Imię i nazwisko	Grupa
Klaudia Szeniszewska	135NCI A
Tytuł ćwiczenia **4 i 9**	Frameworki front-endowe Wielowątkowość w aplikacjach WWW - Web Workers
Git Hub Strona WWW	https://github.com/keszfeayym/ORW.git http://klaudiaszeniszewska.cba.pl/wordpress/

1) Cel ćwiczenia:

a) Prezentacja możliwości działającego wybranego przeze mnie framework'a i obiektów Web Workers w rozwiązaniu problemu obliczeniowego.

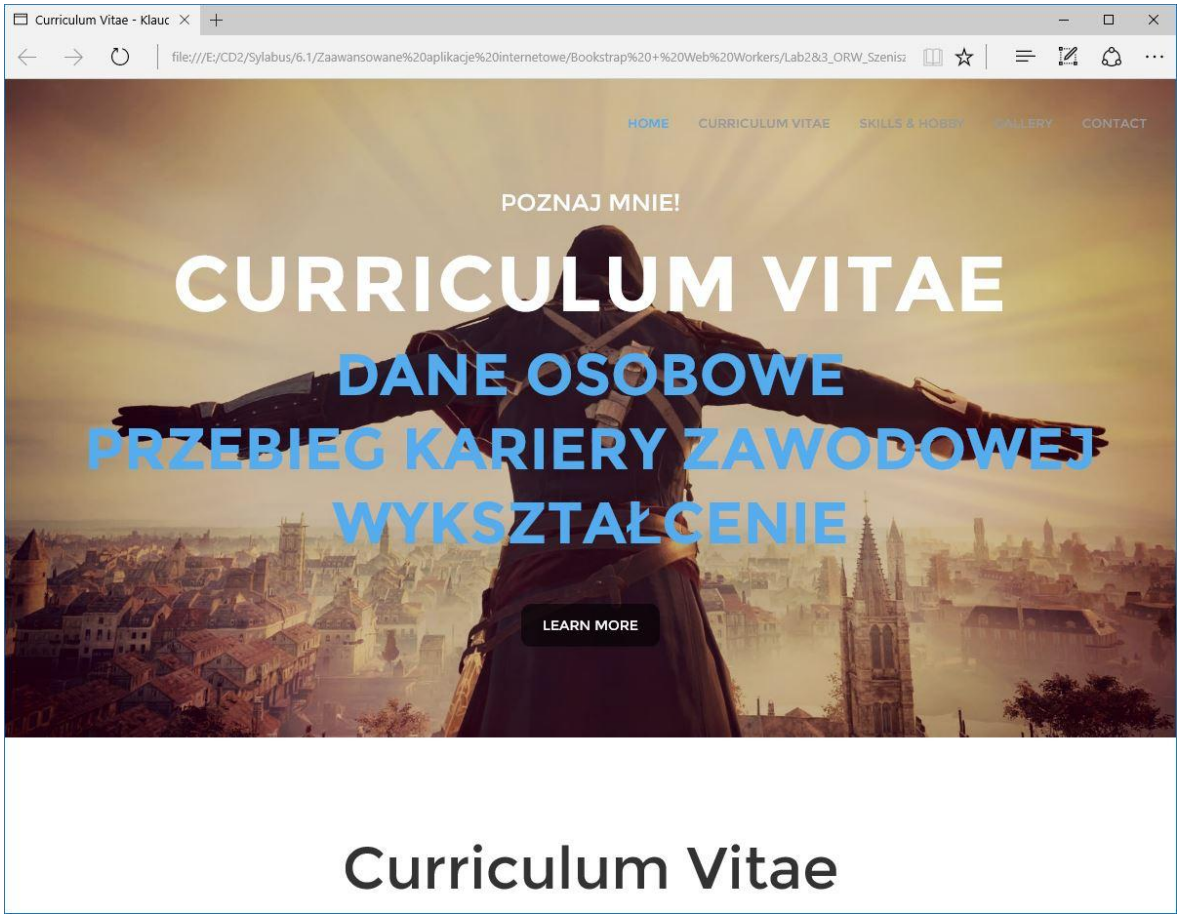
2) Przebieg ćwiczenia:

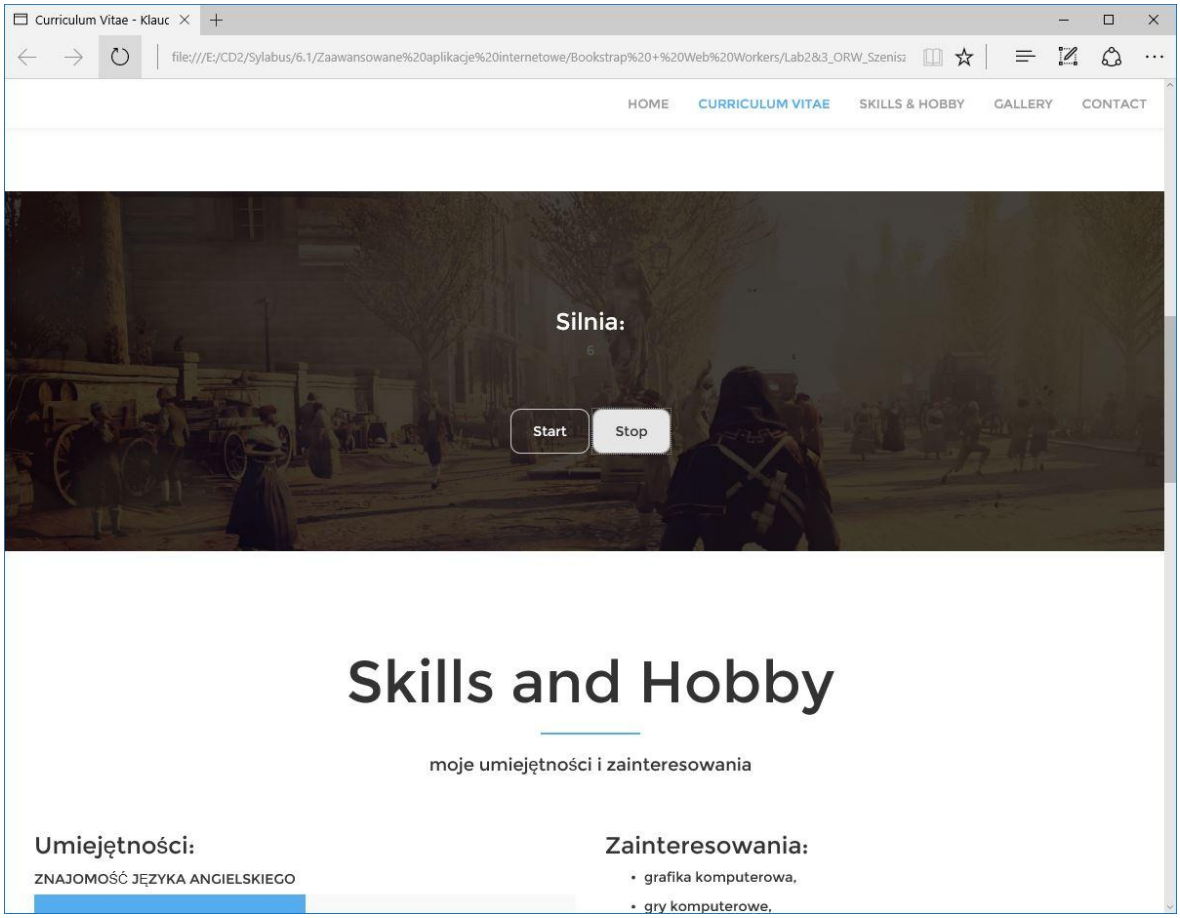
a) Wpierw zapoznałam się z instrukcją zleconych zadań, decydując się na realizację laboratorium numeru 4 i 9, mianowicie „Frameworki front-endowe" i „Wielowątkowość w aplikacjach WWW - Web Workers".

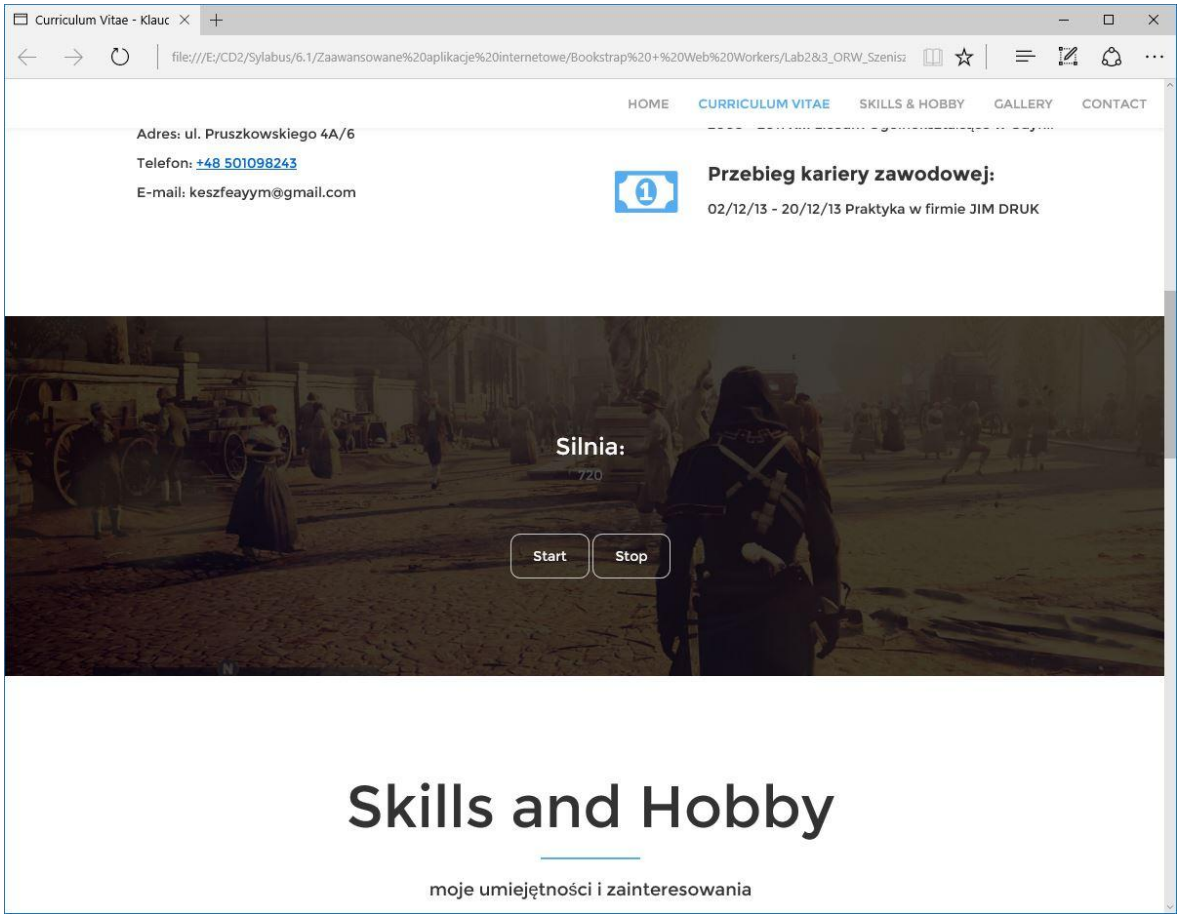
b) Zgodnie z powyższym, najpierw skorzystałam z „Bootstrap", pobierając jego pliki i modyfikując je w celach dostosowania do urzeczywistnienia wybranego wyglądu.

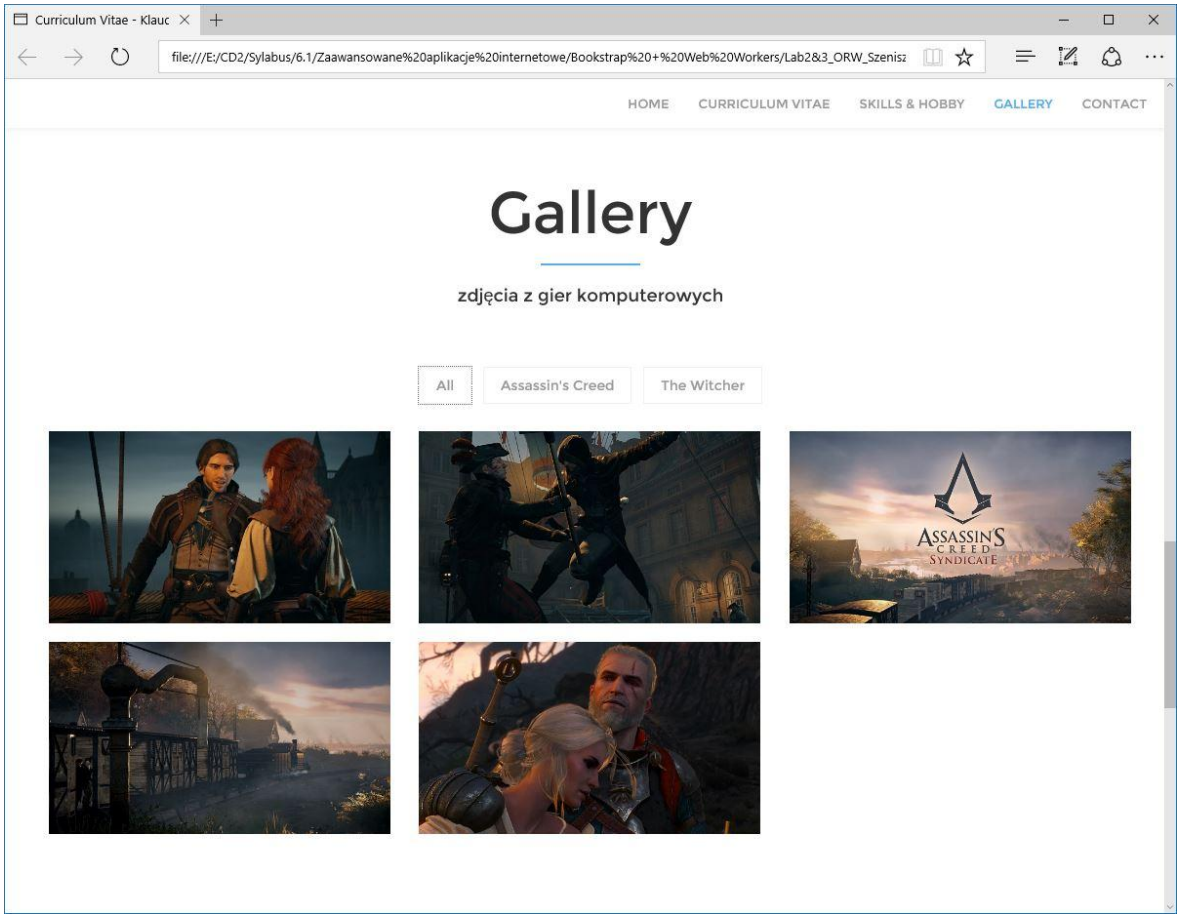
c) Dalej, rozpoczęłam pracę nad wybraniem odpowiedniego problemu obliczeniowego, aż zdecydowałam się na wyliczenie silni. Zgodnie z tym, nakreśliłam skrypt w Java Script (silnia), a także w pliku index.html dodałam kod odpowiedzialny za działanie i wyświetlanie Web Workers.

d) Całość prezentuje się następująco, wraz z zademonstrowaniem zmieniających się w liczb, a dokładniej zwiększającej się silni. Strona nie jest cudem techniczny, zawiera też nie wiele treści, gdyż tylko moje dane osobowe, a także umiejętności i zainteresowania, czy niewielką galerię zdjęć i formularz do kontaktu.


Curriculum Vitae - Klauc
file:///E:/CD2/Sylabus/6.1/Zaawansowane%20aplikacje%20internetowe/Bookstrap%20+%20Web%20Workers/Lab2&3_ORW_Szenisz
HOME
CURRICULUM VITAE
SKILLS & HOBBY
GALLERY
CONTACT
POZNAJ MNIE!
CURRICULUM VITAE
DANE OSOBOWE
PRZEBIEG KARIERY ZAWODOWEJ
WYKSZTAŁCENIE
LEARN MORE
Curriculum Vitae


Curriculum Vitae - Klauc
file:///E:/CD2/Sylabus/6.1/Zaawansowane%20aplikacje%20internetowe/Bookstrap%20+%20Web%20Workers/Lab2&3_ORW_Szenisz
HOME
CURRICULUM VITAE
SKILLS & HOBBY
GALLERY
CONTACT
Silnia:
Start
Stop
Skills and Hobby
moje umiejętności i zainteresowania
Umiejętności:
ZNAJOMOŚĆ JĘZYKA ANGIELSKIEGO
Zainteresowania:
• grafika komputerowa,
• gry komputerowe,


Curriculum Vitae - Klauc
file:///E:/CD2/Sylabus/6.1/Zaawansowane%20aplikacje%20internetowe/Bookstrap%20+%20Web%20Workers/Lab2&3_ORW_Szenisz
HOME
CURRICULUM VITAE
SKILLS & HOBBY
GALLERY
CONTACT
Adres: ul. Pruszkowskiego 4A/6
Telefon: +48 501098243
E-mail: keszfeayym@gmail.com
Przebieg kariery zawodowej:
02/12/13 - 20/12/13 Praktyka w firmie JIM DRUK
Silnia:
Start
Stop
Skills and Hobby
moje umiejętności i zainteresowania


Curriculum Vitae - Klauc
file:///E:/CD2/Sylabus/6.1/Zaawansowane%20aplikacje%20internetowe/Bookstrap%20+%20Web%20Workers/Lab2&3_ORW_Szenisz
HOME
CURRICULUM VITAE
SKILLS & HOBBY
GALLERY
CONTACT
Gallery
zdjęcia z gier komputerowych
All
Assassin's Creed
The Witcher
ASSASSIN'S CREED SYNDICATE

3) Wnioski:

a) Bootstrap - framework CSS, rozwijany przez programistów Twittera, wydawany na licencji MIT[1]. Zawiera zestaw przydatnych narzędzi ułatwiających tworzenie interfejsu graficznego stron oraz aplikacji internetowych. Bazuje głównie na gotowych rozwiązaniach HTML oraz CSS (kompilowanych z plików Less[2]) i może być stosowana m.in. do stylizacji takich elementów jak teksty, formularze, przyciski, wykresy, nawigacje i innych komponentów wyświetlanych na stronie. Framework korzysta także z języka JavaScript.

b) MVVM jest to przede wszystkim wzorzec programowania i jak każdy wzorzec jest rozwiązaniem pewnej dziedziny problemów. MVVM należy do kategorii wzorców prezentacyjnych i został stworzony przez Microsoft w 200 5 roku. Jego unikalną cechą jest wykorzystywanie mechanizmu “wiązań” (and. binding) do celów prezentacji / manipulacji na danych. Jest to zalecane podejście dla programistów WPF/Silverlight ze względu na wbudowane wsparcie dla wzorca.

   i) Zalety (korzyści):

      (1) Testowalność – ze względu na separacji UI (View) od logiki UI (ViewModel), możemy w łatwy sposób testować kod.

      (2) Przejrzystość - jest to subiektywne, ale kod stworzony w ten sposób poprostu łatwo czytać.

      (3) Łatwa zmiana widoków – mając gotowy ViewModel możemy w łatwy sposób tworzyć nowe widoki np. w Blend.

      (4) Zmiana technologii – dla dobrze napisanego ViewModelu, możemy dodawać widoki w Silverlight, WPF, Windows 8 Modern.

   ii) Wady:

      (1) Spory narzut – szczególnie na początku pracy z wzorcem.

      (2) Obejścia – nie jest to właściwie wada, ale z mojego doświadczenia wynika, że programiści czasami z braku czasu / chęci piszą kod częściowo w mvvm, a częściowo w CD (Code Behind) i powstają różne hybrydy.

      (3) Wsparcie starszych kontrolek – niestety w rzeczywistych projektach istnieją starsze kontrolki (nie można wszystkiego refaktorować ..) i trzeba się nagłowić aby działały zgodnie z filozofią MVVM.

c) Architektura MVC – Architektura Model View Controller jest obecnie jednym z najpopularniejszych wzorców projektowych używanych do projektowania zaawansowanych aplikacji biznesowych. Ideą stosowania architektury MVC jest rozdzielenie logiki biznesowej od warstwy prezentacji. Możemy tego dokonać rozdzielając kod źródłowy na poszczególne elementy. Aby to uczynić, musimy sięgnąć po trzeci element - strukturę kontrolera, który będzie odpowiedzialny za obsługę nadchodzących żądań. Pełna lista elementów architektury prezentuje się następująco:

i) Kontroler (z ang. Controller) - element odpowiedzialny za obsługę żądań przychodzących od użytkowników.

ii) Model (z ang. Model) - element odpowiedzialny za obsługę logiki biznesowej.

iii) Widok (z ang. View) - element odpowiedzialny za obsługę warstwy prezentacji.

Dzięki rozdzieleniu aplikacji na warstwy, w znaczący sposób ułatwiamy rozwój aplikacji. Od tej pory programiści i projektanci nie będą sobie wchodzić w drogę i każda z osób tworząca zespół projektowy będzie odpowiedzialna za ściśle określony fragment aplikacji.

d) Dedykowane wątki robocze (Web Workers) zapewniają prosty sposób na uruchamiania skryptów w postaci wątków w tle treści internetowych. Po utworzeniu, wątek roboczy może przekazywać informacje do menedżera zadań, poprzez wysyłanie wiadomości do procesu obsługi zdarzeń, określonego przez twórcę. Jednak działają one w kontekście globalnym, który różni się od kontekstu bieżącego okna (wywoływanego za pomocą skrótu window, zamiast self, dlatego próba otrzymania obecnego zakresu globalnego komendą Worker zwróci błąd).

   Wątki robocze mogą wykonywać zadania bez interakcji z interfejsem użytkownika. Ponadto, mogą one obsługiwać wejście/wyjście używając XMLHttpRequest (chociaż responseXML i właściwości channel będą zawsze równe null).

e) Tworzenie nowego wątku jest proste. Wszystko, co musisz zrobić, to wywołać konstruktor Worker(), podając URI skryptu do wykonania w wątku roboczym, a jeśli chcesz być w stanie otrzymywać powiadomienia od wątku, ustawić właściwości wątku Worker.onmessage na obsługę odpowiedniej funkcji zdarzenia.