Dzienniczek ucznia - dokumentacja­­­­­­­­­­­­

Zleceniodawca:

Jerzy Sołowianiuk

Wykonawca:

Gustaw Sołdecki 4pi

Spis treści

1. **Przedstawienie projektu3**
   1. Szczegółowe przedstawienie projektu3
   2. Analiza wymagań zleceniodawcy, określenie wymaganych funkcjonalności3
2. **Konceptualny projekt diagramów4**
3. **Projekt logiczny6**
4. **Projekt funkcjonalny aplikacji - dokumentacja7**

1. Przedstawienie projektu

Celem projektu jest stworzenie aplikacji bazodanowej z możliwością zarządzania internetowego oraz dostępu do interfejsu użytkownika poprzez przeglądarkę.

Tematem projektu jest realizacja bazy danych dla personalnego dziennika ucznia według podanego wzoru. Wykorzystane zostaną: dla bazy danych język SQL oraz narzędzie phpMyAdmin,   
dla interfejsu użytkownika język HTML, PHP, JavaScript oraz formatowanie CSS.

* 1. Szczegółowe przedstawienie tematu

Projekt opiera się na założeniu schematu przedstawionego przez zleceniodawcę (rys. 1). Wykonany ma zostać projekt bazy danych oraz implementacja jej jako działająca strona internetowa. Spełniona ma zostać funkcjonalność dodawania ocen dla konkretnych przedmiotów oraz wyświetlania ich w uporządkowany sposób dla adekwatnych użytkowników. Projekt zakłada łatwe wprowadzanie nowych danych do bazy danych.

1.2 Analiza wymagań zleceniodawcy, określenie wymaganych funkcjonalności

Projekt zakłada możliwość logowania kilku użytkowników do bazy danych.

Funkcjonalności jakie ma spełniać projekt:

- automatyczne wyświetlanie danych ocen dla zalogowanego użytkownika

- możliwość logowania różnych użytkowników

- łatwe wprowadzanie nowych danych

- graficzne przedstawienie danych z tabel z użyciem minimum 3 zapytań

Dodatkowe założenia projektowe:

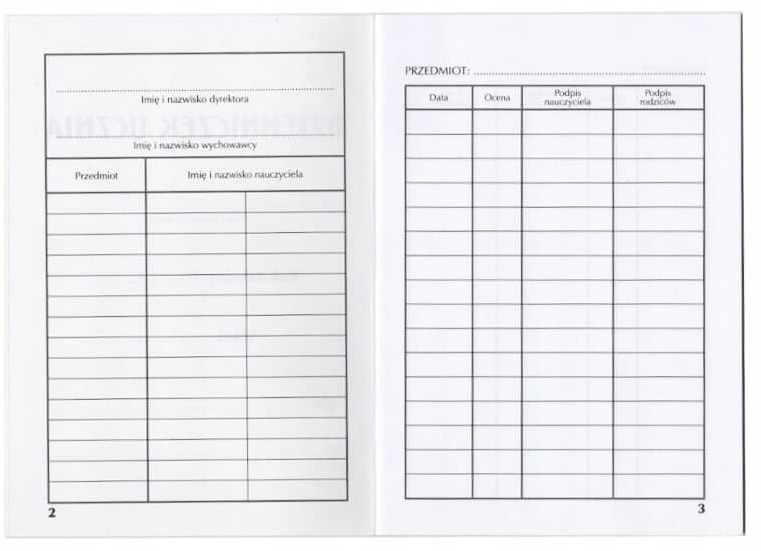
- wykonanie bazy danych w środowisku XAMPP

- wykorzystanie systemu MySQL z narzędziem phpMyAdmin

- użycie skryptów PHP w wizualizacji

- sporządzenie dokumentacji

- załączenie pliku tekstowego zawierającego wyeksportowaną bazę danych



*Rysunek 1 - przykładowy szkic bazy danych*

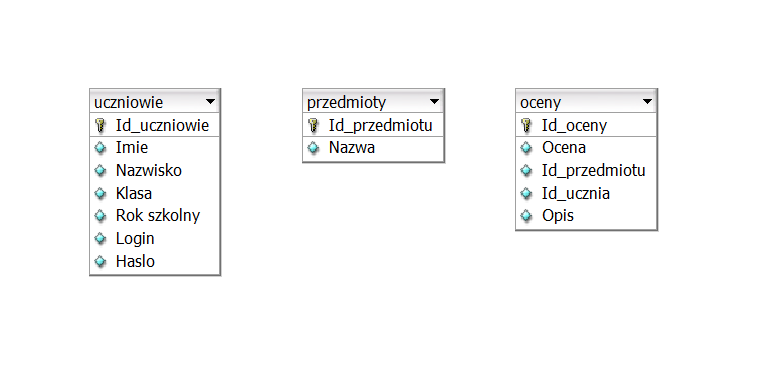
2. Konceptualny projekt diagramów

Encje oraz atrybuty zostały przedstawione na rys. 2. Projekt obejmuje trzy tabele (*oceny, przedmioty, uczniowie*) w bazie danych *dzienniczek\_ucznia*. Pomiędzy tabelami zachodzą   
relacje (rys. 3).

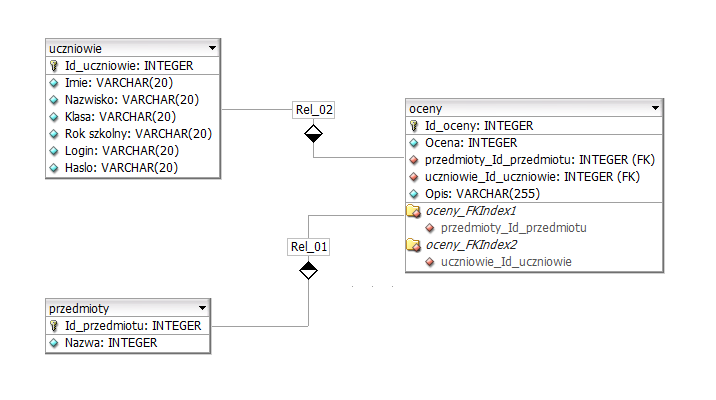
Główną tabelą jest encja *oceny*, każda ocena posiada automatycznie uzupełnianie *Id\_oceny*, które jest kluczem głównym. Kolumna *Id\_przemiotu* oraz *Id\_ucznia* w tej tabeli są związane relacjami odpowiednio z tabelami *przedmioty* oraz *uczniowie*. *Opis* jest nie wymaganą kolumną typu VARCHAR z maksymalną ilością znaków 200.

Encja *przedmioty* służy pomocniczo, aby przechowywać łatwo nazwy przedmiotów adekwatnie do przypisanego *Id\_przemiotu*.

Drugą najważniejszą tabelą jest *uczniowie*, umożliwia ona logowanie na odpowiednich użytkowników (uczniów). Zawiera ona informacje o uczniu takie jak: imię (*Imie*), nazwisko (*Nazwisko*), klasa (*Klasa*), rok szkolny (*Rok szkolny*) oraz login (*Login*) i hasło (*Haslo*). Dla każdego ucznia przypisywane jest automatycznie *Id\_ucznia* które jest kluczem głównym oraz czynnikiem przypisania oceny do ucznia.



*Rysunek 2 – encje oraz ich atrybuty*



*Rysunek 3 – diagram relacji pomiędzy encjami*

3. Projekt logiczny

Tabela oceny

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa | Typ danych | NULL | Opis |
| Id\_oceny | Int(11) | Nie | AI, klucz główny |
| Ocena | Int(2) | Nie |  |
| Id\_przedmiotu | Int(11) | Nie |  |
| Id\_ucznia | Int(11) | Nie |  |
| Opis | Varchar(200) | Tak |  |

Tabela uczniowie

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa | Typ danych | NULL | Opis |
| Id\_ucznia | Int(20) | Nie | AI, klucz główny |
| Imie | Varchar(20) | Nie |  |
| Nazwisko | Varchar(20) | Nie |  |
| Klasa | text | Nie |  |
| Rok szkolny | text | Nie |  |
| Login | Varchar(20) | Nie |  |
| Haslo | Varchar(20) | Nie |  |

Tabela przedmioty

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa | Typ danych | NULL | Opis |
| Id\_przedmiotu | Int(11) | Nie | AI, klucz główny |
| Nazwa | Varchar(20) | Nie |  |

3.1 Analiza zależności funkcyjnych 3NF – norm normalizacji

1NF – Pierwsza postać normalna:

Powyższa baza danych spełnia 1NF ponieważ wartości atrybutów są elementarne, atomowe, niepodzielne. Każda kolumna jest wartością skalarną - nie posiada własnej struktury, nie zawiera kolekcji. Nie występuje w bazie danych redundancja – nadmiarowość, informacje nie powtarzają się.

2NF – Druga postać normalna:

Żadna kolumna nie jest częściowo funkcyjnie zależna od jakiegokolwiek klucza potencjalnego. Baza danych spełnia drugą postać znormalizowaną.

3NF – Trzecia postać normalna:

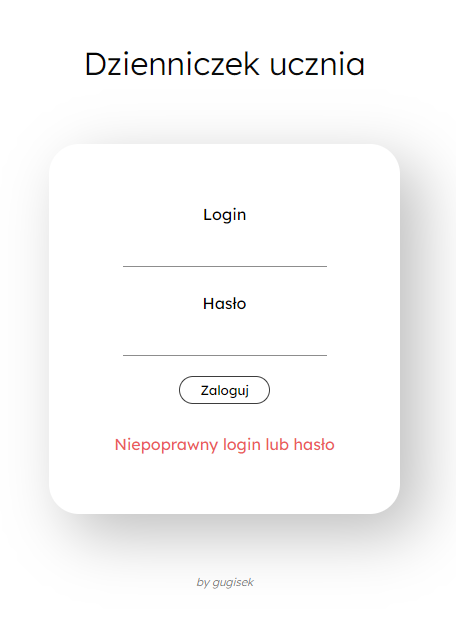
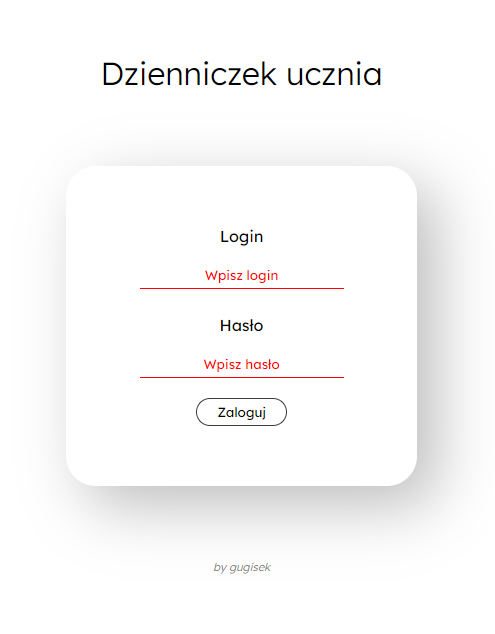
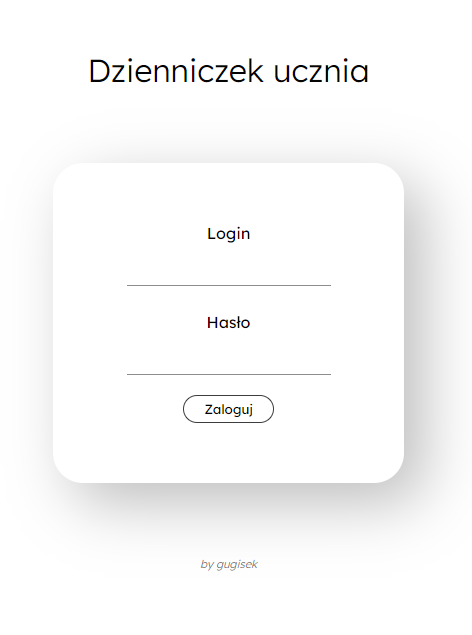
Przedstawiona baza danych nie spełnia tej formy normalizacji. W celu usprawnienia działania aplikacji tabele zostały w formie 2NF. Spełnienie formy 3NF wymagało by rozbicia tabeli *uczniowie*na encje: uczniowie, klasy, lata\_szkolne, loginy, hasła. Dla zastosowania w projekcie ta forma   
nie wykazywała by wystarczającej wydajności.

4. Projekt funkcjonalny aplikacji - dokumentacja

4.1 Logowanie

Logowanie odbywa się za pomocą skryptów w języku PHP (auth.php). Wykonanie operacji logowania wykonuję się za pomocą wpisania prawidłowego loginu w pierwsze pole wpisowe   
oraz adekwatnego hasła w drugie pole wpisowe (rys. 4). Domyślne konta (login hasło): Gustaw 1234   
oraz Andrzejek 1234.

Aplikacja posiada zabezpieczenia przed niepożądanym wtargnięciem do panelu użytkownika poprzez zastosowanie funkcji sesji ($\_SESSION). Również funkcjonują skrypty pełniące funkcje prewencyjne przed wysłaniem formularza z pustymi danymi (rys. 5) oraz nieprawidłowymi atrybutami hasła i loginu (rys. 6).



*Rys. 4 – formularz logowania Rys. 5 – przesłanie pustego formularza Rys. 6 –niepoprawne dane logowania*

4.2 Przegląd funkcjonalności panelu użytkownika

Panel użytkownika został zaprojektowany wedle zasad: intuicyjności, łatwości odbioru   
oraz pełnej funkcjonalności. Dodatkowo zostały zaimplementowane funkcje zegara i aktualnej daty poprzez załączenie skryptu JavaScript. Podstawowymi funkcjami są wyświetlanie danych w tabeli oraz wprowadzanie nowych danych.

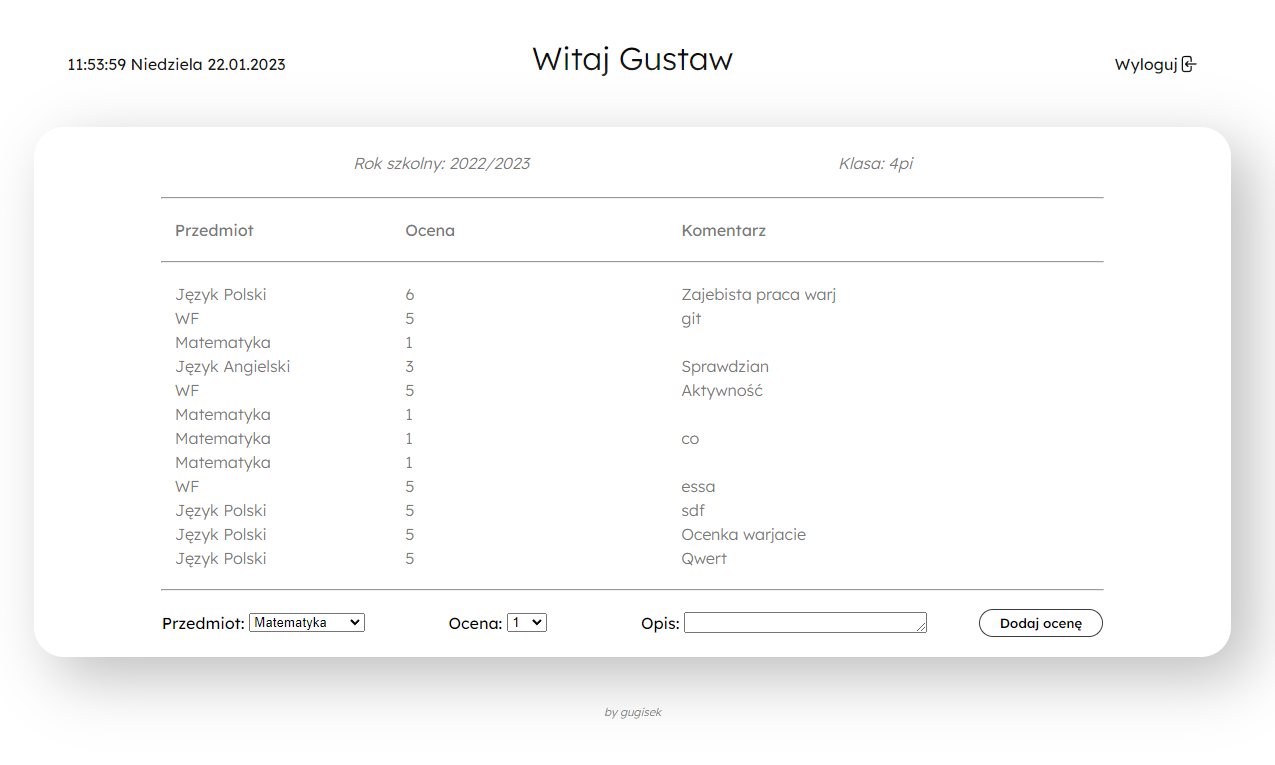
Prezentacja danych zaczyna się od wybrania z bazy danych informacji danych o użytkowniku  
 i wyświetlenie ich w odpowiednich sekcjach (rys. 7). Następnie automatycznie skrypt wyświetla dane poprzez pętle while dopóki nie wyświetli wszystkich wierszy (rys. 8).

Wprowadzanie informacji opiera się na formularzu oraz skrypcie w języku PHP (insert.php) wywoływanym poprzez zatwierdzenie formularza (rys. 8). Dane zostają poddane sprawdzeniu   
czy nie są puste i po wybraniu odpowiednich parametrów w formie, zostają one przesłane do bazy danych z nadanym odpowiednim *Id\_ucznia* w zależności od zalogowanego użytkownika.

Dodatkowo został zaimplementowany skrypt powodujący wylogowanie użytkownika (logout.php), było to niezbędne do poprawnego działania aplikacji oraz wygody użytkowania.



*Rysunek 7 – informacje o użytkowniku w panelu*



*Rysunek 8 – panel użytkownika*

4.3 Funkcjonalności daty

4.4 Przechowywanie danych sesji

4.5 Responsywność aplikacji

5. Dokumentacja techniczna

W wykonaniu aplikacji zostały użyte języki: HTML 5, CSS 3, PHP 8.0, SQL, JavaScript   
oraz narzędzia / aplikacje: Visual Studio Code, pre-class by gugisek, DBDesigner 4.

Dokumentacja dzienniczek ucznia

Wykonał:

Gustaw Sołdecki 4pi

Zlecił:

Jerzy Sołowianiuk