

Wzorzec MVC w tworzeniu aplikacji internetowych

Laboratorium 2

Ćwiczenia

Zadania znajdować będą się w przeznaczonych im katalogach, tzn. dla **zadanie 1** katalog powinien nazywać się **zad1**. Każde zadanie będzie zawierać potrzebną na jego potrzeby strukturę katalogów oraz plików. Utworzone pliki, muszą mieć odpowiedni format. W większości przypadków, będzie to format **.js**.

Zadanie 1

Stwórz nowy projekt **Node.JS** a następnie utwórz w nim plik **app** oraz katalog **views**. Plik **app** powinien zawierać:

- stałą **PORT** o wartości **3000**,
- prosty serwer aplikacji nasłuchujący na porcie **3000**,
- **listeningListener** tego serwera, wyświetlać będzie za pomocą **console.log**, treść „**Server is running on \${PORT}**”,
- stwórz przykładowy routing, który będzie przekierowywał na odpowiednie widoki z katalogu **views** (uwzględnij ścieżkę „/” oraz „**student**”).

Katalog **views** powinien zawierać dwa pliki – **home** oraz **student**. Oba pliki powinny zawierać metodę **renderPage**, która będzie zwracała unikalny dla nich, szablonowy widok strony.

```
<html lang="pl">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>NAZWA_PLIKU</title>
    <body>
      <div>NAZWA_PLIKU</div>
    </body>
  </head>
</html>
```

Zadanie 2

Przerób metodę **renderPage** w pliku **home** tak by zwracał prosty formularz z przyciskiem oraz z następującymi polami:

- imię (atrybuty: typ - tekstowy, name - name),
- nazwisko (atrybuty: typ – tekstowy, name – lastname),
- wiek (atrybuty: typ – numeryczny, name – age),
- płeć (lista wyboru: male, female, other oraz atrybuty: name – gender),
- numer albumu (atrybuty: typ – numeryczny, name – code),
- kierunek (atrybuty: typ – tekstowy, name – studyField).

Zadanie 3

Dodaj przypadek, gdy podany **url** nie istnieje. W przypadku zaistnienia takiej sytuacji:

- zwrócony powinien zostać nagłówek o kodzie **404** i **Content-Type** o wartości **text/html**,
- zwrócona treść „**404 Not Found**”.

Dodatkowo, utwórz katalog **routes** a w nim plik **index**, który:

- będzie posiadał metodę **handleHome**, która obsłuży zwrócenie **renderPage** z katalogu **views** z pliku **home**,
- będzie posiadał metodę **handleStudent**, która obsłuży zwrócenie **renderPage** z katalogu **views** z pliku **student**.

Dodatkowo, obsłuż działanie formularza tak, by następowało przekierowanie z wykorzystaniem metody **POST** na ścieżkę **/student**.

Zadanie 4

Przerób metodę **renderPage** w pliku **student** tak by zwracał prosty układ z danymi:

- w nagłówku **h1** tekst **Student profile**,
- w **paragrafach** (jeden na jedną informację) następujące dane w następującej kolejności: numer albumu, imię, nazwisko, płeć, wiek, kierunek,
- każdy paragraf powinien zawierać treść w formie: **NAZWA_INFORMACJI: \${WARTOŚĆ_INFORMACJI}**, np. **Age: 27**,
- wyświetlane dane powinny być pobierane z danych przekazanych z formularza.

Przekazane dane formularza, powinny dodatkowo być zapisane w pliku o rozszerzeniu **.txt** a nazwie **numer albumu**, np. **12345.txt**.