

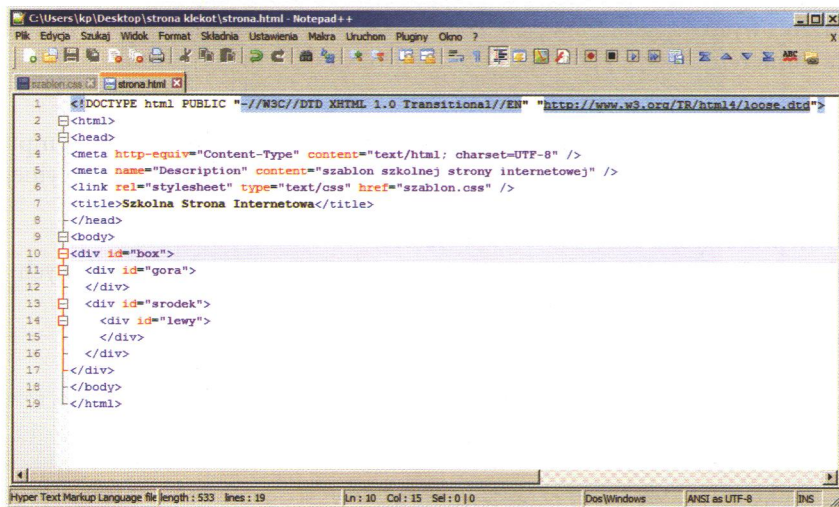
ROZWIĄZANIE ZADANIA III

1. Utworzenie szablonu strony internetowej składającego się z pliku strona.html zawierającego szablon podziału strony na określone bloki oraz szablon.css będącego zewnętrznym arkuszem stylu formatowania poszczególnych bloków strony.

Te zagadnienia dotyczą efektów kształcenia E.14.1.1, E.14.1.2, E.14.1.3, E.14.1.4, E.14.1.8, E.14.1.11. Materiał niezbędny do wykonania tego zadania znajduje się w rozdziałach 1., 3. – 5., 10., 13., 15.–19. podręcznika *Witryny internetowe* Wydawnictw Szkolnych i Pedagogicznych (Warszawa 2013).

Kaskadowe arkusze stylów (*Cascading Style Sheets*) to język służący do opisu formy wyświetlania stron WWW. Arkusz stylów CSS to lista reguł ustalających, w jaki sposób przeglądarka internetowa ma wyświetlić zawartość wybranego elementu (lub elementów) języka HTML lub XML. W ten sposób można opisać pojęcia odpowiedzialne za prezentację elementów dokumentów internetowych, takie jak rodzina czcionek, kolor tekstu, marginesy, odstęp międzywierszowy, pozycja danego elementu względem innych elementów bądź okna przeglądarki. CSS pozwala odseparować strukturę dokumentu od formy jego prezentacji, co upraszcza dokument i ułatwia wprowadzanie w nim zmian. Zewnętrzne arkusze CSS mogą być wspólne dla wielu dokumentów, co daje możliwość zmiany wyglądu wielu stron jednocześnie bez ingerowania w sam kod HTML.

Zawartość strony została opisana w dwóch dokumentach. Dokument strona.html opisuje układ oraz treść strony. Zawartość pliku strona.html została pokazana na rys. 3.2.

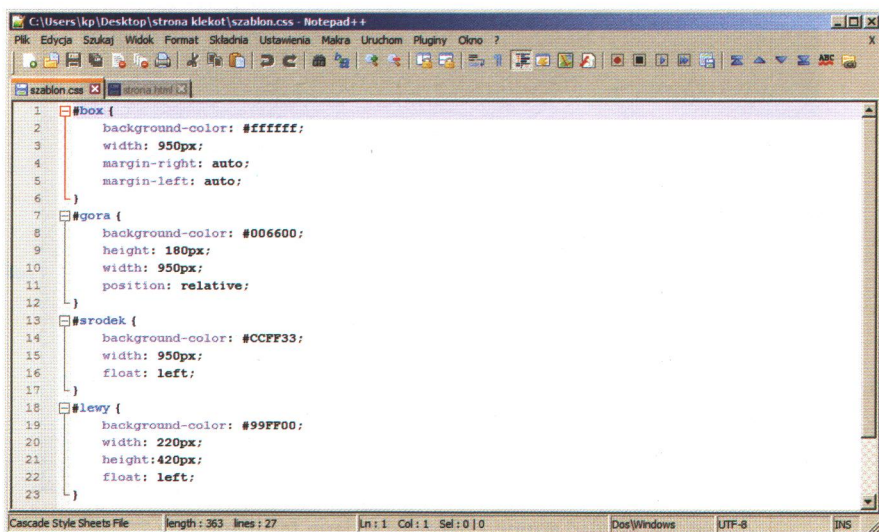


```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
5 <meta name="Description" content="szablon szkolnej strony internetowej" />
6 <link rel="stylesheet" type="text/css" href="szablon.css" />
7 <title>Szkolna Strona Internetowa</title>
8 </head>
9 <body>
10 <div id="box">
11 <div id="gora">
12 </div>
13 <div id="srodek">
14 <div id="lewy">
15 </div>
16 </div>
17 </div>
18 </body>
19 </html>
```

Rys. 3.2. Zawartość pliku strona.html

W nagłówku pliku, oprócz instrukcji opisujących sposób kodowania znaków oraz tytułu, została zapisana instrukcja `<link rel="Stylesheet" type="text/css" href="szablon.css" />`. Instrukcja ta informuje przeglądarkę o nazwie zewnętrznego pliku stylów kaskadowych, w którym znajdują się reguły opisujące sposób wyświetlania strony. W wierszu 10 znacznik `<div>` rozpoczyna definicję bloku (box), w którym będą się znajdować pozostałe bloki strony. Blok (box) został podzielony na dwie części o nazwach gora (opisany w wierszach 11 i 12) oraz srodek (opisany w wierszach od 13 do 16). Ponieważ szerokości bloków gora i srodek są równe szerokości bloku box, to są tworzone dwa wiersze (podział pionowy bloku box). Ponadto w bloku srodek utworzono dodatkowy blok o nazwie lewy (opisany w wierszach 13 i 14). Blok ten ma szerokość 220 punktów i jest wyrównywany do lewej strony, co powoduje podział bloku srodek na dwie kolumny (podział poziomy).

W pliku szablon.css został opisany sposób formatowania poszczególnych fragmentów dokumentu. Zawartość pliku jest pokazana na rys. 3.3.



Rys. 3.3. Zawartość pliku szablon.css

W wierszach od 1 do 6 został zdefiniowany sposób wyświetlania dla bloku z identyfikatorem box. Poszczególne wiersze oznaczają:

- background-color: #ffffff; – kolor tła: biały;
- width: 950px; – szerokość bloku: 950 punktów;
- margin-left: auto; – margines lewy: wybór automatyczny;
- margin-right: auto; – margines prawy: wybór automatyczny.

W wierszach od 7 do 12 został zdefiniowany sposób wyświetlania dla bloku z identyfikatorem gora. Poszczególne wiersze oznaczają:

- background-color: #006600; – kolor tła: #006600 (zapis szesnastkowy);
- height: 180px; – wysokość bloku: 180 punktów;
- width: 950px; – szerokość bloku: 950 punktów;
- position: relative; – pozycjonowanie względne.

W wierszach od 13 do 17 został zdefiniowany sposób wyświetlania dla bloku z identyfikatorem srodek. Poszczególne wiersze oznaczają:

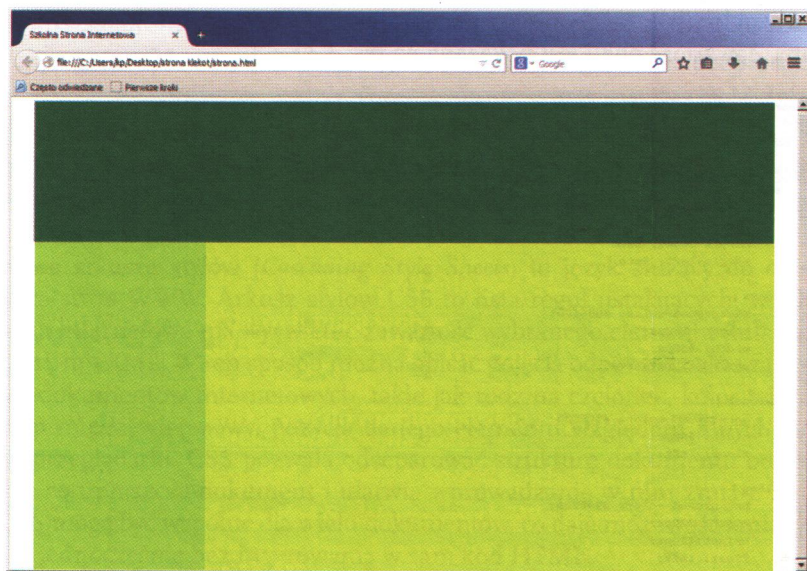
- background-color: #CCFF33; – kolor tła: #CCFF33 (zapis szesnastkowy);

- width: 950px; – szerokość 950 punktów;
- float: left; – wyrównywanie bloku do lewej.

W wierszach od 18 do 23 jest zdefiniowany sposób wyświetlania dla bloku z identyfikatorem lewy. Poszczególne wiersze oznaczają:

- background-color: #99FF00; – kolor tła: #99FF00 (zapis szesnastkowy);
- width: 220px; – szerokość bloku 220 punktów;
- height: 420px; – wysokość bloku 420 punktów;
- float: left; – wyrównywanie bloku do lewej.

Poszczególne bloki w dokumencie są rozróżniane za pomocą identyfikatorów wprowadzanych w opisie. Wygląd strony z różnymi stylami formatowania został pokazany na rys 3.4.



Rys. 3.4. Wygląd strony z różnymi stylami formatowania

Gotowe pliki (strona.html i szablon.css) należy zapisać na dysku C:\ w folderze o nazwie zgodnej z twoim numerem PESEL.

2. Utworzenie skryptu z formularzem wysłanym na adres e-mailowy adam@wp.pl z wcześniejszą weryfikacją wprowadzonych danych.

Te zagadnienia dotyczą efektów kształcenia E.14.1.1, E.14.1.2, E.14.1.9, E.14.3.1, E.14.3.3, E.14.3.4, E.14.3.5, E.14.3.9. Materiał niezbędny do wykonania tego zadania znajduje się w rozdziałach 1., 3.–5., 10. i 12. podręcznika *Witryny internetowe* Wydawnictw Szkolnych i Pedagogicznych (Warszawa 2013) oraz rozdziałach 1., 3. – 7., 9., 12. podręcznika *Aplikacje internetowe* Wydawnictw Szkolnych i Pedagogicznych (Warszawa 2013).

W formularzach wypełnianych przez internautów mogą występować błędne dane lub niewypełnione pola formularza. Literówki, niezaznaczone wymagane pola czy wprowadzone niewłaściwe wartości mogą zakłócić działanie serwisu, uszkodzić bazę danych lub nawet udostępnić poufne informacje osobom niepowołanym. W celu uniknięcia kłopotów należy wszelkie dane pochodzące od użytkownika poddać dokładnej analizie i sprawdzeniu pod kątem kompletności i poprawności. Walidację formularzy przeprowadza się za pomocą skryptów. Skrypty mogą być wykonywane po stronie serwera, np. skrypty w języku

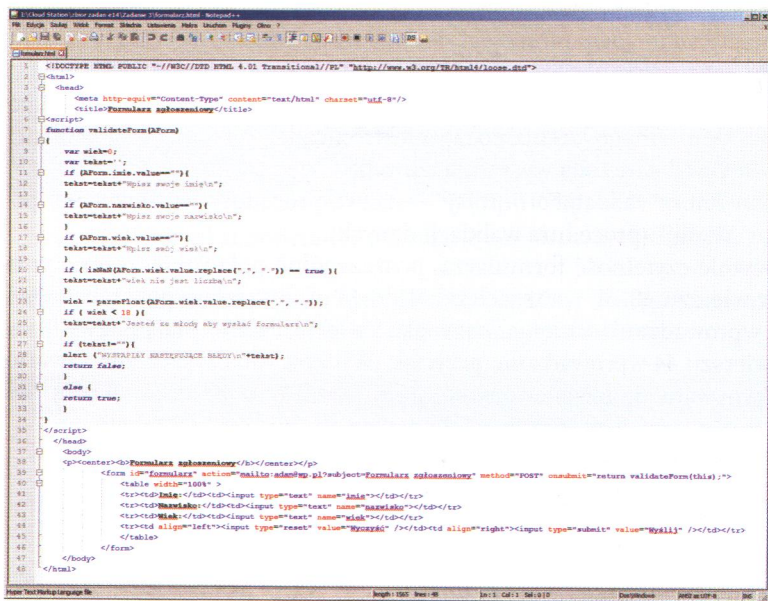
PHP lub po stronie przeglądarki, np. skrypty w języku JavaScript. Weryfikacja wprowadzanych danych może być prowadzona:

- przed wysłaniem danych (skrypty wykonywane po stronie przeglądarki);
- po wysłaniu danych do serwera (skrypty wykonywane po stronie serwera).

Walidacja danych przez przeglądarkę jest realizowana zazwyczaj w języku JavaScript. Przeglądarka nie komunikuje się z serwerem, dzięki czemu unika się sytuacji, w której użytkownik wypełnia formularz, przesyła go na serwer i czeka pewien czas tylko po to, by zobaczyć komunikat informujący o błędach. Wadą walidacji danych przez przeglądarkę jest możliwość obejścia zabezpieczeń.

Walidacja danych po stronie serwera powinna być wykonywana przed rozpoczęciem procesu przetwarzania danych. Wszelkie dane pochodzące od użytkownika, np. otrzymane za pomocą formularzy, należy w skrypcie traktować jako potencjalnie niebezpieczne. Kod analizujący poprawność danych, np. w języku PHP jest wykonywany przez serwer. Tylko sprawdzenie faktycznie odebranych danych w skrypcie PHP można traktować jako pewne zabezpieczenie serwisu.

Kod źródłowy strony ze skryptem pokazano na rys. 3.5.



Rys. 3.5. Kod źródłowy strony ze skryptem do walidacji danych

W nagłówku strony umieszczono informacje o języku strony, sposobie kodowania znaków i tytule strony. Ponadto w nagłówku został umieszczony kod języka JavaScript wykorzystywany do walidacji danych. Kod ten rozpoczyna się w wierszu 6 znacznikiem `<script>` i kończy w wierszu 35 znacznikiem `</script>`. W wierszu 9 jest umieszczona deklaracja zmiennej `wiek` (jednocześnie przypisano jej wartość początkową 0), wykorzystywanej do przechowywania wieku wypełniającego formularz. W wierszu 10 została umieszczona deklaracja zmiennej `tekst` (jednocześnie przypisano jej tekst pusty), wykorzystywanej do wyświetlania informacji o błędach. **Instrukcja IF** w wierszach od 11 do 13 sprawdza, czy pole imie formularza zostało wypełnione. Jeżeli pole to pozostało puste to do zmiennej `tekst` dopisywany jest komunikat „Wpisz swoje imię” i symbol przejścia do nowej linii. Podobnie instrukcja IF zapisana w wierszach od 14 do 16 sprawdza pole nazwisko.

a instrukcja IF w wierszach od 17 do 19 sprawdza wiek. **Instrukcja IF** w wierszu od 20 do 22 sprawdza, czy wartość wprowadzona w pole wiek jest liczbą. Ponieważ użytkownik może wpisać wiek jako liczbę ułamkową z przecinkiem lub kropką, to funkcja `replace(„”, „.”)` zamienia przecinek lub przecinek ze spacją na kropkę. Jeżeli wprowadzonej danej w pole wiek nie uda się zamienić na liczbę, to do zmiennej tekst jest dopisywany tekst „wiek nie jest liczbą”. W wierszu 23 do zmiennej wiek jest przypisywana, zamieniona na liczbę typu Float, wartość wprowadzona do pola wiek formularza. **Instrukcja IF** w wierszu od 24 do 26 sprawdza, czy wartość zmiennej wiek jest mniejsza niż 18. Jeżeli tak, to do zmiennej tekst jest dopisywany tekst „Jesteś za młody, aby wysłać formularz”. **Instrukcja IF** w wierszu od 27 do 33 sprawdza, czy zmienna tekst jest tekstem pustym. Jeżeli nie, to jest wyświetlana jej zawartość (wszystkie komunikaty dopisywane podczas walidacji danych) oraz zwracana wartość false, oznaczająca, że walidacja nie zakończyła się sukcesem. Jeżeli zmienna tekst jest tekstem pustym, to jest zwracana wartość true oznaczająca, że walidacja zakończyła się sukcesem.

Kod formularza zaczyna się w wierszu 39, a kończy w wierszu 46. W wierszu 39 określono parametry formularza:

- **id="formularz"** – identyfikator formularza;
- **action=mailto:adam@wp.pl?subject=Formularz zgłoszeniowy** – akcja podejmowana po kliknięciu przycisku „Wyślij”. W tym przypadku dane zebrane w formularzu zostaną wysłane na adres `adam@wp.pl`, w wiadomości zatytułowanej „Formularz zgłoszeniowy”, za pomocą domyślnego klienta poczty elektronicznej;
- **method="POST"** – metoda wysyłania danych;
- **onsubmit="return validateForm(this)"** – nazwa procedury wykonywanej po kliknięciu przycisku „Wyślij” (procedura walidacji danych).

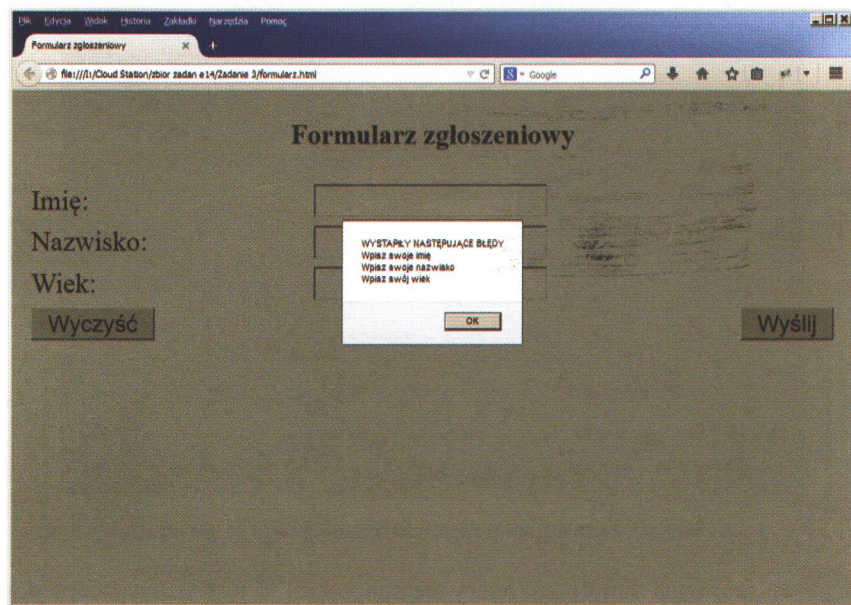
Aby zapewnić czytelność formularza, poszczególne pola umieszczono w tabeli zdefiniowanej w wierszu 40. W wierszach od 41 do 43 umieszczono pola tekstowe formularza służące do wprowadzania imienia, nazwiska i wieku. Pola te mają nazwy `imie`, `nazwisko` i `wiek`. W wierszu 44 wprowadzono przycisk „Wyczyść” do czyszczenia zawartości formularza (wyrównywany do lewej strony komórki tabeli) oraz przycisk „Wyślij” do wysyłania zawartości formularza (wyrównywany do prawej strony komórki tabeli).

Na rys. 3.6. pokazano wygląd formularza do wprowadzania danych.

The image shows a web browser window with a form titled "Formularz zgłoszeniowy". The form is displayed on a white background. It contains three text input fields, each preceded by a label: "Imię:", "Nazwisko:", and "Wiek:". Below these fields are two buttons: "Wyczyść" (left) and "Wyślij" (right). The browser's address bar shows the file path "file:///C:/Users/zadane14/Zadanie 3/formularz.html".

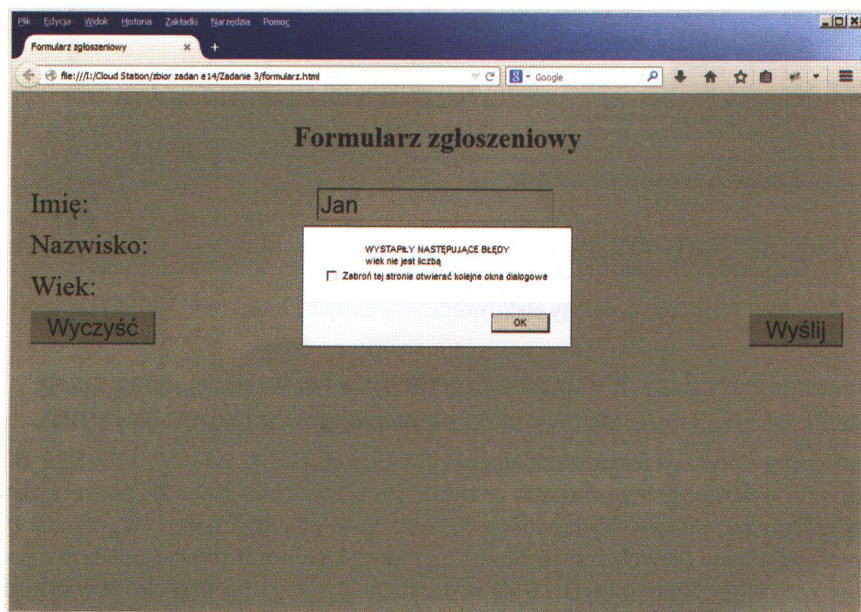
Rys. 3.6. Formularz do wprowadzania danych

W przypadku gdy użytkownik nie wprowadzi żadnych danych do formularza i kliknie przycisk „Wyślij”, wyświetla się komunikat zawierający informacje o braku danych w polach (rys. 3.7).



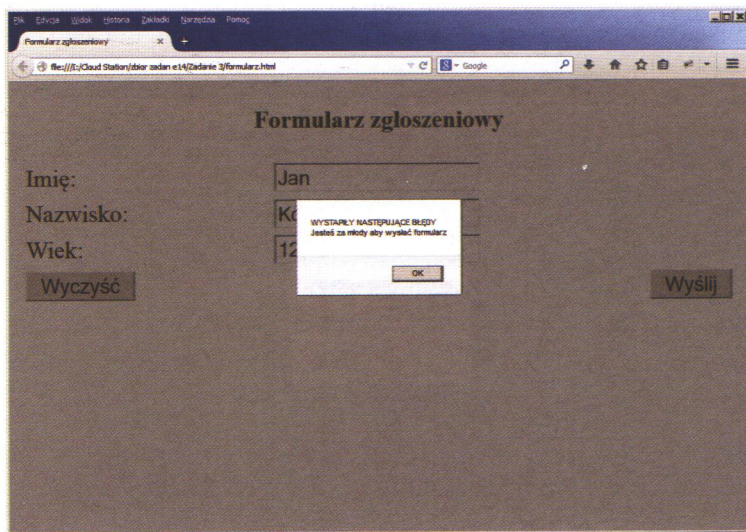
Rys. 3.7. Komunikat zawierający informacje o braku danych w polach

Jeżeli użytkownik wprowadzi do pola wiek dane, których nie można zamienić na liczbę, to generuje się komunikat o niewłaściwym formacie danych (rys. 3.8).



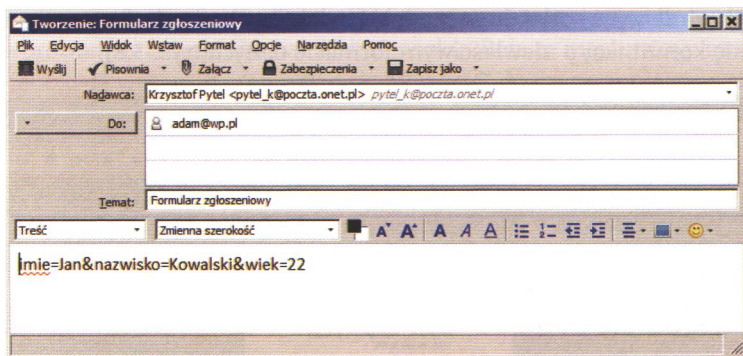
Rys. 3.8. Komunikat o niewłaściwym formacie danych

Jeżeli wpisany w pole wiek ma wartość mniejszą niż 18, to generuje się komunikat o niespełnieniu kryterium wieku (rys. 3.9).



Rys. 3.9. Komunikat o niespełnieniu kryterium wieku

Jeżeli proces walidacji danych zakończy się sukcesem, to jest uruchamiany domyślny klient poczty elektronicznej ze wstępnie wypełnionymi polami adres i temat. W zawartości wiadomości są umieszczane dane zebrane w formularzu (rys. 3.10).



Rys. 3.10. Wiadomość poczty elektronicznej z danymi zebranymi w formularzu

3. Utworzenie bazy danych o nazwie przychodnia zawierającej jedną tabelę o nazwie lekarze znormalizowaną do pierwszej postaci normalnej (1PN).

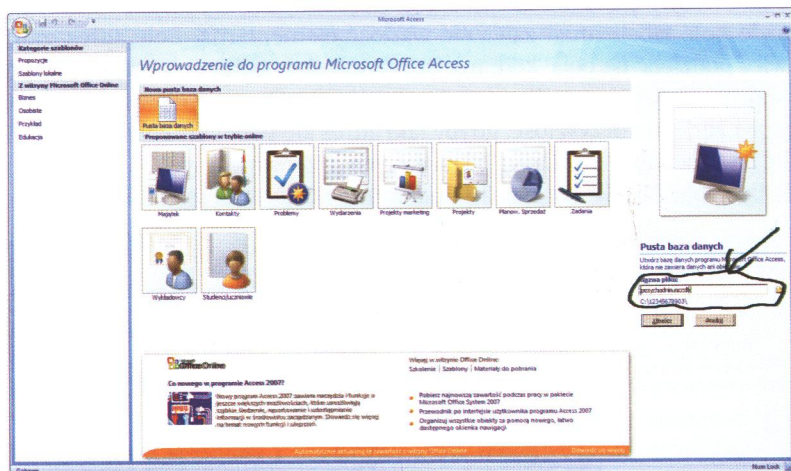
Te zagadnienia dotyczą efektów kształcenia E.14.2.3, E.14.2.4, E.14.2.7, E.14.1.13. Materiał niezbędny do wykonania tego zadania znajduje się w rozdziałach 9. i 27. podręcznika *Bazy danych i systemy baz danych* Wydawnictw Szkolnych i Pedagogicznych (Warszawa 2013).

Normalizacja bazy danych jest to proces mający na celu eliminację powtarzających się danych w relacyjnej bazie danych. Idea normalizacji polega na przechowywaniu danych tylko w jednym miejscu, co zwiększa bezpieczeństwo danych i zmniejsza ryzyko powstania niespójności (problemów anomalii). W celu określenia poziomu normalizacji baza

danych może być porównywana ze schematami nazywanymi **postaciami normalnymi** PN (*normal forms*). Normalizacja bazy danych może wymagać rozbicia dużych tabel na mniejsze i ponownego ich łączenia przy każdym zapytaniu. Normalizacja nie powoduje utraty danych tylko zmienia sposób ich przechowywania. **Pierwsza postać normalna** (1PN) oznacza, że każda składowa w każdej komórce jest wartością elementarną (nie można podzielić jej na mniejsze wartości).

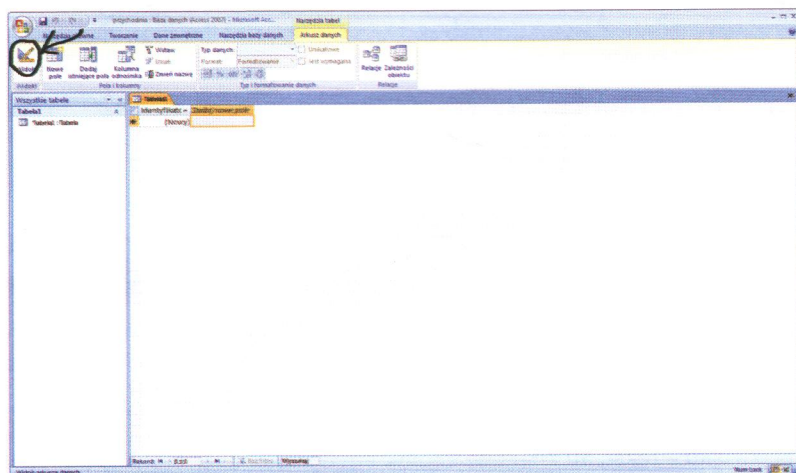
Aby utworzyć bazę danych lekarzy w przychodni zawierającą podane w zadaniu dane, należy wykonać następujące operacje:

- utworzyć bazę danych. Należy otworzyć aplikację MS Access i wskazać ikonę tworzenia nowej pustej bazy danych. W prawej części okna (na rys. 3.11 zaznaczone strzałką) wprowadzić nazwę bazy: przychodnia, oraz ścieżkę dostępu do miejsca przechowywania tej bazy. Kliknąć przycisk **Utwórz**;



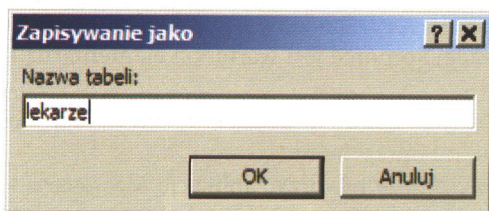
Rys. 3.11. Okno tworzenia nowej pustej bazy danych

- zmienić tryb pracy z widoku na projektowanie tabeli. Należy kliknąć przycisk Widok (na rys. 3.12 zaznaczony strzałką);



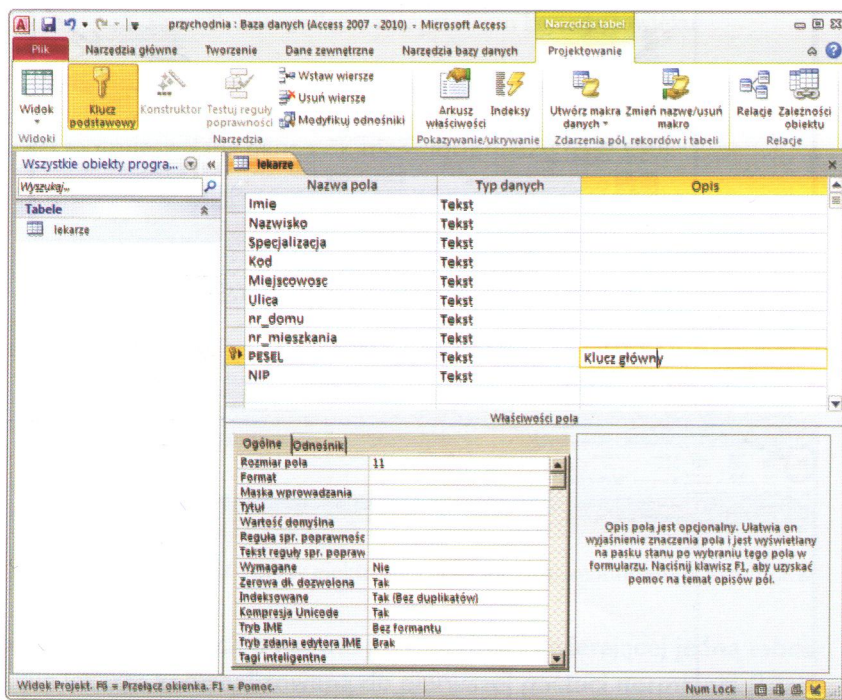
Rys. 3.12. Zmiana trybu pracy z widoku na projektowanie tabeli

- zapisać tabelę – zmiana trybu pracy z widoku na projektowanie tabeli wymaga zapisania tabeli. W oknie pokazanym na rys. 3.13 należy wprowadzić nazwę tabeli: lekarze;



Rys. 3.13. Zapisywanie tabeli

- wprowadzić nazwy pól oraz określić ich typy danych. Zgodnie z zasadami normalizacji każde pole tabeli musi zawierać wartość elementarną. Pole adres_zamieszkania, trzeba więc podzielić na pola: kod, miejscowość, ulica, numer_domu, numer_mieszkania (wprawdzie w danych w tabeli nie ma żadnego adresu z numerem mieszkania, ale jest to zabezpieczenie na przyszłość, gdyby taki adres się pojawił). Pole PESEL / NIP należy podzielić na dwa pola PESEL i NIP. Dla danych przedstawionych w tabeli kluczem głównym ma być pole PESEL. Na rys. 3.14 pokazano strukturę tabeli lekarze.



Rys. 3.14. Struktura tabeli lekarze

Dla poszczególnych pól przyjęto następujące właściwości:

- pole Imię – tekst, 30 znaków;
- pole Nazwisko – tekst, 50 znaków;
- pole Specjalizacja – tekst, 50 znaków;

- pole Kod – tekst, 6 znaków;
 - pole Miejscowosc – tekst, 30 znaków;
 - pole Ulica – tekst, 30 znaków;
 - pole nr_domu – tekst, 4 znaki (czasem w numerach domu używa się liter, np. 1a);
 - pole nr_mieszkania – tekst, 4 znaki (czasem w numerach mieszkań używa się liter, np. 1a);
 - pole PESEL – tekst, 11 znaków (liczba całkowita długa nie pozwala na zapisanie numeru PESEL, ponadto niektórzy użytkownicy podają numer ze spacjami w środku);
 - pole NIP – tekst, 10 znaków.
- zmienić tryb pracy na Widok i wprowadzić dane do tabeli. Na rys. 3.15 pokazano tabelę lekarze z wprowadzonymi danymi.

Imię	Nazwisko	Specjalizacja	Kod	Miejscowosc	Ulica	nr_domu	nr_mieszkanie	PESEL	NIP
Andrzej	Cieslak	Ginekolog	90-987	Warszawa	Piękna	3		65120302014	7652349123
Anna	Gawrońska	Internista	91-611	Łódź	Mala	34		74100923967	7567983949
Maciej	Piękny	Chirurg	90-008	Pila	Zenona	32		65123009034	6789066565
Angelika	Męska	Internista	23-098	Warszawa	Mackiewicz	2		74040434098	4643384844
Michał	Lasota	Chirurg	78-098	Pila	Śmieta	8		88020187234	7788989900

Rys. 3.15. Tabela lekarze z wprowadzonymi danymi

Plik wynikowy należy zapisać na dysku C:\ w folderze o nazwie zgodnej z twoim numerem PESEL.

Egzaminator przyznaje punkty za osiągnięcie przez egzaminowanego określonych rezultatów. Przykładowe rezultaty i liczba punktów możliwych do uzyskania po ich osiągnięciu została pokazana w tabeli 3.2. Jest to tabela przykładowa, natomiast egzaminator będzie posługiwał się szczegółowym kluczem, opisującym precyzyjnie reguły przyznania każdego punktu.

Tabela 3.2. Przykładowe rezultaty zadania praktycznego

E.14. Tworzenie aplikacji internetowych i baz danych oraz administrowanie bazami

SCHEMAT OCENIANIA TESTU PRAKTYCZNEGO 3

Lp.	Elementy podlegające ocenie / kryteria oceny	Maksymalna liczba punktów możliwych do uzyskania za element / spełnienie kryterium
1.	Rezultat 1: Wykonany szablon strony internetowej	40
1	Utworzono plik html zawierający szablon podziału strony na bloki o nazwach Box, Gora, Srodek i Lewy	5
2	Do zdefiniowania bloków strony użyto znacznika <div>	7
3	Utworzono plik css jako zewnętrzny arkusz stylów oraz zdefiniowano odniesienie do niego w pliku html	5

Lp.	Elementy podlegające ocenie / kryteria oceny	Maksymalna liczba punktów możliwych do uzyskania za element / spełnienie kryterium
4	Ustalono w zewnętrznym pliku css wielkości poszczególnych bloków Box (szerokość – 950 px), Gora (szerokość – 950 px i wysokość – 180 px) i Srodek (szerokość – 950 px) oraz Lewy na (szerokość – 220 px wysokość – 420 px)	5
5	Zdefiniowano poszczególne kolory tła na: Box – #FFFFFF, Gora – #006600, Srodek – #CCFF33, Lewy – #99FF00. Dopuszczalne jest zastosowanie innego, poprawnego sposobu kodowania kolorów	5
6	Zdefiniowano w bloku Gora pozycjonowanie relatywne	2
7	Zdefiniowano w blokach Środek i Lewy umieszczenie elementu z lewej strony	2
8	Ustalono w bloku Box automatyczne marginesy	2
9	Zapisano rezultaty pracy w dwóch plikach o nazwach: strona.html i szablon.css oraz zapisano na dysku C:\ w folderze o nazwie zgodnej z numerem PESEL	2
10	Gotowy szablon jest zgodny ze wzorem z rysunku 1	5
2.	Rezultat 2: Utworzony skrypt formularza weryfikujący wiek	30
1.	Skrypt uruchamia się jako formularz oraz umożliwia wprowadzanie danych	10
2	Skrypt weryfikuje wiek wypełniającego	8
3	W przypadku wprowadzenia wieku poniżej 18 wyświetlony zostaje komunikat błędu	5
4	Skrypt zawiera akcję wysłania danych na adres adam@wp.pl	5
5	Skrypt zapisano pod nazwą formularz na dysku C:\ w folderze o nazwie zgodnej z numerem PESEL	2
3.	Rezultat 3: Utworzona znormalizowana tabela bazy danych	35
1	Utworzono bazę danych o nazwie przychodnia oraz zapisano ją na dysku C:\ w folderze o nazwie zgodnej z numerem PESEL	5
2	Utworzono jedną tabelę o nazwie lekarze	5

Lp.	Elementy podlegające ocenie / kryteria oceny	Maksymalna liczba punktów możliwych do uzyskania za element / spełnienie kryterium
3	Tabela zawiera elementarne wartości atrybutów – 1PN (Rozdzielono pole PESEL/NIP na dwa pola oraz pole Adres zamieszkania na co najmniej 3 pola)	10
4	Wprowadzono do utworzonej tabeli dane zgodne z Tabelą 1 w treści zadania	5
5	Ustawiono właściwości pól w sposób umożliwiający właściwe wpisanie danych	5
6	Ustawiono klucz główny dla pola PESEL	5
Liczba punktów razem		100