

Programowanie aplikacji internetowych

Include, Heredoc, funkcje.
Szablony treści.

PHP

PHP wspiera metodyki programowania proceduralne i obiektowe, można też dołączać całe pliki PHP

- Często wykonywane czynności można zamknąć w funkcji
- Bardziej złożone i powiązane ze sobą czynności można zamknąć w definicji klasy, jej metod i właściwości
- Potrzebne funkcje i klasy można skopiować do nowego projektu
- Często wykorzystywane funkcje i klasy można zapisać w oddzielnym pliku i dołączyć do nowego projektu

... Raz napisany kod PHP można używać wielokrotnie.

PHP

Biblioteki PHP

- Istnieje wiele gotowych bibliotek PHP, które dodają nawet bardzo złożone funkcjonalności, np.:
 - TCPDF – generowanie plików PDF
<https://tcpdf.org/>
 - PHPMailer – wysyłanie i odbieranie poczty elektronicznej
<https://github.com/PHPMailer/PHPMailer>
 - RedBinPHP – ORM (mapowanie obiektowo-relacyjne)
<https://redbeanphp.com/index.php>
 - PHP Markdown – parser języka Markdown
<https://github.com/michelf/php-markdown>

PHP

Biblioteki PHP

- Frameworki aplikacji webowych:
 - Symfony, <http://symfony.com/>
 - Laravel, <http://laravel.com/>
 - Yii, <http://www.yiiframework.com/>
 - Silex, <http://silex.sensiolabs.org/> (micro framework)
- Systemy szablonów treści:
 - Twig, <http://twig.sensiolabs.org/> (część Symfony)
 - Mustache, <https://github.com/bobthecow/mustache.php>
 - Latte, <https://latte.nette.org/pl/>

Przegląd innych bibliotek: <https://gist.github.com/llbbl/7607016>

PHP include/require

Włączanie pliku PHP

- Skrypt PHP może wykorzystywać inne skrypty PHP, służą do tego instrukcje:
 - include, include_once
 - require, require_once
- Wskazany w instrukcji include/require skrypt jest wczytywany i dołączany w miejscu użycia tej instrukcji
- Jeżeli dołączany skrypt zawiera HTML, jest on wysyłany do wyjściowego strumienia skryptu głównego, jeżeli PHP – jest interpretowany
- Rozszerzenie nazwy pliku nie jest istotne; niektóre podręczniki zalecają .inc (jako skrót od include)
- Pliki powinny być umieszczone poza public_html (niepraktyczne) lub zabezpieczone przez bezpośrednim wykonaniem

PHP include/require

Włączanie pliku PHP

- Instrukcja include/require zwykle służy do włączania gotowego, powtarzającego się fragmentu HTML:

```
<?php
include "header.php";
include "banner.php";

// zmienne elementy strony

include "footer.php";
?>
```

PHP include/require

Włączanie pliku PHP

- Instrukcja `include_once/require_once` zwykle służy do włączania pomocniczych skryptów, zawierających definicje klas, funkcji oraz zmiennych:

```
//index.php
<?php
require_once "tepmlate.php";
```

```
//template.php
<?php
$pageTemplate = "...";
function Render (...) {
...
}
```

PHP include/require

Włączanie pliku PHP

- Zabezpieczenie przed wczytaniem pomocniczego skryptu: tablica sesji (zwykle i tak jest używana) i przekierowanie

```
//index.php
<?php
session_start();
require_once "tepmlate.php";

//template.php
<?php
if (!isset ($_SESSION)) {
    header ("Location: index.php");
    die ();
}
```

PHP, łańcuchy znaków

Długie łańcuchy znaków

- Są powszechne w PHP, ponieważ głównym celem PHP jest wytworzenie dokumentu HTML i przesłanie go do przeglądarki, w kawałkach lub w całości
- W większości przypadków, choć nie zawsze, do łańcucha trzeba wstawiać wartości zmiennych
- Istnieje kilka technik, ułatwiających tworzenie i zmiany długich łańcuchów znaków:
 - operator konkatencji i wsparcie NetBeans
 - notacja heredoc (spotykana też w innych językach)
 - notacja nowdoc (tylko w PHP)

PHP, łańcuchy znaków

Konkatenacja i wsparcie NetBeans

- Łańcuchy można łączyć operatorem konkatenacji "." (kropka)
- NetBeans po wciśnięciu <Enter> dzieli łańcuch między linie, dzięki czemu można wygodnie wpisywać dłuższe fragmenty tekstu (pozycja kursora: **|**) :

```
// Przed naciśnięciem <Enter>:
```

```
$info = "<a href='{ $href } '>{ $name }</a>\n|";
```

```
// Po naciśnięciu <Enter>:
```

```
$info = "<a href='{ $href } '>{ $name }</a>\n"  
      . "|";
```

PHP, łańcuchy znaków

Konkatenacja i wsparcie NetBeans

- ...
- Niestety, taki kod nie jest czytelny, a zmiany w nim są trudne, zwłaszcza jeżeli zawiera dużo nazw zmiennych:

```
$info = "<div class='grid-box'>"  
    . "<a href='{ $href} '>"  
    . "<img src='./images/{ $name} '.jpg'"  
    . " alt='{ $description} '>"  
    . "</a>"  
    . "</div>";
```

PHP, łańcuchy znaków

Konkatenacja i wsparcie NetBeans

- ...
- Dodatkowo w łańcuchach ograniczonych cudzysłowami trzeba używać apostrofów lub sekwencji ucieczki, natomiast żeby kod HTML był czytelny, należy dodawać znaki końca linii:

```
$info = "<div class=\"grid-box\">\n"  
    . "<a href=\"{$href}\">{$name}</a>\n"  
    . "</div>\n";
```

PHP, łańcuchy znaków

Heredoc

- Poprawia czytelność i ułatwia utrzymanie łańcuchów znaków, umożliwia używanie cudzysłowów, zapewnia interpolację zmiennych, zachowuje podziały linii:

```
$info = <<<EOM
```

```
<div class="grid-box">
```

```
<a href="{ $href} ">{ $name} </a>
```

```
</div>
```

```
EOM;
```

PHP, łańcuchy znaków

Heredoc

- ...
- Początek heredoc to ciąg trzech znaków "<" + dowolna nazwa, opcjonalnie w cudzysłowach (ale nikt tego nie robi); PHP manual używa EOT (End of Text) lub END
- Koniec heredoc to powtórzenie nazwy początku heredoc, opcjonalnie ze średnikiem; Wcięcie końca heredoc nie może być większe, niż zawartości heredoc:

```
$info = <<<"END"
```

```
<p>Wcięcie nie może być mniejsze, niż tu
```

```
END;
```

PHP, łańcuchy znaków

Nowdoc

- Podobny do heredec, ale nie ma interpolacji zmiennych;
Nazwa początku nowdoc musi być zapisana w apostrofach

```
$pageTemplate = <<<'EOM'  
    <!DOCTYPE ...  
    <html>  
        <head>  
        </head>  
  
    EOM;
```

Funkcje PHP

Funkcje

- Definiowanie funkcji

```
function Name ($arg1, $arg2, ..., $argn)
{
    // dowolne instrukcje
    // opcjonalnie:
    return result;
}
```

- Wywołanie funkcji

```
$val1 = 7;
$val2 = "value";
$res = Foo ($val1, $val2, 13, "other");
```

Funkcje PHP

Argumenty i rezultat funkcji

- W PHP nie ma deklaracji zmiennych, dlatego typ rezultatu i typy argumentów nie muszą być formalnie określone;
- Każda funkcja zwraca rezultat (niejawnie: NULL)
- Wiele standardowych funkcji PHP może zwracać rezultaty różnego typu, zależnie od sytuacji

```
function getConnection ()  
{  
    if (...)  
        return $dbConnection;  
    else  
        return False  
}
```

Funkcje PHP

Argumenty domyślne

- Wiele standardowych funkcji PHP ma argumenty domyślne, można też ich używać we własnych funkcjach

```
function Foo ($str, $low=True, $trm=True)
{
    ...
}
```

```
Foo ($name);           // Foo($name, True, True)
Foo ($name, False);    // Foo($name, False, True)
```

Funkcje PHP

Argumenty nazwane funkcji

- Umożliwia przekazanie argumentów według ich nazw, zamiast według kolejności

```
function Foo ($index, $value)
{
    ...
}
```

```
Foo($i, $v); // wg kolejności, $i - $index
```

```
Foo(value: $v, index: $i) // wg nazw
```

- Element składni wprowadzony w PHP 8.0

Szablony PHP

Tworzenie strony na podstawie szablonu

- Opracowanie szablonu
wzorcowa strona (tylko HTML), wszystkie elementy, które będą na stronach tworzonych dynamicznie
- Zastąpienie elementów zmiennych unikalnymi ciągami znaków, (np. ===Nazwa===), zapisanie szablonu do zmiennej PHP
- Dynamiczne tworzenie – w skrypcie PHP – tylko elementów zmiennych strony, zapisywanych do zmiennych PHP, np.:

```
$pageBody = "<h2>Xxx</h2>";  
while (...)   
    $pageBody .= "<p>Yyy";  
$pageBody .= "<p>Zzz";
```

Szablony PHP

Tworzenie strony na podstawie szablonu

- ...
- Na koniec – wstawienie utworzonych dynamicznie elementów do szablonu strony i wysłanie całości do przeglądarki:

```
$page = str_replace(  
    "===PageBody===", $pageBody, $pageTemplate  
);  
echo $page;
```

Szablony PHP

Korzyści tworzenia strony na podstawie szablonu:

- Wszystkie strony serwisu mają dokładnie taki sam układ oraz takie samo formatowanie,
- Zmiana wyglądu jest łatwa – zamiast zmieniać kod wielu skryptów, wystarczy zmodyfikować szablon,
- Szablony można zapisać w bazie danych, wówczas zmiana wyglądu może być dokonana dynamicznie,
- Dokument HTML, produkowany przez skrypt, jest wysyłany do przeglądarki dopiero na sam koniec działania skryptu:
 - W dowolnej chwili można zaprzestać tej produkcji lub całkowicie zmienić treść (np. w reakcji na błąd) – normalnie to niemożliwe
 - W dowolnej chwili można skorzystać z nagłówków HTTP (np. funkcje `session_start`, `set_cookie`, `header`) – normalnie jest to możliwe tylko na samym początku działania skryptu.

Programowanie aplikacji internetowych

PAI Scale Models
(aka Cars)

PAI Scale Models



PAI Scale Models

Piotr Kaczorek

Start

Login

Products

Products

Search

Lines

Work

Styles

Test

1936 Mercedes-Benz 500K Special Roadster [✎]

This 1:18 scale replica is constructed of heavy die-cast metal and has all the features of the original: working doors and rumble seat, independent spring suspension, detailed interior, working steering system, and a bifold hood that reveals an engine so accurate that it even includes the wiring. All this is topped off with a baked enamel finish. Color white.

\$ 53.91

Scale: 1:18

Vendor: Studio M Art Models

Line: Vintage Cars



PAI Scale Models



PAI Scale Models
Piotr Kaczorek

☰

Login

Your login

Your password

Log in

PAI Scale Models

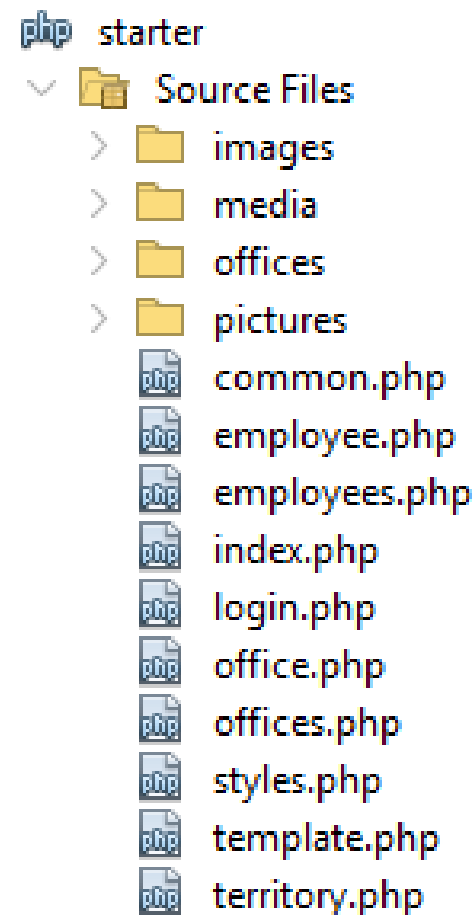
Aplikacja PHP Scale Models

- Oparta na modelowej bazie danych MySQL,
- Wersja "Starter", do pobrania ze strony WWW, zawiera wiele gotowych elementów:
 - Szkielet aplikacji, oparty na prostych szablonach treści, w tym funkcje generujące menu główne i menu boczne
 - Funkcje ułatwiające korzystanie z bazy danych, obejmujące obsługę błędów
 - Funkcje generujące formularze
 - Wzory wielu elementów strony
 - Zdjęcia modeli pojazdów i "pracowników"

PAI Scale Models

Struktura aplikacji:

- images – zdjęcia modeli
- pictures – zdjęcia pracowników
- offices – zdjęcia miast
- media – pliki CSS i Javascript
- common.php – funkcje bazy danych
- template.php – szablony treści oraz menu
- styles.php – wzory elementów strony
- index.php – strona główna, pusta
- login.php – formularz logowania
- pozostałe pliki PHP – przykłady użycia elementów wymienionych wyżej, jako wzór do wykorzystania przy rozwijaniu aplikacji



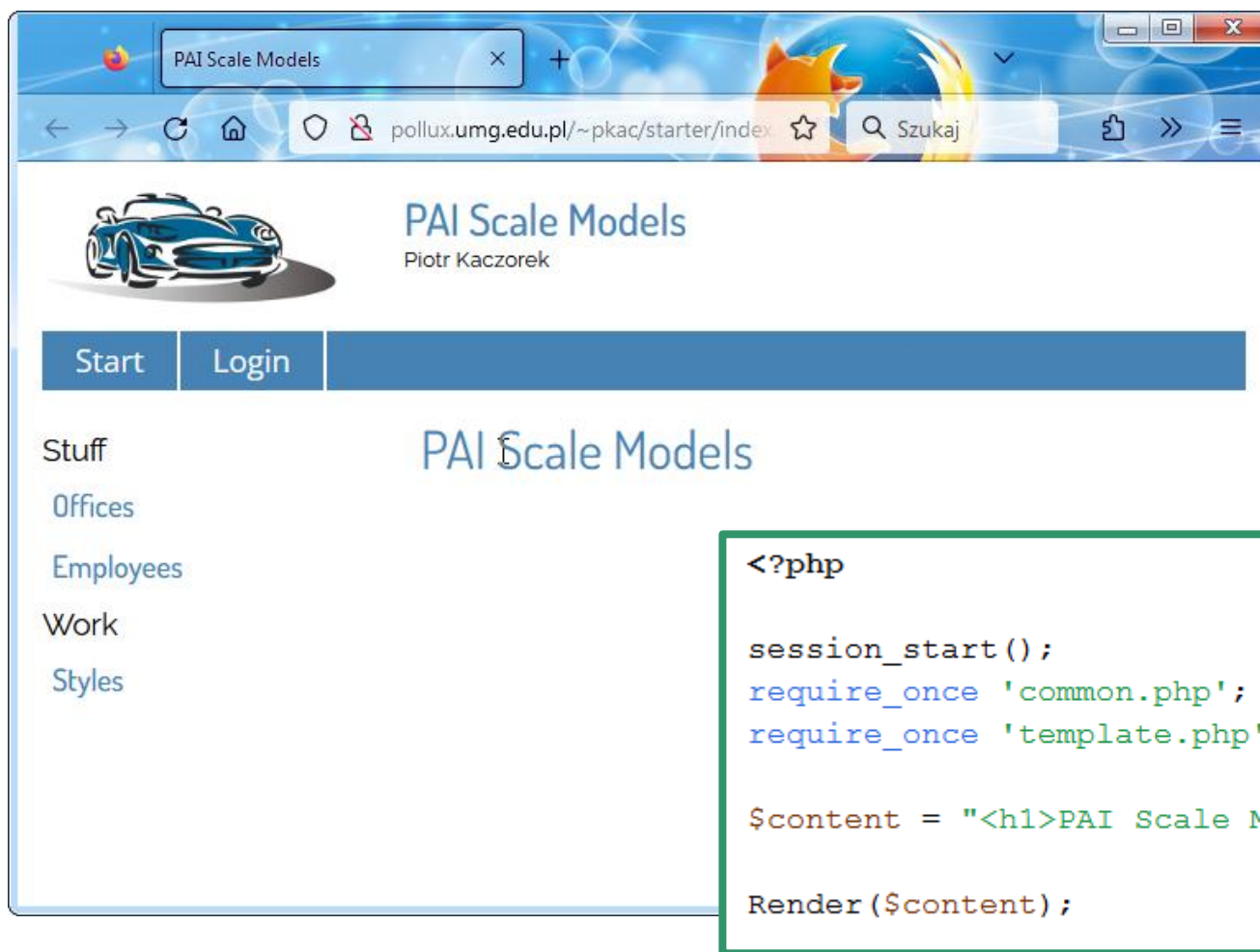
PAI Scale Models

Szablon treści



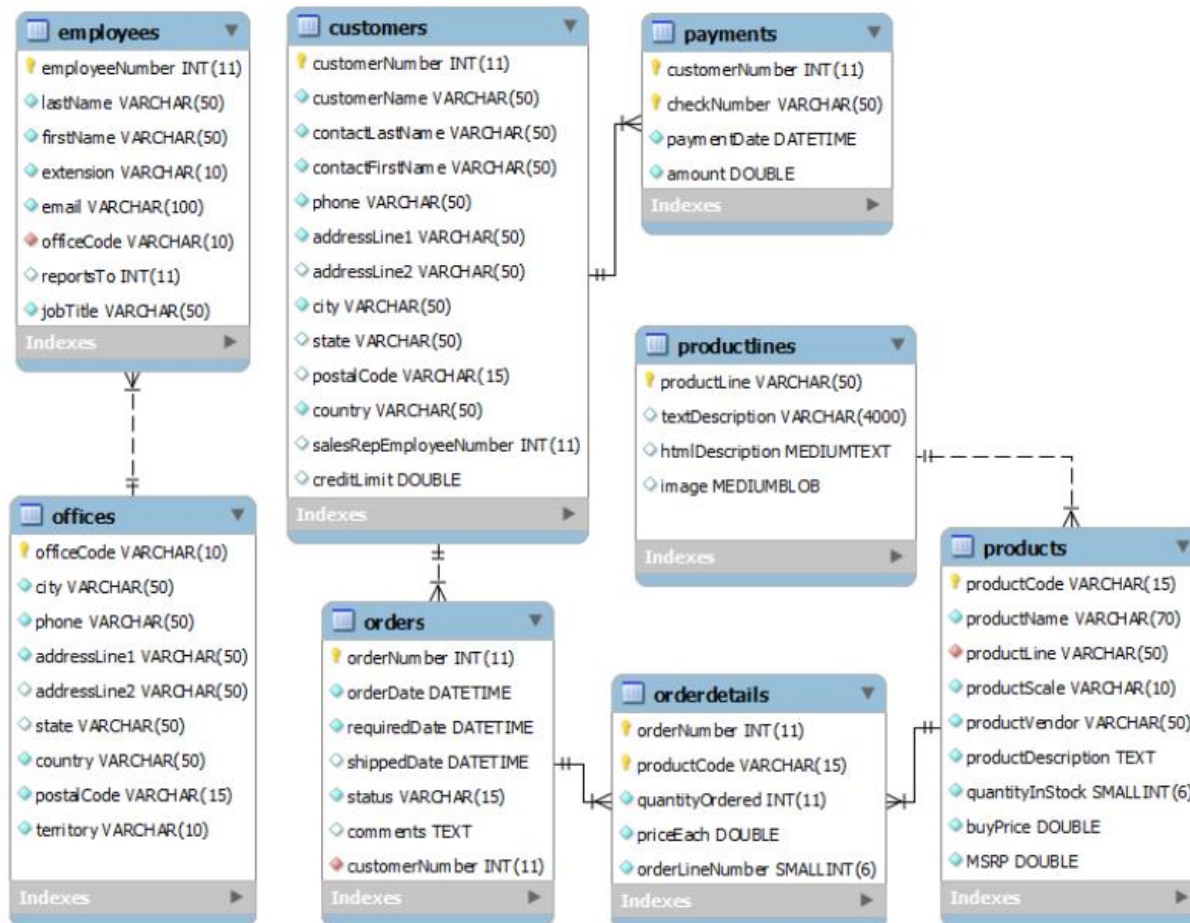
PAI Scale Models

Szablon treści



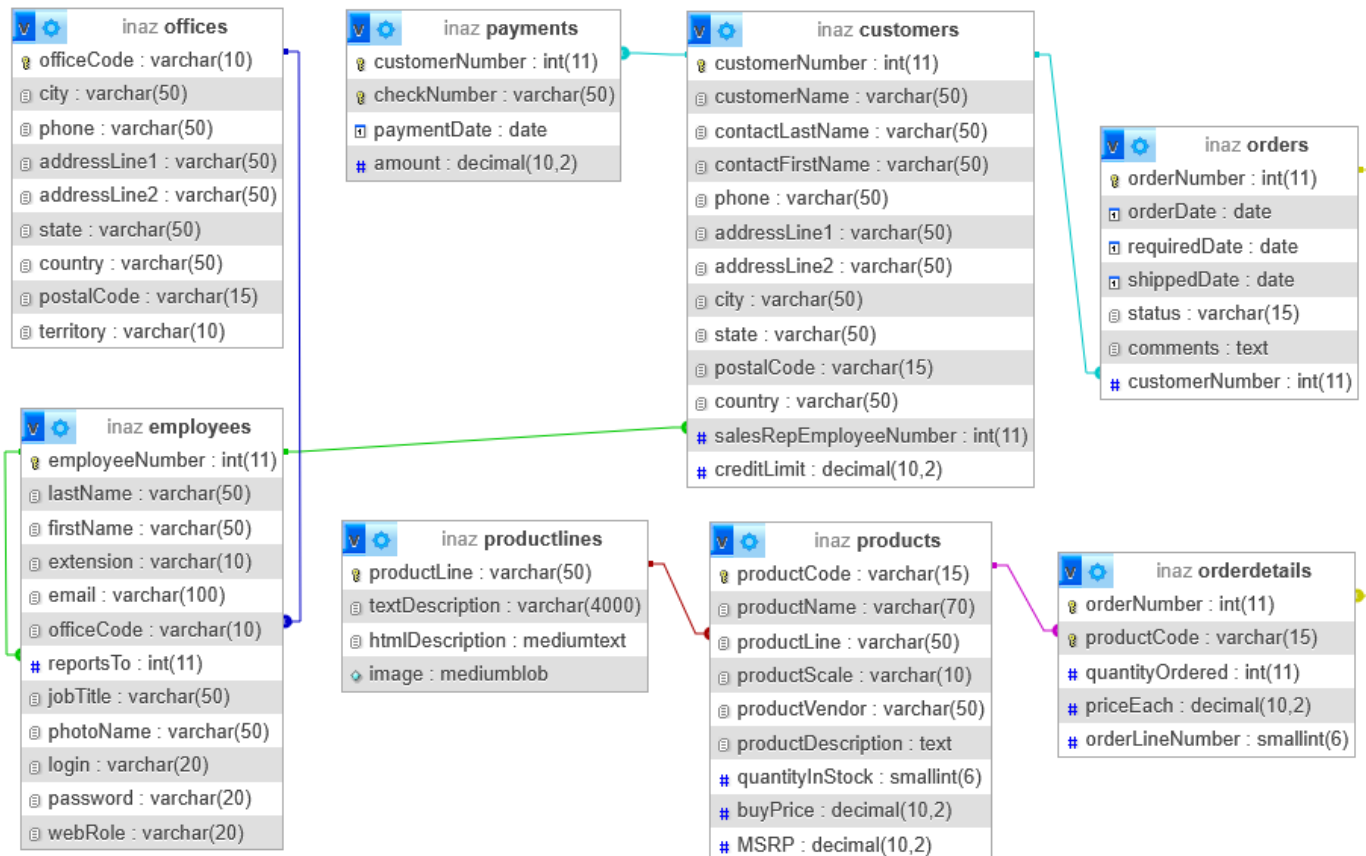
PAI Scale Models

Baza danych



PAI Scale Models

Baza danych



PAI Scale Models

Zmiany w bazie danych, w stosunku do schematu:

- Tabela employees:
 - photoName – nazwa pliku zdjęcia, bez folderu i rozszerzenia,
 - login – login w konwencji INAZ ,
 - password – hasło (kernel21),
 - webRole – rola w aplikacji, np. "admin" lub "stuff"

PAI Scale Models

Zmiany w bazie danych, w stosunku do schematu:

- Tabele orders i payments:
 - Wszystkie daty (orderDate, requiredDate, shippedDate, paymentDate) przesunięte do współczesności;
 - Można zmienić, wykonując np. w phpMyAdmin, w oknie "Edytuj w linii", kod:

```
SET @target = '2025-05-15';
SET @days = (SELECT DATEDIFF(@target, (SELECT MAX(orderDate) FROM orders )));
UPDATE orders SET
    orderDate = DATE_ADD(orderDate, INTERVAL @days DAY),
    requiredDate = DATE_ADD(requiredDate, INTERVAL @days DAY);
UPDATE orders SET
    shippedDate = DATE_ADD(shippedDate, INTERVAL @days DAY)
    WHERE NOT shippedDate IS NULL;
UPDATE payments SET
    paymentDate = DATE_ADD(paymentDate, INTERVAL @days DAY);
```

PAI Scale Models

Klasa dbQuery

- Ułatwia korzystanie z bazy danych, w tym obsługę błędów
- Odpowiada za utworzenie połączenia z serwerem MySQL, wysłanie zapytania i zwrócenie odpowiedzi
- Można wykorzystać na dwa sposoby – przez utworzenie obiektu albo przez metody statyczne

PAI Scale Models

Klasa dbQuery

- W pliku common.php trzeba wpisać własne dane uwierzytelniające do połączenia z bazą danych:

```
$conn = new mysqli(  
    "host", "user", "pass", "db");
```

gdzie:

host – serwer, zwykle "localhost",

user – nazwa użytkownika,

pass – hasło,

db – nazwa bazy danych (na Polluksie jak nazwa użytkownika)

PAI Scale Models

Klasa dbQuery

- Sposób 1 – utworzenie obiektu dbQuery i pętla while:

```
$sql = "SELECT * FROM employees";  
$query = new dbQuery($sql);  
  
while (($employee = $query->next()) != NULL) {  
    // $employee zawiera dane pracownika  
    // np. $employee["firstName"] - imię  
}
```

Konstruktor dbQuery łączy się z bazą danych i wysyła zapytanie;
Każde kolejne użycie metody next pobiera jeden wiersz z wyniku,
każdy w formie tablicy asocjacyjnej

PAI Scale Models

Klasa dbQuery

- Sposób 2 – metoda statyczna *getAll* i pętla foreach

```
$sql = "SELECT * FROM employees";  
$employees = dbQuery::getAll($sql);  
  
foreach ($employees as $employee) {  
    // $employee zawiera dane pracownika  
    // np. $employee["firstName"] - imię  
}
```

Metoda `getAll` łączy się z bazą danych, wysyła zapytanie i zwraca rezultat jako tablicę tablic – każdy element tablicy to wiersz odpowiedzi, sam też w formie tablicy asocjacyjnej

PAI Scale Models

Klasa dbQuery

- Sposób 2 – metoda statyczna *getOne*:

```
$sql = "SELECT * FROM employees WHERE ... ";  
$employee = dbQuery::getOne($sql);
```

```
if (isset($employee)) {  
    // $employee zawiera dane pracownika  
    // np. $employee["firstName"] - imię  
}
```

Metoda *getOne* łączy się z bazą danych, wysyła zapytanie, pobiera z wyniku wyłącznie pierwszy wiersz i zwraca go, w formie tablicy asocjacyjnej, albo NULL, jeżeli rezultat jest pusty

PAI Scale Models

Klasa dbQuery

- Reakcja na błędy – w przypadku błędu połączenia z bazą danych lub w zapytaniu, wykonanie skryptu zostanie przerwane i zostanie wyświetlony komunikat:

SQL query error

SQL message

Unknown column 'county' in 'WHERE'

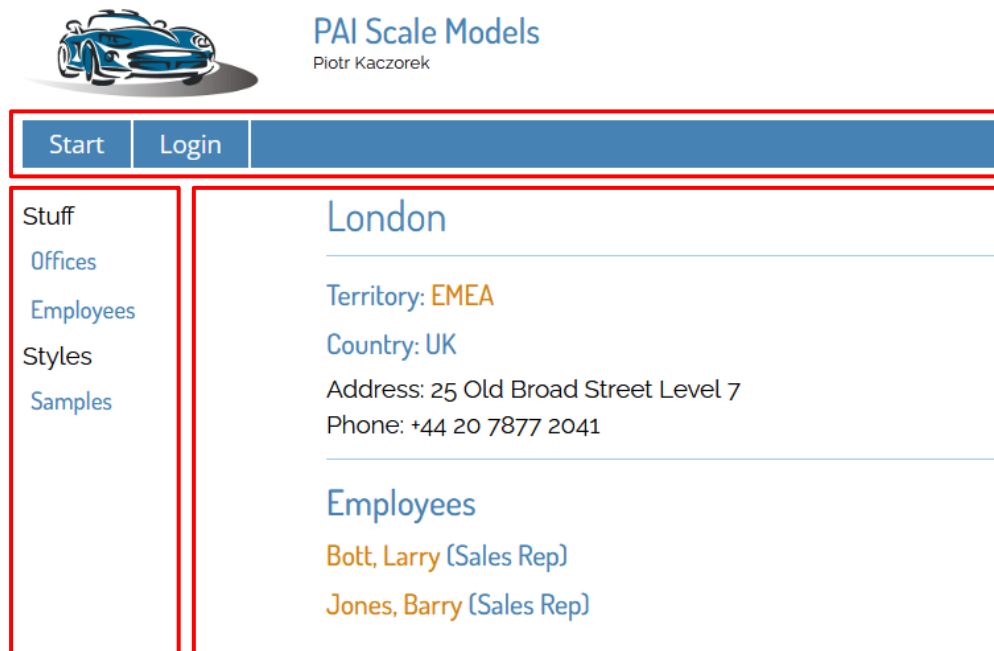
SQL query

```
SELECT employeeNumber, firstName, lastName, city FROM employees JOIN  
offices USING (officeCode) WHERE county='usa' ORDER BY lastName
```

PAI Scale Models

Funkcja Render

- Tworzy gotowy dokument HTML, wykorzystując szablon treści
- Dokument ma trzy zmienne elementy: menu główne (nav), menu boczne (aside) i główną treść (main):



PAI Scale Models

Funkcja getMainMenu

- Menu główne jest tworzone przez funkcję getMainMenu;
Można ją modyfikować, dodając kolejne pozycje;
W funkcji można wykorzystać tablicę \$_SESSION

```
function getMainMenu() {  
    // init  
    $mnu = "";  
    // home  
    $mnu .= getMenuItems("./", "Start");  
    // login/logout  
    $mnu .= getMenuItems("login.php", "Login");  
  
    return $mnu;  
}
```

PAI Scale Models

Funkcja getSideMenu

- Menu boczne jest tworzone przez funkcję getSideMenu; Można ją modyfikować, podobnie jak getMainMenu. Tu także można wykorzystać tablicę \$_SESSION

```
function getSideMenu () {  
    // init  
    $mnu = "";  
    // offices & employees  
    $mnu .= "<li><p>Stuff\n"  
    . getMenuitem("offices.php", "Offices")  
    . getMenuitem("employees.php", "Employees");  
  
    return $mnu;  
}
```

PAI Scale Models

Główna treść

- Główna treść powinna być stopniowo dodawana do jednej zmiennej (np. `$content`), zaczynając od tytułu.
- Kiedy treść jest gotowa, należy przekazać ją funkcji `Render`:

```
session_start();  
require_once 'common.php';  
require_once 'template.php';  
  
$content = "<h1>PAI Scale Models</h1>";  
$content .= "<p>Wersja demo";  
  
Render($content);
```

PAI Scale Models

Dodawanie podstrony

- Dodanie pozycji do menu głównego lub (częściej) bocznego;

```
$mnu .= getMenuItems("./skrypt.php", "Link");
```

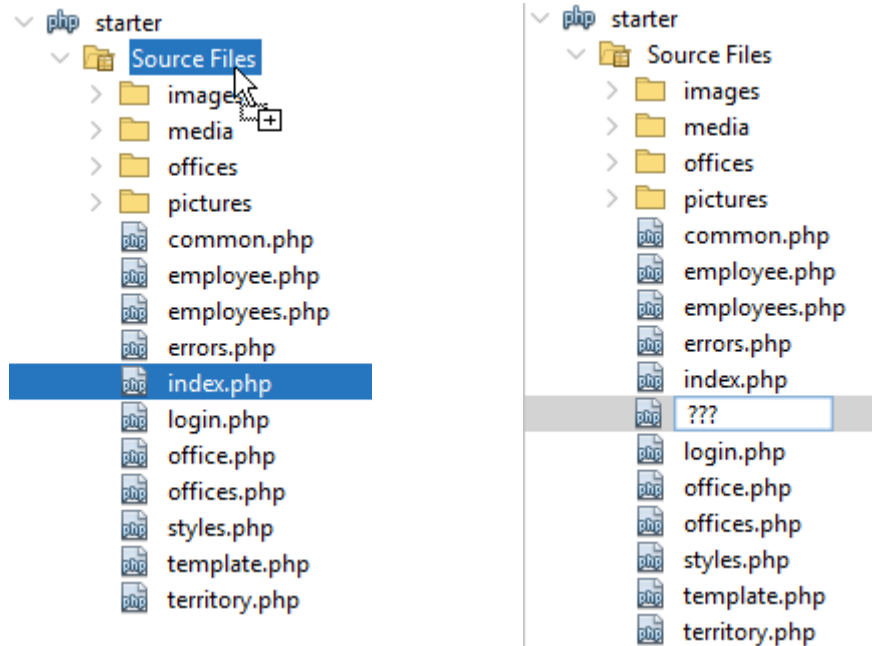
Argumenty funkcji getMenuItems to nazwa skryptu i treść wyświetlana w menu.

Należy pamiętać o użyciu operatora .= zamiast =

PAI Scale Models

Dodawanie podstrony

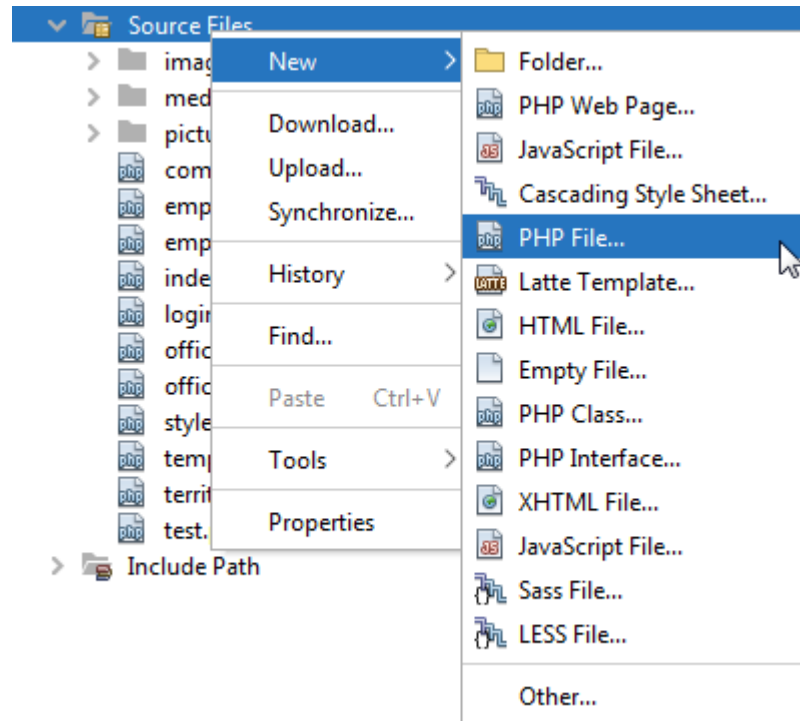
- Dodanie skryptu do projektu – metoda 1
Skopiowanie istniejącego skryptu (przeciągnięcie do folderu „Source Files” z wciśniętym <Ctrl>) i zmiana nazwy (<F2>)



PAI Scale Models

Dodawanie podstrony

- Dodanie skryptu do projektu – metoda 2
Kliknięcie prawym klawiszem myszy "Source Files" w projekcie NetBeans i wybranie polecenia "New,, oraz...



PAI Scale Models

Dodawanie podstrony

- ... skopiowanie do nowego skryptu kodu istniejącej podstrony, łącznie z wywołaniem session_start, aż do Render:

```
<?php
session_start();
require_once 'common.php';
require_once 'template.php';

$content = "<h1>Employees</h1>\n";
$sql = "SELECT * FROM employees";
$query = new dbQuery($sql);
while (($employee = $query->next()) != NULL) {
    $content .= "...";
}
Render($content);
```

PAI Scale Models

Dodawanie podstrony

- Zmiana zapytania SQL i treści dodawanych do contentu, np. pracownicy...

```
$content = "<h1>Employees</h1>\n";  
$sql = "SELECT * FROM employees";  
$query = new dbQuery($sql);  
while (($employee = $query->next()) != NULL) {  
    $content .= "{$employee["firstName"]}";  
}
```

PAI Scale Models

Dodawanie podstrony

- Zmiana zapytania SQL i treści dodawanych do contentu, np. pracownicy zmieniamy na produkty

```
$content = "<h1>Products</h1>\n";
$sql = "SELECT * FROM products";
$query = new dbQuery($sql);
while (( $product = $query->next() ) != NULL) {
    $content .= " { $product[\"productName\"] } ";
}
```

PAI Scale Models

Dodawanie podstrony

- Wykorzystanie wzoru formatowania i zdjęć:

1936 Mercedes-Benz 500K

Vendor: Studio M Art Models

Line: Vintage Cars



- Proszę korzystać ze wzoru, kopiując odpowiedni fragment strony Styles oraz wstawiając w miejsce przykładowych danych dane odczytane z bazy danych:

```
<!-- Product thumbnail -->
<section class='thumb'>
  <div>
    <h2>1936 Mercedes-Benz 500K</h2>
    <p>Vendor: Studio M Art Models
    <p>Line: Vintage Cars
  </div>
  <div>
    <img src='./images/S18_1367.jpg'>
  </div>
</section>
```

PAI Scale Models

Dodawanie podstrony szczegółów

- Przedstawione powyżej kroki wystarczą do utworzenia podstrony z listą – pracowników, produktów, klientów itp. Dodawanie podstrony szczegółów wymaga dodatkowej umiejętności: przekazania o który dokładnie element listy (którego pracownika, produkt, klienta, ...) chodzi
- Robi się to przez link o specjalnym adresie:
`skrypt.php?nazwaId=wartośćId`
np. w przypadku pracowników, w wersji starter:
`employee.php?employeeNumber=1102`
- Skrypt podstrony z listą musi takie linki wygenerować, natomiast skrypt podstrony szczegółów odczytać ID i wstawić je do zapytania SQL

PAI Scale Models

Dodawanie podstrony szczegółów

- Generowanie linku z identyfikatorem:

```
foreach ($rows as $row) {  
    $href = "./skrypt.php?id={$row["id"]}";  
    $name = "{$row["name"]}";  
    $link = "<a href='{$href}'>{$name}</a>";  
}
```

- Ważne:

- Adres skryptu może być bezwzględny
(<http://pollux.umg.edu.pl/~INAZ/cars/skrypt.php>)
lub względny (./skrypt.php), gdzie kropka = ten sam folder
- Po nazwie skryptu pytajnik "?" i para nazwa=wartość
- W łańcuchach ograniczonych cudzysłowami atrybuty należy podawać w apostrofach lub stosować sekwencję ucieczki, \"
- Obowiązuje notacja rozszerzona, {\$wiersz["kolumna"]}
- Proszę zajrzeć do plików offices lub employees po dalsze wskazówki

PAI Scale Models

Dodawanie podstrony szczegółów

- Odczytanie identyfikatora z linku odbywa się tak, jakby identyfikator pochodził z formularza z metodą GET (bo para nazwa=wartość jest w żądaniu, po pytajniku "?")
- Identyfikator należy następnie wstawić do odpowiednio skonstruowanego zapytania SQL:

```
$id = $_GET["id"];  
$sql = "SELECT * FROM tableName "  
      . "WHERE id='{ $id } '";
```

- Ważne:
 - W SQL wartości należy podawać w apostrofach: `id='{ $id } '`
 - Proszę zajrzeć do plików office lub employee po dalsze wskazówki

PAI Scale Models

Jak zacząć?

1. Pobrać ze strony PAI plik starter.zip,
2. Rozpakować albo skopiować zawartość do folderu,
3. Utworzyć w NetBeans projekt, w nowym folderze;
Uwaga! Należy zachować projekt HTML/CSS
oraz ewentualnie inne projekty do sprawdzenia,
4. Po standardowym sprawdzeniu (echo "Działa?"),
z projektu NetBeans usunąć plik index.php,
5. Metodą drag & drop wgrać zawartość rozpakowanego folderu
starter do wirtualnego folderu "Source files" projektu;
Uwaga! Nie da się tego zrobić wprost z archiwum,
6. Kliknąć "Play" (skrót <F6>),
7. Poprawić błędy w kodzie PHP, wrócić do punktu 6.

PAI Scale Models

Zadania

- Wyświetlanie różnych dostępnych w bazie danych informacji oraz pokazanie powiązań między nimi:
 - Informacja o modelach:
 - » Lista modeli z linkami do szczegółów produktu
 - » Paginacja listy modeli
 - » Szczegóły modelu, z linkami do kategorii i producenta
 - » Lista kategorii, opcjonalnie z przykładowymi (np. losowo wybranymi) modelami
 - » Szczegóły kategorii, z opisem i listą wszystkich modeli
 - » Wyszukiwarka modeli

PAI Scale Models

Zadania

- Wyświetlanie różnych dostępnych w bazie danych informacji oraz pokazanie powiązań między nimi:
 - ...
 - Informacja o klientach i zamówieniach:
 - » Lista klientów z linkami do szczegółów klienta
 - » Szczegóły klienta, z listą zamówień
 - » Szczegóły zamówienia, z listą zamówionych modeli
 - » Wyszukiwarka klientów, np. na podstawie nazwy albo kraju
 - » Ranking klientów, np. wg zamówień z ostatnich 12 miesięcy
 - » Lista zamówień w trakcie realizacji
 - » Lista zamówień (z kwotami) i płatności

PAI Scale Models

Zadania

- Wyświetlanie różnych dostępnych w bazie danych informacji oraz pokazanie powiązań między nimi:
 - ...
 - Informacja o pracownikach (uzupełnienie)
 - » Lista klientów, którymi opiekuje się pracownik
 - » Lista najlepszych pracowników, np. wg zamówień przypisanych im klientów, z ostatnich 12 miesięcy (przypisanych klientów mają tylko „Sales Rep”)

PAI Scale Models

Zadania

- Zarządzanie danymi w bazie danych:
 - Edycja wybranych danych modeli lub klientów (np. nazwa, opis i cena w przypadku modeli)
 - Edycja statusu zamówień
 - Dodawanie nowych modeli, klientów lub pracowników
 - Dodawanie płatności

Implementacja edycji lub dodawania danych ma sens tylko wtedy, kiedy te dane można potem zobaczyć w innej części serwisu

PAI Scale Models

Zadania

- Uwierzytelnianie i uprawnienia użytkowników:
 - Implementacja logowania i wylogowania
(wykorzystanie formularzy oraz mechanizmu sesji)
 - Udostępnianie różnych części serwisu,
zależnie od uprawnień użytkownika:
 - » Użytkownik niezalogowany i zalogowany
 - » Użytkownik zalogowany, zależnie od roli

PAI Scale Models

Zadania

- Wszystkie strony powinny być powiązane, jak tylko się da, aby można było po nich wygodnie przechodzić (np. lista modeli > model > kategoria > inny model > producent > inny model > ... itp.)
- Sposób powiązania różnych stron został pokazany na przykładzie stron o pracownikach (pracownik > biuro > obszar > biuro > pracownik > ...)
- Zadania można rozwijać dowolnie długo, ulepszając istniejące funkcje (np. filtry wyszukiwania) lub dodając nowe (np. koszyk zakupów i składanie zamówień). Niektóre z tych funkcji można znaleźć w wersji wzorcowej, dostępnej na stronie WWW