścieżka alternatywna** | 2a. Użytkownik podaje błędne zapytanie  3a. System wyświetla komunikat o określonym błędzie składni języka SQL |
| **Warunki końcowe** | Zapis testu |

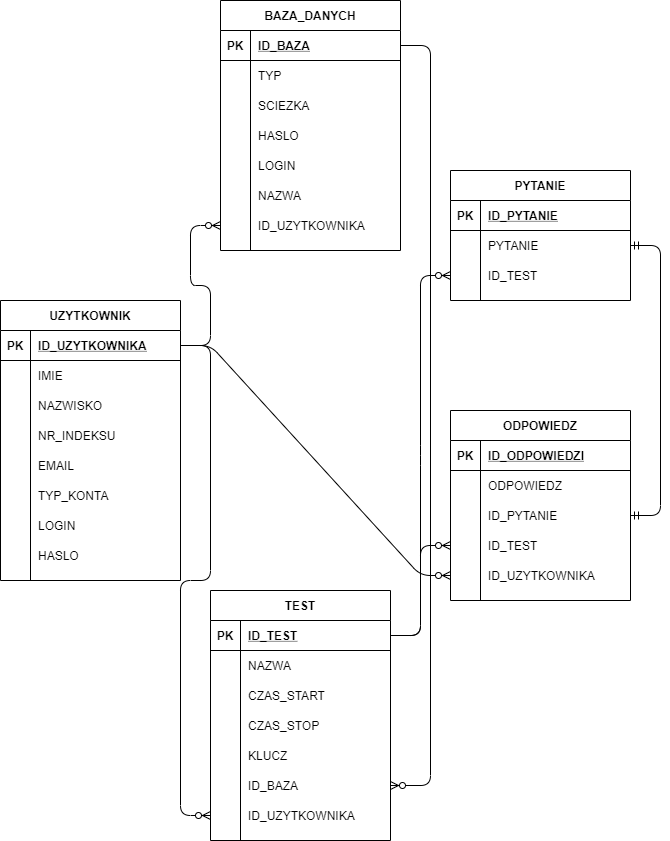
Tabela 3.12: PU-12 –Weryfikacja testu

|  |  |
| --- | --- |
| **Lp.** | PU-12 |
| **Nazwa** | Weryfikacja testu |
| **Warunki początkowe** | Użytkownik zalogowany oraz test napisany przez egzaminowanych |
| **Opis** | Użytkownik sprawdza odpowiedzi |
| **Ścieżka główna** | 1. Użytkownik naciska ikonę odpowiedzi przy wybranym teście  2. System otwiera formularz z listą osób które pisały dany test  3. Użytkownik naciska przycisk „Odpowiedzi” przy wybranym wybranej osobie  4.System otwiera formularz z odpowiedziami oraz danymi użytkownika |
| **Warunki końcowe** | Odczyt odpowiedzi |

## 3.2 Diagram przebiegu(sekwencji)



# 3.3 Diagram ERD



**4. Implementacja**

**4.1 Wykorzystane środowisko i narzędzia programistyczne**

**Microsoft SQL Server** – System do zarządzania bazami danych zaprojektowany przez Microsoft. Charakteryzuje się on językiem zapytań Transact-SQL, który jest rozwinięciem standardu ANSI/ISO. Jest to platforma bazodanowa typu klient-server.

**Oracle Database** – System zarządzania bazami danych stworzony przez Oracle Corporation. Obsługuje on standardowy system zapytań SQL oraz posiada język tworzenia procedur składowych PL/SQL

**PostgreSQL** – Otwarty system zarządzania relacyjnymi bazami danych. Pozwala on pisać w różnych językach programowania. Takich jak PL/pgSQL, PL/Phyton, PL/Perl, PL/Tcl. Posiada on też system wyzwalaczy, który jest uruchamiany po takich operacjach jak UPDATE, INSERT, DELETE, TRUNCATE

**Visual Studio** – Środowisko programistyczne do tworzenia oprogramowania konsolowego z graficznym interfejsem użytkownika. Aplikacje mogą być uruchamiane na takich platformach jak: Microsoft Windows, Windows CE, .NET Framework, Microsoft Silverlight, Linux, MacOS, XBOX. Pozwala tworzyć aplikacje WPF, Windows Forms, Web Sites, Web Applications oraz inne.

**C#** - Język programowania zaprojektowany dla Microsoft. Napisany program w tym języku kompilowany jest do Common Intermediate Language(CIL). Następnie wykonywany jest w środowisku uruchomieniowym .NET Framework, .NET Core, Mono, DotGNU. Jest więc językiem obiektowym ogólnego przeznaczenia.

**HTML –** Jest to język znaczników do tworzenia dokumentów hipertekstowych. Język ten określa strukturę strony internetowej oraz wygląd strony w przeglądarce. Pozwala też dodawać obiekty plikowe oraz elementy baz danych i tworzyć interaktywne formularze.

**CSS** – Język służy do wyświetlania strony internetowej. Jest to lista poleceń decydujących w jaki sposób elementy na stronie jak i sama strona mają być rozłożone w przeglądarce. Możemy określać takie elementy jak kolor, czcionkę, marginesy i wiele innych parametrów.

**AJAX –** Technika programowania aplikacji internetowych. Pozwalająca załadować nowe elementy do witryny bez przeładowywania jej w całości. Umożliwia on bardziej dynamiczną interakcje z użytkownikiem niż tradycyjny model przesyłania strony HTML. Podstawę Ajara tworzą XMLHttpRequest, JavaScript, XML.

**4.2** Problemy implementacyjne

**7.Wykaz Literatury**