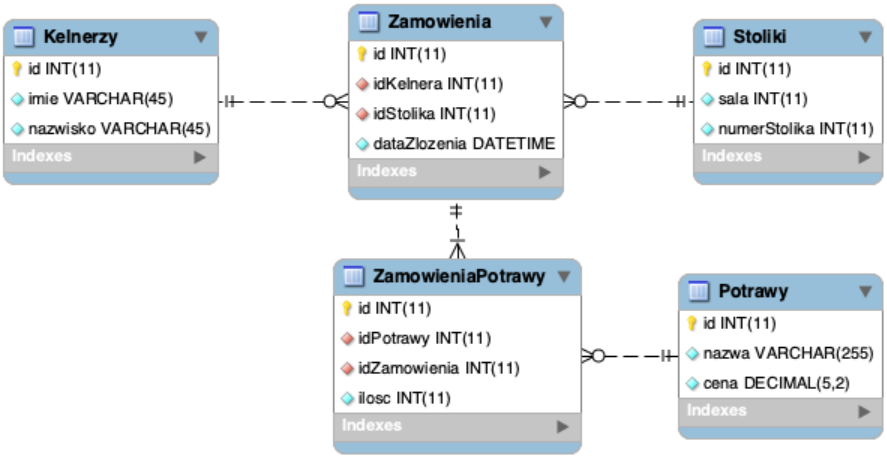
Zadanie egzaminacyjne

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

Wykonaj aplikację internetową dla restauracji, wykorzystując zainstalowane na komputerze aplikacje.

Aby wykonać zadanie, należy zalogować się na konto **Egzamin** bez hasła. Na pulpicie znajduje się archiwum 7z o nazwie *pliki19* zabezpieczone hasłem: **ReStAuRac1a**
Archiwum należy rozpakować. Na pulpicie konta **Egzamin** należy utworzyć folder. Jako nazwy folderu należy użyć numeru zdającego. Wszystkie wyniki należy zapisać w tym folderze.

Operacje na bazie danych
Do wykonania operacji na bazie należy wykorzystać tabele przedstawione na ilustracji 1.



Ilustracja 1. Baza danych

Za pomocą narzędzia phpMyAdmin wykonaj następujące operacje na bazie danych:

- Utwórz bazę danych o nazwie *restauracja*, z zestawem polskich znaków (np. utf8_polish_ci)
- Do utworzonej bazy zaimportuj tabele z pliku *restauracja.sql*
- Wykonaj zrzut ekranu po imporcie. Zapisz zrzut 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 *restauracja*. Zapytania zapisz w pliku *kwerendy.txt*. Wykonaj zrzuty ekranu przedstawiające wyniki działania kwerend. Zrzuty zapisz w formacie PNG i nadaj im nazwy *kw1*, *kw2*, *kw3*, *kw4, kw5*. Zrzuty powinny obejmować cały ekran monitora z widocznym paskiem zadań
  - Zapytanie 1: wybierające jedynie imiona i nazwiska kelnerów posortowane malejąco według nazwiska
  - Zapytanie 2: zliczające jedynie liczbę zamówień, które zostały złożone dla stolika numer 3 w sali numer 1. Należy posłużyć się relacją
  - Zapytanie 3: wybierające jedynie nazwy potraw, które zostały zamówione dnia 9 czerwca 2024. Należy posłużyć się relacją

- Zapytanie 4: tworzące nowego użytkownika lokalnego o nazwie **administrator** i haśle **boss123!**
- Zapytanie 5: zmieniające strukturę tabeli *Potrawy* poprzez zmianę typu kolumny *cena* w taki sposób, aby przechowywała ona liczby rzeczywiste składające się z maksimum 4 cyfr w części całkowitej (przed przecinkiem) i 2 cyfr w części ułamkowej (po przecinku). Typ ten powinien przechowywać dane w zapisie stałoprzecinkowym. Należy zachować wszystkie dotychczasowe ograniczenia nałożone na kolumnę

Witryna internetowa

Składa się z pliku *index.html*. Jej wygląd przedstawiono na ilustracji 2. Układ bloków z potrawami może się różnić w zależności od szerokości okna przeglądarki.



Ilustracja 2. Wygląd strony internetowej

Cechy strony *index.html*:

- Zapisana w języku HTML5
- Zadeklarowany polski język zawartości witryny
- Jawnie zastosowany właściwy standard kodowania polskich znaków
- Tytuł strony widoczny na karcie przeglądarki: „Restauracja Marchewka”
- Arkusz stylów w pliku o nazwie *styl.css* prawidłowo połączony z kodem strony



Ilustracja 3. Układ bloków

- Podział strony na bloki: blok nagłówkowy zawierający dwie sekcje, blok główny zawierający dwie sekcje, blok stopki. Podział zrealizowany za pomocą semantycznych znaczników języka HTML5 tak, aby po uruchomieniu w przeglądarce układ bloków był zgodny z ilustracją 3

- Zawartość lewej sekcji nagłówkowej:
  - Nagłówek pierwszego stopnia o treści „Restauracja Marchewka”
  - Nagłówek trzeciego stopnia o treści „Twoje wegetariańskie miejsce na mapie miasta”
- Zawartość prawej sekcji nagłówkowej: obraz *logo.png* z tekstem alternatywnym „logo”
- Zawartość lewej sekcji głównej:
  - Nagłówek stopnia czwartego o treści „Twoje zamówienie”
  - Lista punktowana (nieuporządkowana) - lista w stanie początkowym jest pusta
  - Przycisk o treści „Wyczyść”, którego kliknięcie powoduje wykonanie skryptu 2
  - Przycisk o treści „Zatwierdź”, którego kliknięcie powoduje wykonanie skryptu 3
- Zawartość prawej sekcji głównej: 6 bloków z potrawami zawierających obraz oraz paragraf. Kolejne bloki zawierają odpowiednio:
  - Obraz *nalesniki.webp* z tekstem alternatywnym „Naleśniki” i paragraf o treści „Naleśniki”
  - Obraz *lasagne.webp* z tekstem alternatywnym „Lasagne szpinakowa” i paragraf o treści „Lasagne szpinakowa”
  - Obraz *marchewka.webp* z tekstem alternatywnym „Pieczona marchewka” i paragraf o treści „Pieczona marchewka”
  - Obraz *rosol.webp* z tekstem alternatywnym „Wege rosół” i paragraf o treści „Wege rosół”
  - Obraz *tarta.webp* z tekstem alternatywnym „Tarta szparagowa” i paragraf o treści „Tarta szparagowa”
  - Obraz *kompot.webp* z tekstem alternatywnym „Kompot” i paragraf o treści „Kompot”
  - Kliknięcie każdego z bloków powoduje wykonanie skryptu 1
- Zawartość stopki: paragraf o treści „Wykonał:”, dalej wstawiony numer zdającego

Styl CSS witryny internetowej

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

- Domyślnie, dla wszystkich selektorów: krój czcionki „Verdana”, zaokrąglenie obramowania 5 px
- Dla selektora ciała strony: kolor tła #D1E2C4
- Dla lewej sekcji nagłówkowej: szerokość 80%, kolor tła #778A35, wysokość 200 px
- Dla prawej sekcji nagłówkowej: szerokość 20%, kolor tła #778A35, wysokość 200 px
- Dla nagłówka pierwszego stopnia: rozmiar czcionki 50 px, margines wewnętrzny lewy 200 px
- Dla nagłówka trzeciego stopnia: rozmiar czcionki 30 px, czcionka pochylona, margines wewnętrzny lewy 200 px
- Dla lewej sekcji głównej: kolor tła #EBEBE8, szerokość 25%
- Dla prawej sekcji głównej: szerokość 75%
- Dla bloków z potrawami: kolor tła #778A35, tekst wyrównany do środka, marginesy zewnętrzne 20 px, wyświetlanie elementów blokowych obok siebie zgodnie z ilustracją 2
- Dla selektora obrazu znajdującego się w bloku z potrawą: marginesy zewnętrzne 15 px, wysokość 150 px
- Gdy kursor znajdzie się na bloku z potrawą pojawia się obrys (outline) linią ciągłą, o szerokości 2 px i kolorze białym
- Dla stopki: kolor tła #778A35, margines zewnętrzny górny 10 px, marginesy wewnętrzne 10 px, wyrównanie tekstu do prawej strony
- Dla przycisków: kolor tła #778A35, marginesy wewnętrzne 5px

Uwaga: styl CSS ciała strony, obrazu znajdującego się w bloku z potrawą należy zdefiniować wyłącznie przy pomocy selektorów tych znaczników. Jest to uwarunkowane projektem późniejszej rozbudowy witryny.

Skrypty działające po stronie klienta

Wymagania wspólne dotyczące skryptów:

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

Skrypt 1

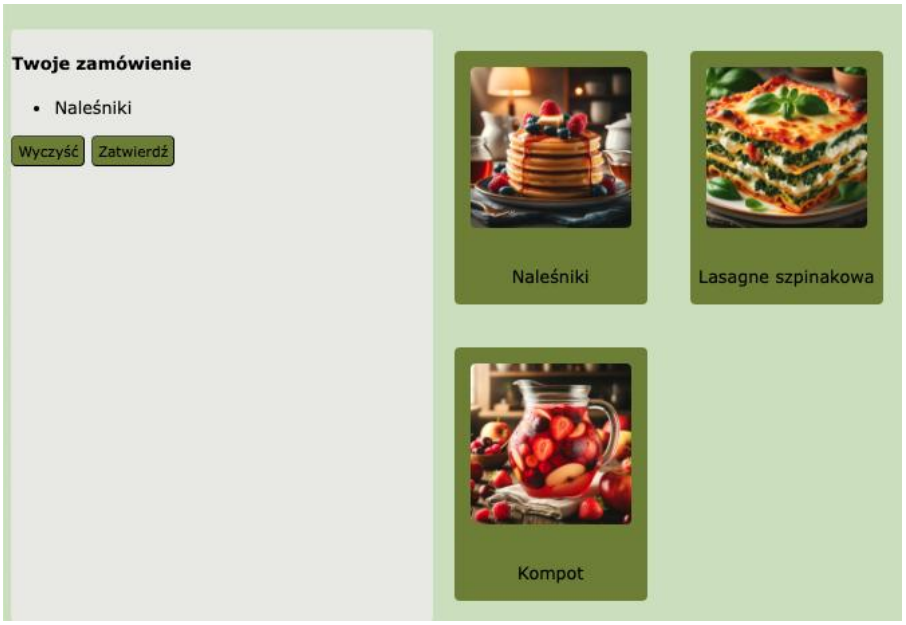
- Wywoływany jest po kliknięciu na blok z potrawą
- Dodaje nowy element do listy w lewej sekcji głównej. Element ten zawiera nazwę kliknietej potrawy (ilustracja 4)

Skrypt 2:

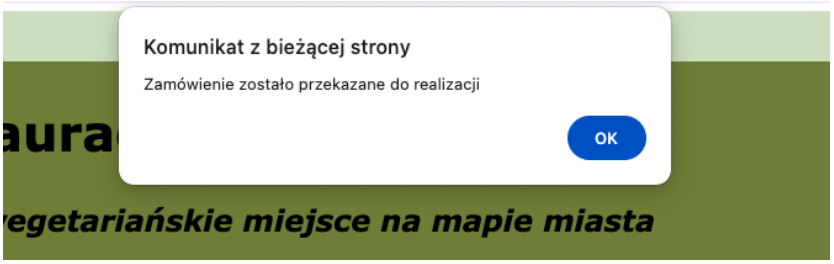
- Wywoływany jest po kliknięciu na przycisk Wyczyść
- Czyści zawartość listy lewej sekcji głównej

Skrypt 3:

- Wywoływany jest po kliknięciu na przycisk Zatwierdź
- Na stronie pojawia się alert z informacją „Zamówienie zostało przekazane do realizacji” (ilustracja 5)
- Zawartość listy z lewej sekcji głównej jest czyszczona



Ilustracja 4. Działanie skryptu 1. Po kliknięciu na blok z potrawą „naleśniki” nazwa potrawy została dodana do listy



Ilustracja 5. Działanie skryptu 3

Tabela 1. Wybrane właściwości modelu flexbox. Pogrubienie oznacza wartość domyślną, znak pionowej kreski oddziela dostępne wartości atrybutów

Właściwość CSS	**Znaczenie**
display: flex;	Element staje się blokowym kontenerem flexbox.
flex-wrap: **nowrap** | wrap | wrap-reverse | initial | inherit;	Określa, czy elementy zawarte w kontenerze flexbox powinny być zawijane do nowych linii, czy też nie.
flex-direction: **row** | row-reverse | column | column-reverse | initial | inherit;	Określa kolejność zawijania elementów.
justify-content: **flex-start** | flex-end | center | space-between | space-around | space-evenly | initial | inherit;	Wyrównuje elementy kontenera flexbox, gdy nie zajmują one całej dostępnej przestrzeni w osi głównej (poziomej).
align-items: **normal** | stretch | positional alignment | flex-start | flex-end | baseline | initial | inherit;	Określa domyślne wyrównanie elementów w kontenerze flexbox lub grid. W kontenerze flexboxa elementy są wyrównywane w osi pionowej.

Tabela 2. Wybrane pola i metody modelu DOM języka JavaScript

Wyszukiwanie elementów	**Zmiana elementów**
document.getElementById(*id*)	element.innerHTML = "*wartość*"
document.getElementsByTagName(*TagName*)	element.*attribute_name* = "*wartość*"
document.getElementsByClassName(*ClassName*)	element.setAttribute(*atrybut*, *wartosc*)
document.getElementsByName(*ElementName*)	element.style.*property_name* = "wartość"
document.querySelector(*CSSselector*)	
document.querySelectorAll(*CSSselector*)	

Operacje na elementach dokumentu	**Wybrane właściwości obiektu style**
document.createElement(*element*)	backgroundColor
document.removeChild(*element*)	color
document.appendChild(*element*)	fontSize
document.replaceChild(*element*)	fontStyle = "normal | italic | oblique | initial | inherit"
document.write(*text*)	fontWeight = "normal | lighter | bold | bolder | value | initial | inherit"
	listStyleType = "circle| decimal | disc | none | square | initial..."

Wybrane zdarzenia HTML

Zdarzenia myszy	**Zdarzenia klawiatury**	**Zdarzenia obiektów**
onclick	onkeydown	onload
ondblclick	onkeypress	onresize
onmouseover	onkeyup	onfocusin
onmouseout		onfocusout

Elementy formularzy	Metody i pola obiektu string (JS)
Ważniejsze typy pola input: button, checkbox, number, password, radio, text, range, file **Inne elementy:** select, textarea	Length indexOf(*text*) search(*text*) lastIndexOf() substr(*startIndex, endIndex*) replace(*textToReplace, newText*) toUpperCase() toLowerCase()

Tabela 3. Semantic Elements in 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: index.html, styl.css, kwerendy.txt, przeglądarka.txt, zrzuty ekranu: import.png, kw1.png, kw2.png, kw3.png, kw4.png, kw5.png, 6 plików formatu WEBP, logo.png, ewentualnie inne przygotowane pliki. Po nagraniu płyty sprawdź poprawność jej odczytu. Opisz płytę numerem zdającego i pozostaw zapakowaną w pudełku na stanowisku wraz z arkuszem egzaminacyjnym.

Czas przeznaczony na wykonanie zadania wynosi 150 minut.

Ocenie podlegać będzie 5 rezultatów:

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