

**AKADEMIA HANDLOWA  
NAUK STOSOWANYCH W RADOMIU**



**RADOM  
ACADEMY OF ECONOMICS**

# **Akademia Handlowa Nauk Stosowanych w Radomiu**

## **Technologie internetowe Laboratorium 2**

**Radom 2021/2022**

## 1. Cel zadania

Obecnie rozwijana aplikacja (z laboratorium 1) powinna otrzymać funkcjonalność zapisu danych klienta/administratora oraz mieć możliwość wyświetlania/edycji tych danych.

## 2. Potrzebne narzędzia.

- Dowolny zestaw komputerowy z dowolnym systemem operacyjnym
- Dowolny serwer stron www

INFORMACJA: Ze względu na charakter zajęć zalecane jest użycie narzędzia XAMPP lub równoważnego; konfiguracja serwera jest sprawą drugorzędną, zaś najważniejsze aspekty (projektowanie, programowanie i łączenie opcji serwerowych i lokalnych) są w tym narzędziu dostępne bez jakiegokolwiek konfiguracji.

## 3. Informacje wstępne

Mając stworzony szablon dowolnej witryny/aplikacji w poprzednim laboratorium można zauważyć, że brakuje tejże aplikacji istotnej funkcjonalności - możliwości zapisywania i wyświetlania danych, które pierwotnie ma ona przechowywać. Jednym z najpopularniejszych rozwiązań służących do przetwarzania dużej ilości danych są bazy danych. Obecnie najczęściej stosuje się dwa typy baz danych: relacyjne i relacyjno-obiektowe oraz bazy nierelacyjne (płaskie, plikowe, tekstowe).

Pierwszym i historycznie najstarszym rozwiązaniem są bazy nierelacyjne. Pozwalają one na przechowywanie danych, przeważnie tekstowych, wedle różnych formatowań. Najprostszym sposobem na przechowywanie danych jest oddzielić pomiędzy sobą zapisane dane określonym znakiem separacji - może to być pojedynczy biały znak (spacja), przecinek, średnik, znak tabulacji bądź sekwencja znaków. Dodatkowo (choć nie jest to wymagane) poszczególne grupy danych mogą być rozdzielane znakami nowej linii/powrotu kursora (`\r\n`) - jest to wtedy podział na serie danych (wiersze). Niekiedy pliki tego typu baz danych posiadają własny format pliku, w którym wykorzystuje się język XML (wraz ze wszystkimi odmianami) lub popularny ostatnio JSON. Oba formaty mają charakter tekstowy, chociaż nie wyklucza to zapisu w formie binarnej (np. BJSON – odpowiednik nietekstowy formatu oryginalnego JSON). Przykładem plików przechowujących dane mogą być wspomniane formaty xml lub json, ale także pliki csv bądź xsl(x).

Kolejnym rozwiązaniem bazodanowym są bazy relacyjne i relacyjno-obiektowe. Charakteryzują się własnym formatem przechowywanych plików, nierzadko binarnym. Najważniejszą funkcjonalnością, w przeciwieństwie do poprzednich systemów składowania danych, jest możliwość łączenia informacji z różnych grup, nazywanych tutaj tabelami, z danymi w innych tabelach (bądź co najmniej jednej innej tabeli). Relacje w głównej mierze zapobiegają powtarzaniu przechowywanych informacji, co z kolei mogłoby doprowadzić do niejednoznaczności danych. Najpowszechniej występują trzy typy relacji:

- Jeden-do-wielu – jest to najczęściej występująca relacja. Pozwala połączyć jeden, najczęściej unikatowy zasób z zasobami, które mogą go wielokrotnie wykorzystywać. Przykładem takiej relacji mogą być kody pocztowe (które zawsze są unikatowe w skali kraju) z nazwami miast, posiadające nawet kilka kodów pocztowych. Jedno miasto może posiadać wiele kodów pocztowych (jeden-do-wielu)
- Wiele-do-wielu - równie popularna relacja. Występuje zawsze tam, gdzie poszczególne grupy danych muszą być powiązane nieokreśloną ilość razy z grupą innych danych. Relacja ta zawsze wymaga dodatkowej tabeli, która pozwala połączyć unikatowe identyfikatory dwóch łączonych grup. Przykładem takiej relacji może być połączenie lekarza (unikatowy w skali szpitala) z określonymi pacjentami (unikatowi ludzie, którzy mogą korzystać z usług tego samego specjalisty). Dodatkowo pacjent (unikatowy) może korzystać z usług kilku unikatowych lekarzy.

- Jeden-do-jednego – rzadka relacja ze względu na dość specyficzne i ograniczone możliwości jej wykonania. Pozwala łączyć jedne unikatowe dane z określonej grupy danych z innymi unikatowymi danymi z drugiej grupy danych. Przykładem tej relacji może być połączenie obywatela (unikatowego w skali kraju) z dowodem osobistym (unikatowym dokumentem w skali kraju, nie mogącym być wykorzystanym z innym obywatelem).

Bazy danych mogą być obsługiwane poprzez odpowiednio napisane programy zarządzające danymi, zaś komunikacja na poziomie wymiany danych na poziomie baza-aplikacja odbywa się poprzez tzw. sterownik - najczęściej mający postać biblioteki zestaw funkcji umożliwiający wykonywanie operacji na danych w samej bazie.

Bardzo często bazy danych posługują się dodatkowym, wysokopoziomowym językiem zapytań pozwalającym na łatwe i precyzyjne operacje na danych. Najczęściej bazy danych mają zaimplementowaną obsługę Structured Query Language (SQL), szeroko wspieranego standardu do pełnego zarządzania i manipulacji informacjami zebranymi w bazie. Język ten dzieli się na zapytania definiujące bazę danych (DDL), zapytania operowania na już wstawionych danych (DML) oraz zarządzania serwerem bazy danych (DCL). Niekiedy wymienia się też, jako osobną grupę, samo definiowanie zapytań do elementów bazy danych jako język definicji zapytań (DQL).

Połączeń naszej wcześniej stworzonej strony WWW (aplikacji WWW) z bazą danych należy dokonać przy wykorzystaniu języka po stronie serwera WWW. Językiem tym może być dowolny język programistyczny (pierwotnie C/C++), który będzie w stanie zredagować odpowiedź na zapytanie przeglądarki i wysłać odpowiedź już przetworzonego dokumentu HTML. Ponieważ cała odpowiedź jest redagowana po stronie serwera, klient nie zobaczy takich danych jak login i hasło do bazy danych, zapytań do bazy danych (a tym samym nie pozna jej struktur), dodatkowego kodu łączącego pliki i dane w jedną odpowiedź.

Jednym z popularniejszych języków programowania po stronie serwera jest PHP. Język ten zapewnia względnie efektywną manipulację danymi przy jednoczesnie relatywnie niskimi wymaganiami sprzętowymi. PHP należy do grupy języków wysokopoziomowych interpretowanych. Interpreter, jak i całe środowisko uruchomieniowe napisane zostało w języku C. Najnowszą wersją języka jest wersja 8. Sam język składniowo przypomina C, Java, Perl oraz JavaScript. Posiada dynamiczne typy danych, jednak od wersji 8 preferowane jest jawne wskazywanie typu danych, jakie chcemy utworzyć. W pełni obsługiwana jest obiektowość, konwersja typów, wzorce ciągów znakowych i inne, nowoczesne standardy programistyczne.

## 4. Przebieg.

Do rozwijanej już aplikacji (laboratorium pierwsze) należy dołożyć funkcjonalność obsługi bazy danych. Wskazane jest używanie rozwiązań nowoczesnych, mile widziane frameworki (aczkolwiek można też wykonać wszystko samemu). Najistotniejsze jest, by tworzona baza danych jedynie ukazywała możliwości implementacji jej w takiej aplikacji (nie należy przykładać do niej większej wagi). Istotna będzie tutaj komunikacja aplikacji z bazą przy wykorzystaniu języka serwerowego.

## 5. Zakończenie

Rozwiązanie, w postaci adresu WWW do własnego repozytorium, należy przesłać na adres [piotr\\_dobosz@int.pl](mailto:piotr_dobosz@int.pl), w temacie wiadomości zawierając frazę [AHNS\_TI]. Jeżeli odnośnik do repozytorium był przesłany wcześniej - nie trzeba przysyłać nic więcej (przesłany wcześniej projekt po prostu będzie rozbudowany o nową funkcjonalność).