|  |  |  |
| --- | --- | --- |
| **AI1** | Ratańczuk Damian  Album 53964  Grupa 3 | **Wersja 1** |
|  |
| **LAB E** |

TypeScript

# Spis treści

[Spis treści 1](#_Toc183675362)

[Cel zajęć 1](#_Toc183675363)

[Rozpoczęcie 1](#_Toc183675364)

[Uwaga 1](#_Toc183675365)

[Wymagania 2](#_Toc183675366)

[Instalacja Node.js 2](#_Toc183675367)

[Konfiguracja projektu 2](#_Toc183675368)

[Dynamicznie podłączany styl 3](#_Toc183675369)

[Dynamicznie tworzony obszar z linkami 4](#_Toc183675370)

[Commit projektu do GIT 5](#_Toc183675371)

[Podsumowanie 5](#_Toc183675372)

# Cel zajęć

Celem głównym zajęć jest zdobycie następujących umiejętności:

* konfiguracja środowiska do programowania i kompilacji aplikacji z wykorzystaniem TypeScript
* budowa skryptów z wykorzystaniem języka TypeScript

W praktycznym wymiarze uczestnicy zmodyfikują witrynę z LAB A (CSS Garden) do postaci single-page application (SPA), w której style będą wczytywane dynamicznie za pomocą skryptów napisanych w TypeScript.

# Rozpoczęcie

Rozpoczęcie zajęć. Powtórzenie podstaw składni TypeScript.

Wejściówka?

# Uwaga

Ten dokument aktywnie wykorzystuje niestandardowe właściwości. Podobnie jak w LAB A wejdź do Plik -> Informacje -> Właściwości -> Właściwości zaawansowane -> Niestandardowe i zaktualizuj pola. Następnie uruchom ten dokument ponownie lub Ctrl+A -> F9.

# Wymagania

W ramach LAB E zmodyfikowany zostanie kod z LAB A:

* w przeciwieństwie do pierwotnej postaci, witryna ma teraz składać się z jednej strony index.html oraz wielu styli CSS;
* do strony podpięty jest skrypt JS z zewnętrznego pliku budowanego za pomocą webpacka
* w skrypcie napisanym w TS przechowywany jest stan aplikacji:
  + nazwa bieżącego stylu i jego plik;
  + słownik dostępnych stylów i ich plików;
* fragment strony zawierający linki do stron w innym stylu zostają zastąpione przez linki z wywołaniami aplikacji w TS;
* po wywołaniu aplikacji z linka, z DOM usunięte zostaje odwołanie do starego stylu CSS, a dodane zostaje odwołanie do stylu CSS powiązanego w słowniku aplikacji z wybranym stylem

Prowadzący omówi powyższe wymagania. Upewnij się, czy wszystko rozumiesz.

Tu umieść swoje notatki:

…notatki…

# Instalacja Node.js

Wejdź na stronę <https://nodejs.org/en/download/current>. Pobierz wersję Current -> Windows Binary (.zip) -> 64-bit. Rozpakuj archiwum do I:\node. Dodaj ten folder do zmiennej środowiskowej Path użytkownika.

Uruchom nowe okno wiersza poleceń. Wykonaj polecenie npm -v. Powinno zadziałać – udało się zainstalować NODE i NPM lokalnie dla użytkownika. Jeśli nie działa w Twoim terminalu, spróbuj w innym (CMD, PowerShell) albo wyloguj się i zaloguj ponownie (<https://ispot.link/hyttiooa>).

# Konfiguracja projektu

Powiel projekt z LAB A. Zaktualizuj strukturę projektu:

* pozostaw pojedynczy plik HTML
* pliki CSS umieść w osobnym podkatalogu
* utwórz plik script.ts o zawartości:

const msg: string = "Hello!";

alert(msg);

Do pliku HTML podłącz plik dist/script.js:

<head>

...

<script src="dist/script.js" defer></script>

</head>

Zainicjalizuj pakiet NPM i zainstaluj paczkę mix oraz zależności:

npm install --save-dev laravel-mix ts-loader typescript

npx tsc --init

Utwórz plik webpack.mix.js:

let mix = require('laravel-mix');

mix.ts('script.ts', 'dist').setPublicPath('dist');

Włącz kompilację ciągłą:

npx mix watch

Uruchom stronę w przeglądarce.

Wstaw zrzut ekranu komunikatu o sukcesie kompilacji:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Wstaw zrzut ekranu zbudowanej strony:

Obraz zawierający tekst, zrzut ekranu, Fast food, jedzenie

Opis wygenerowany automatycznie

|  |  |  |
| --- | --- | --- |
| Punkty: | 0 | 1 |

# Dynamicznie podłączany styl

Wstaw zrzut ekranu kodu odpowiedzialnego za dynamiczne podłączanie stylu CSS:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

|  |  |  |
| --- | --- | --- |
| Punkty: | 0 | 1 |

# 

# Dynamicznie tworzony obszar z linkami

Wstaw zrzut ekranu kodu odpowiedzialnego za dynamiczne generowanie linków do styli:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Wstaw zrzut ekranu obszaru linków na stronie:



Wstaw zrzut ekranu obszaru linków po dodaniu kolejnego stylu do tablicy stylów:



|  |  |  |
| --- | --- | --- |
| Punkty: | 0 | 1 |

Wstaw zrzut ekranu kodu odpowiedzialnego za zmianę stylu po kliknięciu na link:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Wstaw zrzut ekranu strony po kliknięciu na link zmiany stylu:

Obraz zawierający tekst, w pomieszczeniu, Blat, ściana

Opis wygenerowany automatycznie

Wstaw zrzuty ekranu fragmentu kodu strony (narzędzia developerskie) z podłączonymi linkami przed i po zmianie stylu:

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

|  |  |  |
| --- | --- | --- |
| Punkty: | 0 | 1 |

# Commit projektu do GIT

Zacommituj i pushnij swoje rozwiązanie do repozytorium GIT.

Upewnij się, czy wszystko dobrze się wysłało. Jeśli tak, to z poziomu przeglądarki utwórz branch o nazwie lab-e na podstawie głównej gałęzi kodu.

Podaj link do brancha lab-e w swoim repozytorium:

<https://github.com/Xomil/Aplikacje-Internetowe-1/tree/main/lab-e>

# Podsumowanie

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

…podsumowanie…

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.