

Sprawozdanie z pracy projektowej

Dawid Buchcic, Marcel Zieliński, Bartosz Biel

05.06.2024

1 Cel projektu

Celem naszego projektu było utworzenie aplikacji dającej rekomendację do gry League of Legends.

2 Opis funkcjonalny systemu

1. Generowanie rekomendacji
2. Wyświetlenie stanu serwera League of Legends (Poprzez Riot API)
3. Generowanie listy wszystkich dostępnych bohaterów oraz przedmiotów
4. Rejestracja użytkownika
5. Kontakt Mailowy

3 Opis technologiczny

- Technologie programowania: **C#**, **.NET Framework**
- Front-end: **HTML**, **CSS (Bootstrap)**, **JavaScript**
- Baza danych: **SQL Server**
- Środowisko programistyczne: **Visual Studio**
- System kontroli wersji: **Git**

4 Wyszczególnione wdrożone zagadnienia kwalifikacyjne

1. Framework MVC: Wykorzystanie frameworka ASP.NET Core oraz Serilog dla lepszej organizacji kodu oraz generacji Loggera.

2. Framework CSS: Zastosowanie frameworka Bootstrap w celu skrócenia czasu oraz większej wygody tworzenia interfejsu.
3. Baza danych: Aplikacja jest połączona z bazą danych w SQL, w której są przechowywane dane o bohaterach.
4. Cache: Aplikacja przechowuje Cache dla listy bohaterów
5. Dependency manager: Wykorzystanie systemu zarządzania zależnościami dla efektywnego zarządzania bibliotekami.
6. HTML: Stworzenie szkieletu aplikacji przy użyciu HTML.
7. CSS: Aplikacja jest ostylowana za pomocą CSS używając Bootstrap.
8. JavaScript: Niektóre funkcje, takie jak propozycje w pasku wyszukiwania, zostały wykonane w JavaScript.
9. Routing: Wykorzystanie routingu dla czytelnych adresów URL.
10. ORM: Wykorzystanie mapowania obiektowo-relacyjnego dla efektywnej interakcji z bazą danych.
11. Lokalizacja: Dodanie możliwości przełączania języka aplikacji (Polski oraz Angielski) dla dostosowania do preferencji użytkowników.
12. Mailing: Otrzymywanie maili z aplikacji dla komunikacji z użytkownikami.
13. Formularze: Przesyłanie danych do aplikacji przez formularze dla interakcji użytkowników z systemem.
14. Asynchroniczne interakcje: Zaimplementowane asynchroniczne interakcje z serwerem dla płynnej interakcji użytkownika.
15. Konsumpcja API: Wykorzystanie Riot API dla wyświetlenia statusu serwerów League of Legends.
16. Publikacja API: Opublikowane API udostępnia informacje dotyczące rekomendacji przeciwko bohaterowi oraz jego atrybutów.
17. RWD: Aplikacja jest responsywna, co umożliwia na wyświetlanie jej na różnych urządzeniach.
18. Logger: Logowanie akcji w systemie dla monitorowania działań użytkowników.

5 Instrukcja lokalnego i zdalnego uruchomienia systemu

5.1 Instalacja wymaganych narzędzi

1. Zainstaluj następujące narzędzia:

- Visual Studio
- Microsoft SQL Server
- Git

5.2 Sklonuj repozytorium

1. Sklonuj repozytorium w wybranym folderze używając polecenia:

```
git clone https://github.com/Zielinski-Marcel/Projekt-PipsiI.git
```

5.3 Otwarcie projektu

1. Otwórz plik `PoroCounter2.sln` w programie Visual Studio.

5.4 Konfiguracja bazy danych

1. W pliku `Program.cs`, upewnij się, że łączysz się z bazą danych SQL Server, używając następującej konfiguracji:

```
builder.Services.AddDbContext<DB>(options => {  
    options.UseSqlServer(  
        "Server=*YOURSERVER;  
        Database=PPSIDB2;  
        Trusted_Connection=True;  
        TrustServerCertificate=True;");  
});
```

2. Uruchom program Microsoft SQL Server.
3. W Polu Object Explorer wybierz `Connect` i wybierz: `Database Engine`.
4. W polu `Server name` wpisz: `*YOURSERVER*`.
5. Po połączeniu się z serwerem SQL, zaimportuj bazę z pliku
6. Należy wygenerować kod API od Riot Games DB.bacpac.
7. w `EmailSender.cs` należy podać adres Email który będzie wysyłał maile.

5.5 Uruchomienie aplikacji

1. W programie Visual Studio uruchom aplikację.
2. Przy prawidłowej konfiguracji powinna się uruchomić strona internetowa. Jeśli `champions` albo `items` jest puste, oznacza to, że źle zaimportowana została baza danych lub zamiast tego użyta została komenda `update-database`, która utworzyła nową, pustą bazę danych.

6 Wnioski projektowe

Praca nad projektem do gry League of Legends była dla nas niemałym wyzwaniem. Ogromnym problemem było znalezienie odpowiednich źródeł do nauki, ponieważ większość z nich była przestarzała. Poznawanie .NET było bardzo czasochłonne i mało wydajne, ale na pewno warte poświęconego czasu. Zdobyte doświadczenie na pewno przyda się w przyszłości