

## KARTA PROJEKTU PRACOWNIA PROGRAMOWANIA

| PODSTAWOWE DANE                      |  |
|--------------------------------------|--|
| IMIĘ I NAZWISKO                      | <b>Kacper Maj</b>  |
| NUMER INDEKSU                        | <b>457990</b>  |
| GRUPA                                | <b>2</b>   |
| NUMER WERSJI                         | <b>1</b>   |
| INFORMACJE O PROJEKCIE               |  |
| NAZWA HANDLOWA PROJEKTU              | <b>Watch With Friends</b>  |
| ADRES REPOZYTORIUM GIT               | <a href="http://git.wmi.amu.edu.pl/s457990/Watch-with-friends">http://git.wmi.amu.edu.pl/s457990/Watch-with-friends</a>  |
| EXECUTIVE SUMMARY<br>(150-200 SŁÓW)  | <p>Watch With Friends to projekt, który ma za zadanie umożliwić „spotkanie” się znajomych na wspólnym oglądaniu filmów w dobie pandemii. Postawiłem sobie za zadanie umożliwić połączenie możliwie wielu przyjaciół jednocześnie w jednym, generowanym losowo pokoju do którego dostęp otrzymują osoby, które dostaną unikatowy link. W pokoju tym użytkownicy poproszeni zostaną o ustawienie swojej nazwy a następnie mogą zarządzać puszczanym aktualnie filmem z YouTube, wkleić link do wybranego przez siebie filmu, dowolnie zatrzymywać i puszczać film. Jako dodatkowy element pokoju dodany zostanie chat, który będzie pozwalał na komunikację pomiędzy użytkownikami serwisu w czasie rzeczywistym.</p> <p>Główny język programowania, który wykorzystałem przy budowie projektu to JavaScript. Wykorzystałem go ze względu na mnogość funkcji które oferuje oraz wszechstronność. Oprócz tego, do stworzenia strony internetowej wykorzystam HTML5 oraz CSS, które są technologiami wykorzystywanymi przy tworzeniu stron internetowych. Jako backend wykorzystałem wieloplatformowe środowisko uruchomieniowe Node.js, które także korzysta z języka JavaScript. W połączeniu z biblioteką Socket.io, Node.js tworzy potężne środowisko backendowe pozwalające na komunikację między serwerem a użytkownikiem w czasie rzeczywistym.</p> |
| OPIS PIERWSZEGO MODUŁU (OBOWIĄZKOWY) |  |
| NAZWA MODUŁU                         | <b>Główny moduł projektu – frontend i backend</b>  |

|  |  |
|--|--|
| <p>TECHNICZNY OPIS MODUŁU</p> <p>1. (150-200 SŁÓW)</p> | <p>W głównym module projektu za zadanie postawiłem przed sobą stworzenie frontendu i backendu. Dzięki temu, że głównym językiem programowania aplikacji jest JavaScript nie muszę rozdzielać tworzenia strony wizualnej i strony serwerowej projektu na osobne moduły.</p> <p><b>Frontend:</b><br/>Frontend zostanie stworzony przy pomocy Javascript, CSS oraz HTML. Do projektowania strony internetowej użyję również darmowe oprogramowanie AdobeXD, które służy do projektowania interfejsu strony internetowej. Język JavaScript wykorzystam również do połączenia danych pomiędzy użytkownikami i serwerem w celu wyświetlenia ich na stronie internetowej aplikacji. Jako odtwarzacz wideo wykorzystam gotowe rozwiązanie w postaci API odtwarzacza YouTube udostępnionego przez Google. Odtwarzacz ten dostosuję według swoich potrzeb (możliwość przesyłania danych w czasie rzeczywistym do wielu użytkowników za pośrednictwem serwera).</p> <p><b>Backend:</b><br/>Jako backend posłuży mi uruchomieniowe środowisko Node.js w połączeniu z biblioteką Socket.io. Te dwa narzędzia pozwolą na stworzenie aplikacji internetowej która będzie przysyłała dane pomiędzy serwerem a użytkownikami w czasie rzeczywistym. Głównymi zadaniami serwera w tym projekcie będzie najpierw generowanie losowego URL który następnie prowadzić będzie do dynamicznie generowanych pokoi z odtwarzaczem wideo i chatem. Wszystko zaimplementowane zostanie z użyciem frameworka Express.js.</p> |
| <p>WYKORZYSTYWANE TECHNOLOGIE</p>                      | <p>HTML5, CSS, JavaScript, Node.js, Socket.io, Express.js</p>  |
| <p>HARMONOGRAM PRAC</p>                                | <ol style="list-style-type: none"> <li>1. <b>Zaprojektowanie interfejsu strony internetowej(frontend)</b> – Interfejs strony zostanie zaprojektowany za pośrednictwem oprogramowania Adobe XD i wyeksportowany jako biblioteka CSS i pliki HTML. <b>Data: 3.01.2021</b></li> <li>2. <b>Stworzenie działającej strony powitalnej (frontend)</b> – Do wyeksportowanych plików HTML zostanie dodana faktyczna funkcjonalność poprzez dodanie skryptów JavaScript. <b>Data: 7.01.2021</b></li> <li>3. <b>Dodanie możliwości generowania losowych pokoi (backend + frontend)</b> – Stworzony zostanie prosty serwer Node.js z użyciem frameworku Express.js. Dodana zostanie funkcjonalność generowania losowych pokoi za pośrednictwem kliknięcia przycisku na stronie głównej. <b>Data: 14.01.2021</b></li> <li>4. <b>Stworzenie działającego pokoju (frontend)</b> – Do wyeksportowanego pliku HTML zostanie dodana funkcjonalność w postaci skryptu JavaScript. <b>Data: 16.01.2021</b></li> </ol>  |

|               |   |
|---------------|---|
|               | <p>5. <b>Dodanie możliwości połączenia użytkowników między sobą w czasie rzeczywistym (backend)</b> – Serwer zostanie wzbogacony o wtyczkę Socket.io i synchronizację jej z działającym już pokojem umożliwiając łączenie się wielu użytkowników z jednym pokojem i podgląd połączenia. <b>Data: 23.01.2021</b></p> <p>6. <b>Manipulacja API Youtube w celu przystosowania go do działania strony (backend + frontend)</b> – API udostępniane przez Youtube zostanie dostosowane do potrzeb działania serwisu (kliknięcie np. przycisku zatrzymania oglądania zmienia status filmu globalnie w całym pokoju każdemu użytkownikowi, ale w obrębie tylko jednego pokoju). <b>Data: 30.01.2021</b></p> <p>7. <b>Stworzenie chatu działającego w czasie rzeczywistym (backend + frontend)</b> – Jako ostatnie zadanie będzie synchronizacja połączenia między użytkownikami na czacie w czasie rzeczywistym za pośrednictwem biblioteki Socket.io. <b>Data: 4.02.2021</b></p> |
| <b>STOPKA</b> |   |
| DATA          | 22 Listopad 2020 r.   |
| PODPIS        | KACPER MAJ  |