Zadanie na rozgrzewkę: Kalkulator średniej ocen

Napisz aplikację w JavaScript, która:

1. Pozwala użytkownikowi wprowadzić dowolną liczbę ocen (liczby całkowite od 1 do 6).
2. Po kliknięciu przycisku "Oblicz średnią" wyświetla średnią ocen z dokładnością do dwóch miejsc po przecinku.
3. Jeśli użytkownik wprowadzi nieprawidłowe dane (np. tekst lub liczby spoza zakresu), wyświetla komunikat o błędzie.

Rozwiązanie w JavaScript

```
SQL - FUNKCJE zliczające (agregujące)
SELECT AVG(ocena) FROM ocena WHERE id_przedmiot=10;
SELECT SUM(kolunma1) FROM nazwa_tabeli;
SELECT MIN(kolunma1) FROM nazwa_tabeli;
SELECT COUNT(*) FROM nazwa_tabeli;
SELECT COUNT(DISTINCT ocena) FROM ocena; // ile jest różnych ocen
```

Zadanie egzaminacyjne

UWAGA!

Numer, którym został podpisany arkusz egzaminacyjny (PESEL lub w przypadku jego braku numer paszportu) jest w zadaniu nazywany numerem zdającego.

Wykonaj aplikację internetową obsługującą notatki z zadaniami do wykonania, wykorzystując pakiet XAMPP, edytor grafiki rastrowej oraz edytor zaznaczający składnię.

Aby wykonać zadanie, należy zalogować się na konto `Egzamin` bez hasła. Na pulpicie znajduje się archiwum `7z` o nazwie `PlikiCz202502` zabezpieczone hasłem: `#ToDoList@`

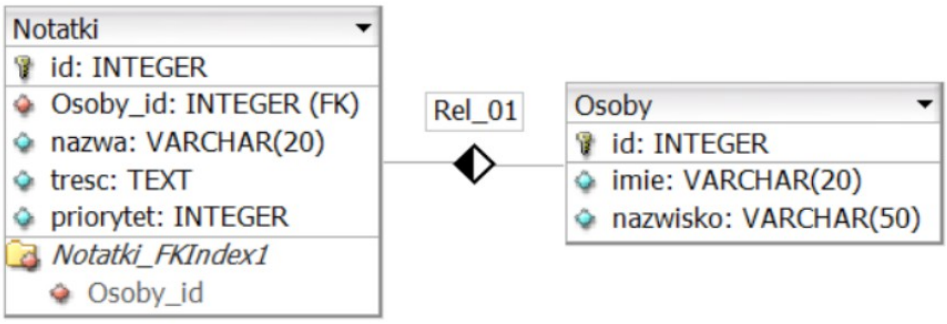
Archiwum należy rozpakować.

Na pulpicie konta `Egzamin` należy utworzyć folder. Jako nazwy folderu należy użyć `numeru zdającego`. Rozpakowane pliki należy umieścić w tym folderze. Po skończonej pracy wszystkie wyniki należy zapisać w tym folderze.

Operacje na bazie danych

Baza danych jest zgodna ze strukturą przedstawioną na `Ilustracji 1`.

Ilustracja 1. Baza danych



Za pomocą narzędzia `phpMyAdmin` wykonaj operacje na bazie danych:

Utwórz bazę danych o nazwie `notatki`, z zestawem polskich znaków (np. `utf8_unicode_ci`).

- Do bazy zaimportuj tabele z pliku `baza.sql` z rozpakowanego archiwum
- Wykonaj zrzut ekranu po imporcie. Zrzut zapisz w formacie `PNG` pod nazwą `import`. Nie kadruj zrzutu. Powinien on obejmować cały ekran monitora, z widocznym paskiem zadań. Na zrzucie powinny być widoczne elementy wskazujące na poprawnie wykonany import tabel.
- Wykonaj `zapytania SQL` działające na bazie notatki. Zapytania zapisz w pliku `kwerendy.txt`. Wykonaj zrzuty ekranu przedstawiające wyniki działania kwerend. Zrzuty zapisz w formacie `JPEG` i nadaj im nazwy `kw1`, `kw2`, `kw3`, `kw4`. Zrzuty powinny obejmować cały ekran monitora z widocznym paskiem zadań:
  - Zapytanie 1: wybierające jedynie najmniejszą `wartość priorytetu notatki` dla osoby o `id` równym `3.`
  - Zapytanie 2: wybierające jedynie `nazwę notatki` i jej `priorytet` dla notatek, których `nazwa` zawiera cząstkę „na”.
  - Zapytanie 3: wybierające jedynie `imię` osoby i odpowiadającą mu `nazwę notatki` dla notatek, których `priorytet` jest równy `5`. Należy posłużyć się `relacją.`
  - Zapytanie 4: `liczące` dla każdego imienia osoby `liczbę jego notatek`. Kwerenda wybiera jedynie `imię` i odpowiadającą mu `liczbę notatek`. Należy posłużyć się `relacją.`

Witryna internetowa:

Ilustracja 2. Witryna internetowa. Stan początkowy

Plik `obraz.jpg`, wypakowany z archiwum, należy przeskalować z zachowaniem proporcji do wysokości `610px`.

Cechy witryny:

- Składa się ze strony o nazwie `notatki.html`
- Zapisana w języku `HTML5`
- Zadeklarowany `polski` język zawartości witryny
- Jawnie zastosowany właściwy standard kodowania polskich znaków
- Tytuł strony widoczny na karcie przeglądarki: „Planer zadań”
- Arkusz stylów w pliku o nazwie `styl.css` prawidłowo połączony z kodem strony
- Podział strony na blok boczny i obok niego bloki: blok `nagłówkowy`, blok `dodający zadanie`, blok `główny`, blok `stopki`.
- Podział strony na bloki zrealizowany za pomocą `semantycznych znaczników` bloków języka `HTML5` tak, aby po uruchomieniu w przeglądarce układ bloków na stronie był zgodny z `Ilustracją 3`
- Zawartość bloku bocznego:
  - Obraz o nazwie `obraz.jpg` z tekstem alternatywnym „notatki”
- Zawartość bloku nagłówkowego:
  - Nagłówek `drugiego` stopnia o treści „Moje zadania”
- Zawartość bloku nawigacyjnego:
  - `Pole edycyjne` z podpowiedzią o treści „Dodaj zadanie...”
  - `Przycisk` „Dodaj”, którego wciśnięcie powoduje wywołanie funkcji w skrypcie
- Zawartość bloku głównego:
  - Lista `punktowana` (nieuporządkowana) zawierająca w stanie początkowym `6 elementów`. Każdy element listy zawiera:
    - Wpis (kolejne wpisy można skopiować do kodu HTML z pliku `zadania.txt`)

•Przycisk o treści „Wykonane”, którego wciśnięcie powoduje wywołanie skryptu dla danego elementu listy

•Zawartość stopki:

•Nagłówek `trzeciego` stopnia o treści: „Notatki: ”, dalej wstawiony `numer zdającego`.

Ilustracja 3. Układ bloków

Styl CSS witryny internetowej

Styl CSS zdefiniowany jest w całości w zewnętrznym pliku o nazwie `styl.css`. Cechy formatowania CSS, działające na stronie:

•Domyślne formatowanie wszystkich selektorów: krój czcionki `Verdana`

•Dla bloku bocznego: szerokość `20%`, wysokość `610 px`

•Dla obrazu: szerokość `100%`, wysokość `610 px`

•Dla bloku nagłówkowego, stopki i nawigacyjnego: kolor tła `MediumPurple`, `biały` kolor czcionki, szerokość `80%`, wysokość `50px`, wyrównanie tekstu `do środka`, cień tekstu o przesunięciu `4px` w obu osiach i kolorze `DimGray`

•Dodatkowo dla bloku nawigacyjnego: margines wewnętrzny górny `10px`

•Dla bloku głównego: kolor tła `Thistle`, szerokość `80%`, wysokość `450px`, paski przewijania dodawane tylko, gdy zawartość nie mieści się w bloku

•Dla przycisków w liście: opływanie `do prawej strony`, kolor tła `MediumPurple`, `biały` kolor czcionki, wysokość `30 px`, bez obramowania

•Dla listy punktowanej: marginesy zewnętrzne górny i dolny `40 px`, lewy i prawy automatycznie wyliczane przez przeglądarkę, szerokość `70%`, rozmiar czcionki `25px`, obramowanie linią `ciągłą` o szerokości `2px` i kolorze `MediumPurple`, lista nie ma znaków punktora.

•Dla elementu listy: marginesy wewnętrzne `10px`

•Dla parzystych elementów listy: kolor tła `#DDCADD`

•Gdy kursor znajduje się na elemencie listy, jego kolor tła zmienia się na `Lavender`

Skrypt

Wymagania dotyczące skryptu:

•Napisany w języku `JavaScript` i działający na liście punktowanej

•Należy stosować znaczące nazewnictwo zmiennych i funkcji w języku polskim lub angielskim

•Dla uproszczenia należy założyć, że w stanie początkowym jest 6 elementów listy, elementy mogą być dodawane, ale nie są usuwane

•Działanie funkcji wywoływanej po wciśnięciu dowolnego z przycisków „Wykonane”:

•Przekreśla treść elementu listy związanego z tym przyciskiem. Na `Ilustracji 4` pokazano efekt wciśnięcia przycisku „Wykonane” w drugim elemencie listy. W wyniku tego tekst „Gimnastyka” został przekreślony

•Działanie funkcji wywoływanej po wciśnięciu przycisku „Dodaj”:

•Tworzy nowy element listy na jej końcu. Element składa się z:

•Treści pobranej z `pola edycyjnego`

•Przycisku o treści „Wykonane”, który jest formatowany tak jak reszta przycisków w liście i jego kliknięcie powoduje wywołanie odpowiedniej funkcji

•Na `Ilustracji 4` pokazano efekt wciśnięcia przycisku Dodaj. Został utworzony nowy element listy „Lekarz” wraz z przyciskiem „Wykonane”. Wybranie utworzonego przycisku spowoduje wykonanie skryptu dla elementu listy „Lekarz”

Ilustracja 4. Działanie aplikacji

Tabela 1. Wybrane pola i metody modelu DOM języka JavaScript

Wyszukiwanie elementów
document.getElementById(Id)
document.getElementByTagName(TagName)
document.getElementsByClassName(ClassName)
document.getElementsByName(ElementName)
document.querySelector(CSSselector)
document.querySelectorAll(CSSselector)

Zmiana elementów
element.innerHTML = "nowa zawartość"
element.attribute_name = "nowa wartość"
element.setAttribute(atrybut,wartość)
element.style.property_name = "nowa wartość"

Operacje na elementach dokumentu
document.createElement(element)
document.removeChild(element)
document.appendChild(element)
document.replaceChild(element)
document.write(text)

Wybrane właściwości obiektu style
color
fontSize
fontStyle = "normal | italic | oblique | initial | inherit"
fontWeight = "normal | lighter | bold | bolder | value | initial | inherit"
listStyleType = "circle| decimal | disc | none | square | initial..."

Wybrane zdarzenia HTML
Zdarzenia myszy: onclick, ondblclick, onmouseover, onmouseout
Zdarzenia klawiatury: onkeydown, onkeypress, onkeyup
Zdarzenia obiektów: onload, onresize, onfocusin, onfocusout

Elementy formularzy
Ważniejsze typy pola input: button, checkbox, number, password, radio, text, range
Inne elementy: select, textarea

Metody i pola obiektu string (JS)
length
indexOf(text)
search(text)
substr(startIndex, endIndex)
replace(textToReplace, newText)
toUpperCase()

Metody i pola obiektu string (JS)
toLowerCase()

Tabela 2. Cień elementu i tekstu w CSS

The box-shadow property attaches one or more shadows to an element.

```
box-shadow: h-offset v-offset blur spread color;
```

The text-shadow property adds shadow to text.

```
text-shadow: h-shadow v-shadow blur-radius color;
```

Tabela 3. Tworzenie elementów w JavaScript

Example

Create a <p> element and append it to the document:

```
const para = document.createElement("p");
para.innerText = "This is a paragraph";
para.className = "nazwaKlasyCSS";
document.body.appendChild(para);
```

Example

Append an item to a list:

```
const node = document.createElement("li");
/* add text, classes and attributes */
document.getElementById("idListy").appendChild(node);
```

Tabela 4. Elementy semantyczne w HTML

Tag	Description
<article>	Defines independent, self-contained content
<aside>	Defines content aside from the page content
<details>	Defines additional details that the user can view or hide
<figcaption>	Defines a caption for a <figure> element
<figure>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<footer>	Defines a footer for a document or section
<header>	Specifies a header for a document or section
<main>	Specifies the main content of a document
<mark>	Defines marked/highlighted text
<nav>	Defines navigation links
<section>	Defines a section in a document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time

UWAGA!

Po zakończeniu pracy utwórz plik tekstowy o nazwie przeglądarka.txt. Zapisz w nim nazwę przeglądarki internetowej, w której weryfikowana była poprawność działania witryny. Umieść go w folderze z numerem zdającego.

Nagraj płytę z rezultatami pracy. W folderze z numerem zdającego, powinny znajdować się pliki: import.png, kw1.jpg, kw2.jpg, kw3.jpg, kw4.jpg, kwerendy.txt, notatki.html, obraz.jpg, przeglądarka.txt, styl.css, ewentualnie inne przygotowane pliki.
Po nagraniu płyty sprawdź poprawność jej odczytu. Opisz płytę numerem zdającego, którym został podpisany arkusz i pozostaw zapakowaną w pudełku na stanowisku wraz z arkuszem egzaminacyjnym.

Czas przeznaczony na wykonanie zadania wynosi 150 minut.

Ocenie będzie podlegać 5 rezultatów:

- operacje na bazie danych
- zawartość witryny internetowej
- działanie witryny internetowej
- styl CSS witryny internetowej
- skrypt.

Modyfikacja na potrzeby lekcji:

1. W miejsce numeru zdającego wstaw swoje imię.
2. Utwórz plik tekstowy grafiki.txt a w nim umieść pliki z grafiką: import.png, kw1.jpg, kw2.jpg, kw3.jpg, kw4.jpg, obraz.jpg.
3. Zamiast nagrywania płyty prześlij pliki z rezultatami na swój gitlab (github). W tym celu utwórz w swoim repozytorium w folderze szkola nowy folder Zadanie1 i w nim umieść przesyłane pliki.
4. W folderze powinny znajdować się pliki: grafilki.txt, kwerendy.txt, notatki.html,, przeglądarka.txt, styl.css, ewentualnie inne przygotowane pliki.
5. Sprawdź czy w przesłanym folderze znajdują się wszystkie wymagane pliki.
6. Nie kompresuj przesyłanych plików.
7. Prześlij tyle plików i w takiej formie, ile zdążysz wykonać na lekcji – będziemy to kontynuowali na kolejnych zajęciach.
8. Brak plików w repozytorium gdy będę to sprawdzał oznacza ocenę ndst za aktywność na lekcji.
9. Wszelkie niejasności wyjaśniam na lekcji.