|  |  |  |
| --- | --- | --- |
| **AI1** | Drobotushchenko Maksym  Album 53952  Grupa 331 | **Wersja 2025-09-26** |
|  |
| **LAB E** |

TypeScript

# Spis treści

[Spis treści 1](#_Toc209822995)

[Cel zajęć 1](#_Toc209822996)

[Rozpoczęcie 1](#_Toc209822997)

[Uwaga 1](#_Toc209822998)

[Wymagania 2](#_Toc209822999)

[Instalacja Node.js 2](#_Toc209823000)

[Konfiguracja projektu 2](#_Toc209823001)

[Dynamicznie podłączany styl 3](#_Toc209823002)

[Dynamicznie tworzony obszar z linkami 4](#_Toc209823003)

[Commit projektu do GIT 5](#_Toc209823004)

[Podsumowanie 5](#_Toc209823005)

# 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:

* witryna ma teraz składać się z jednej strony index.html oraz wielu styli CSS;
* do strony podpięty jest skrypt TS budowany za pomocą VITE;
* w skrypcie napisanym w TS przechowywany jest stan aplikacji:
  + nazwa bieżącego stylu i jego plik;
  + słownik dostępnych stylów i ich plików;
* na stronie musi zostać dynamicznie umieszczony fragment z linkami do stron w innych stylach, obsłużonymi przez wywołania 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 x64 -> Standalone Binary (.zip). Rozpakuj archiwum do katalogu pobranych, przykładowo C:\Users\akarczmarczyk\Downloads\node-v24.9.0-win-x64. Dodaj ten folder do zmiennej środowiskowej Path użytkownika.

**Uwaga!** Rozpakowywanie node do pobranych to zła praktyka. Natomiast w związku z problemem z wydajnością node’a na zasobie zdalnym I:, musimy upchnąć binarki gdzieś na dysku C. Ważne – katalogi na dysku C nie mają gwarantowanych kopii zapasowych.

Uruchom nowe okno wiersza poleceń. Wykonaj polecenia node -v i 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

Utwórz katalog nowego projektu, przykładowo lab-e. Wejdź do niego terminalem. Wykonaj polecenie:

npm create vite@8.0.1 . -- --template vanilla-ts

W efekcie wykorzystane zostanie create-vite we wskazanej wersji, żeby w bieżącym katalogu (.) zainstalować zależności niezbędne do utworzenia aplikacji w tzw. vanilla TS. Na pytanie o rolldown-vite odpowiedz No. Na pytanie Install (…) odpowiedz No.

Wykasuj wszystko z wyjątkiem: .gitignore, package.json, tsconfig.json.

Z LAB A przenieś index.html i umieść w głównym katalogu nowego projektu. Zakomentuj w pliku skrypt podmieniający style z LAB A.

Swoje pliki CSS umieść w katalogu public.

W katalogu src utwórz plik script.ts o zawartości:

const msg: string = "Hello!";

alert(msg);

Do pliku HTML podłącz plik src/script.ts:

<head>

...

<script src="src/script.ts" type="module" defer></script>

</head>

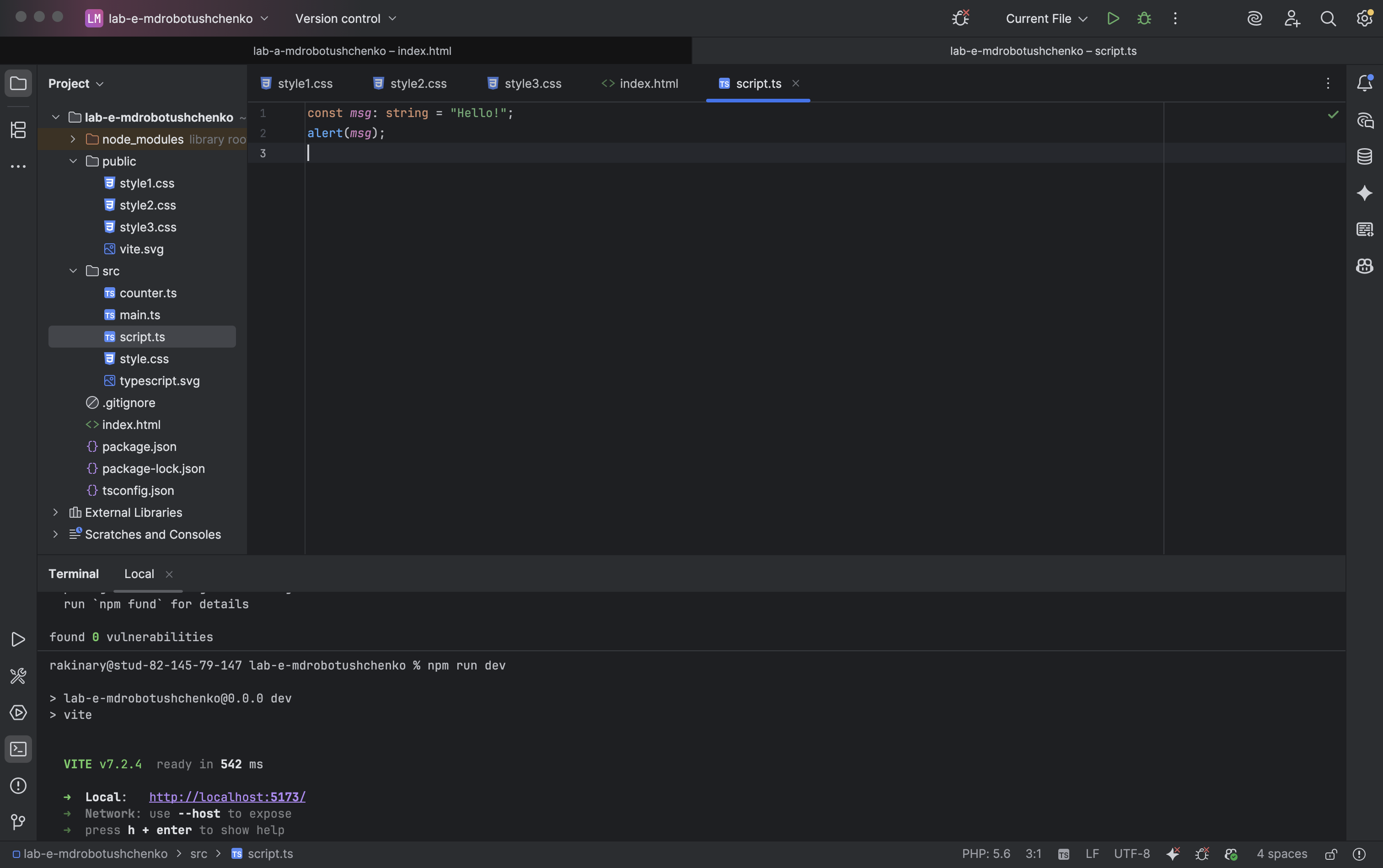
Dokonaj instalacji pakietów NPM i uruchom aplikację:

npm install

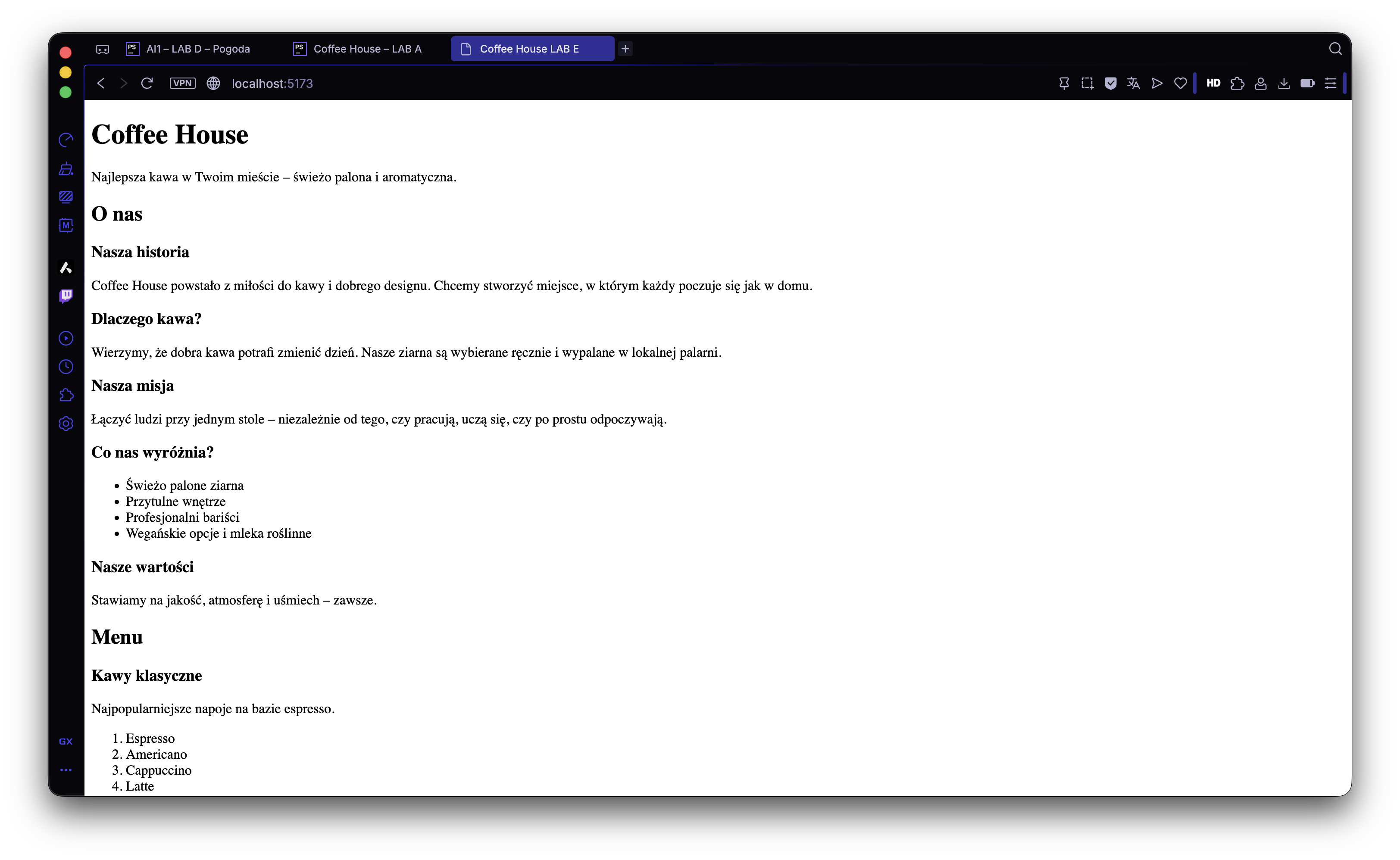
npm run dev

W efekcie wyświetli się adres strony w trybie deweloperskim. Wyświetl stronę w przeglądarce. Powinien pojawić się alert o treści „Hello!”.

Wstaw zrzut ekranu komunikatu terminala o sukcesie kompilacji:



Wstaw zrzut ekranu zbudowanej strony:

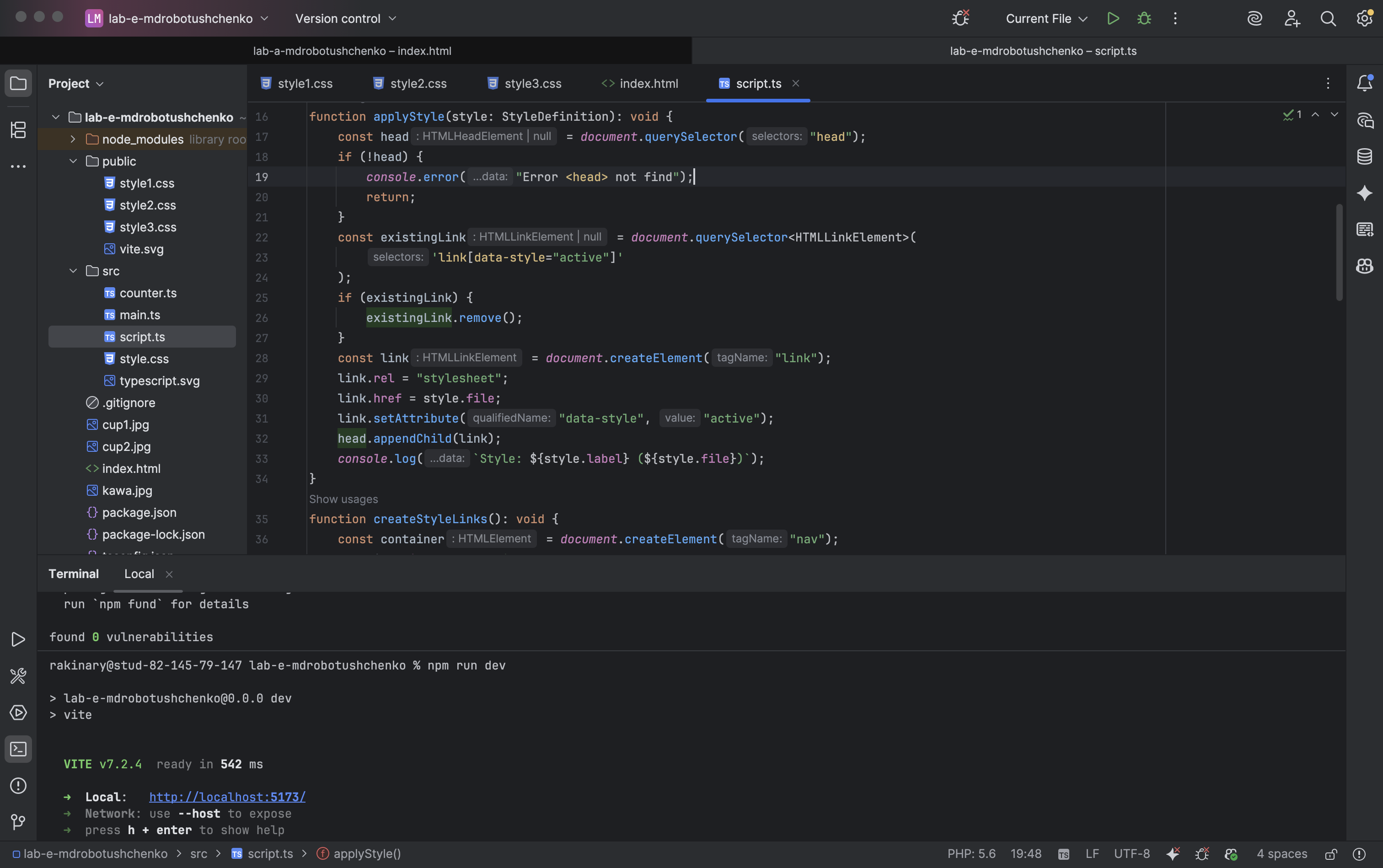
 

|  |  |  |
| --- | --- | --- |
| Punkty: | 0 | 1 |

Przeprowadź implementację swojego rozwiązania zgodnie z wymaganiami. Konsultuj się z innymi uczestnikami laboratorium. Zapoznaj się z kolejnymi sekcjami, żeby wiedzieć, jakie zrzuty ekranu będą potrzebne. Uzupełnij sprawozdanie.

# Dynamicznie podłączany styl

Wstaw zrzut ekranu kodu odpowiedzialnego za dynamiczne podłączanie stylu CSS:

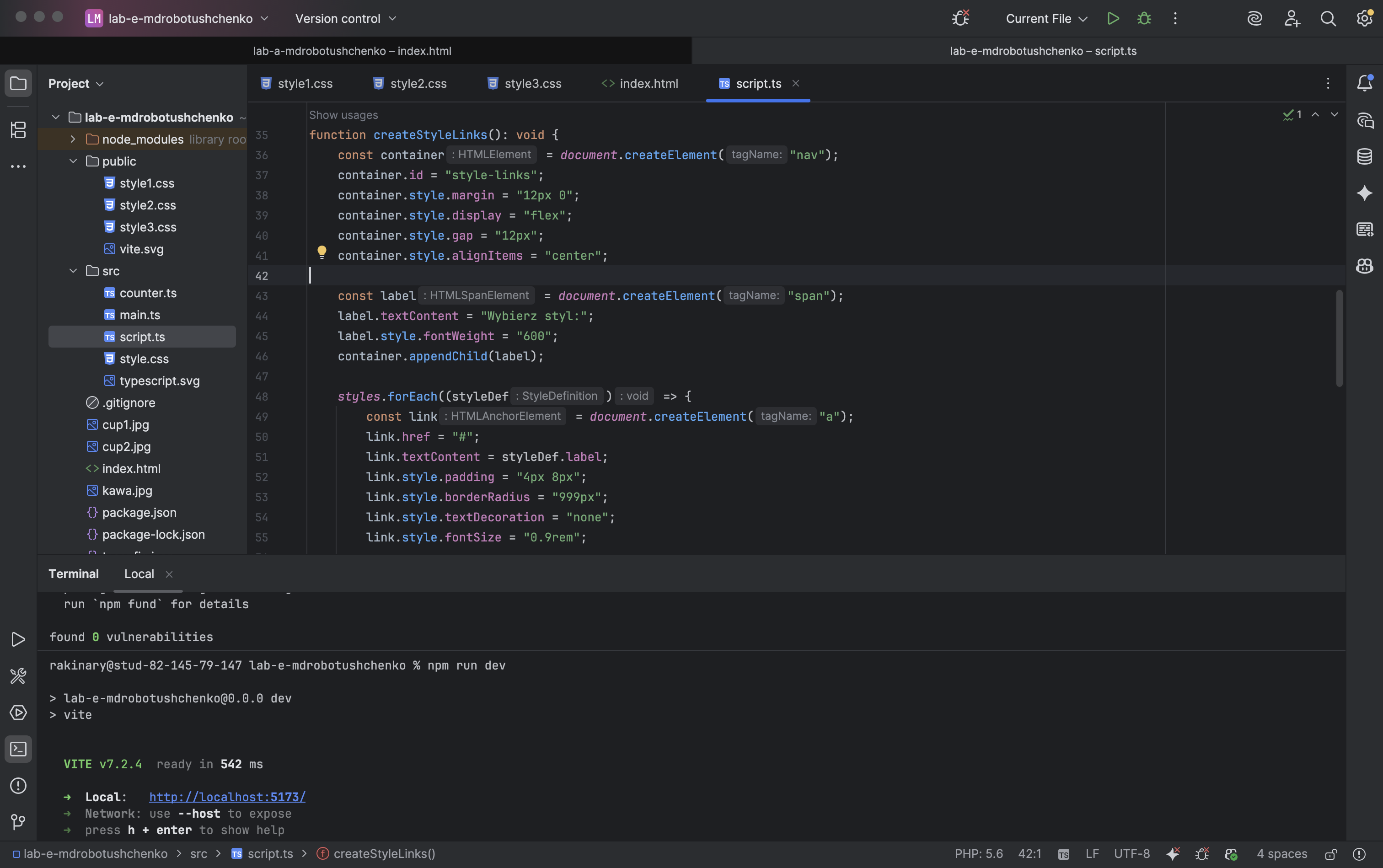


|  |  |  |
| --- | --- | --- |
| Punkty: | 0 | 1 |

# 

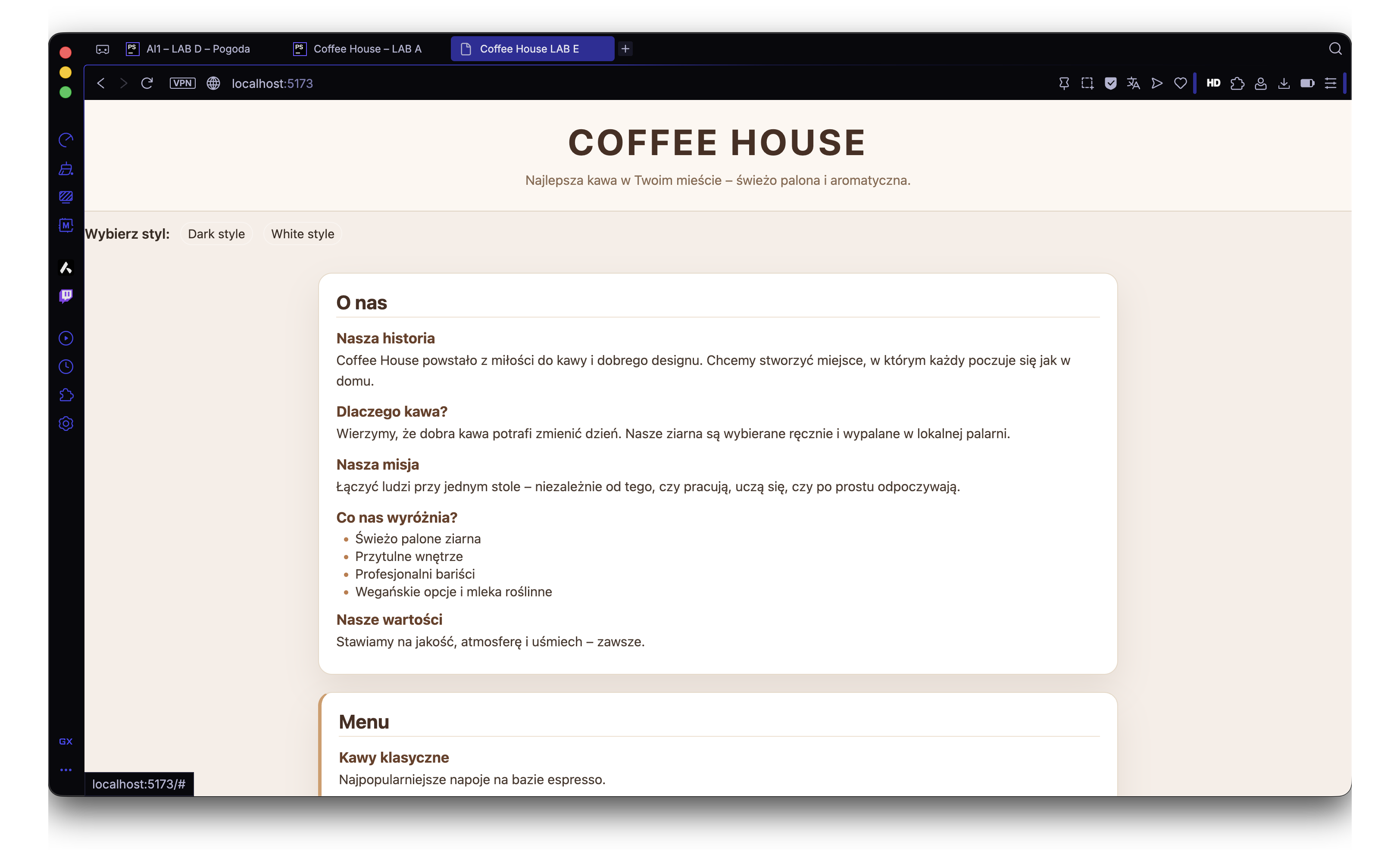
# Dynamicznie tworzony obszar z linkami

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



Wstaw zrzut ekranu obszaru linków na stronie: dwa linki te co wcześniej





Wstaw zrzut ekranu obszaru linków po dodaniu kolejnego stylu do tablicy stylów: a tu trzy style



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

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



Wstaw zrzut ekranu strony po kliknięciu na link zmiany stylu: ten trzeci styl



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

|  |  |  |
| --- | --- | --- |
| 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:

…link, np. <https://github.com/inazwisko/ai1-lab/tree/lab-e…>

# Podsumowanie

W kilku słowach/zdaniach napisz swoje przemyślenia odnośnie tego laboratorium. Nie używaj LLM.

…podsumowanie…

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.