**Notatki**

\*(źródło: <https://www.websitehostingrating.com/internet-statistics-facts>)

**Bibliografia**

1. <https://sekurak.pl/czym-jest-xss/>

2. <https://geekflare.com/monitor-website-changes/>

3. <https://auth0.com/blog/adding-salt-to-hashing-a-better-way-to-store-passwords/>

4. o curl i wget magazyn programisty

**2. ANALIZA PROBLEMU**

OBIEKTY:

1. Role (użytkownik zalogowany, tworca monitora. A także jego kolega)

2. co mogą monitorować i jak to będzie przechowywane

FUNKCJE:

2. Funkcje wraz z ich dokładnym opisem

3. innowacyjność jak we wstępie

MECHANIZMY:

- co bedzie problemem i mechanizmy jego obejcia:

(renderowanie a nie sam curl)

(crossorigin policy, x-frame)

(porownania plikow, co porownywac, hashe)

(jak przedstawić zmiane, co przechowywać jakie informacje)

UWARUNOWANIA PRAWNE:

RODO

**Pytania promotor:**

1. bibliografia jakie artykuly moge zbierac? Wszystkie , programista miesięcznik?

2. czy robic znowu porównanie? nie

3. aktualnie strone przechowuje strone w folderze, w bazie będę przechowywal tylko zmiane treści, albo cale htmle, i ewentualnie jakiś tekst ze zminl się dany plik, ale w folderach będą trzymane tylko najnowesze wersje? ok

3. co powinno byc w analizie problemu? Narazie planuje*: to co powyżej*

4. w jaki sposob omowic skalowalnosc? bo na seminarce krzywiecki wspominal ze warto

**SEMINARIUM 1 8.10.2020**

Wstęp:

Zakres

- Jakie typu zagadnienie

- Jakie mechanizmy jakie algorytmy

Cel

- Założenia funkcjonalne

- Założenia poza funkcjonalne

Zawartość pracy

Slajdy:

1. jaki jest problem?

2.

3. co można z tym problemem zrobić? Jak rozwiązują go już istniejące rozwiązania?

4. system, jaka architektura

5. system realizuje postawione przed nim cele, pokazac jak ladnie itp. Wyniki, zdjęcia

- duzy rysnkow

**SEMINARIUM 2 15.10.2020**

Pytania z seminarki:

1. w jaki sposób przedstawiać różnice na stronach w aplikacji ? Tak żeby to było ergonomicznie i czytelnie

2. Gęstość odpytań, np. w danym okresie odpytywać częściej

3. Skalowanie, co jeśli będzie dużo ludzi korzystać z tego? Jak ja skalować?

Problemy:

- co bedzie problemem i mechanizmy jego obejcia:

(crossorigin policy, x-frame)

(porownania plikow, co porownywac, hashe) → hashe przy plikach nie tekstowych tylko, przy innych hash + biblioteka

(jak przedstawić zmiane, co przechowywać jakie informacje) → jak na githubie, nie caly plik

(renderowanie a nie sam curl) → biblitoeka mockujaca przegladarke

(porównania tych samych stron ? w bazie trzymać tak żeby 2 uzytkownikow jeden proces)

(skalowalność nie wiem o co cho tu)

**SEMINARIUM 3 22.10.2020**

1. Bez sensu propozycja feature + sprawdzić czy legalne jest kopiowanie stron

**SEMINARIUM 4 29.10.2020**

PREZENTACJA SLAJDY:

**1.**

Tematem mojej pracy jest aplikacja webowa monitorujaca zmiany na stronach internetowych. Celem pracy jest …. Aplikacja swoim zakresem obejmuje….

Aplikacja znajdzie zastosowanie np.

* konkurencji z innymi w wyścigu o uzyskanie najlepszych biletów na wydarzenie kulturalne, które właśnie zostały wystawione do sprzedaży
* uzyskania produktu, który właśnie został przeceniony
* niezwłocznego uzyskania informacji o wyniku egzaminu albo rozstrzygnięciu przetargu
* nowego wpisu na ulubionym blogu, czy nowym artykule na ulubionej stronie popularnonaukowej
* DNS-spoofing, CROSSITE SCRIPTING

**2.**

**Funkcjonalnosci:**

* **rejestracja wraz z utworzeniem konta,** które będzie umożliwiać dostęp do wszystkich funkcji aplikacji
* **cykliczne i w pełni automatyczne monitorowanie** zmian zachodzących na podanej przez użytkownika stronie internetowej, w określonym przedziale oraz interwale czasowym
* **monitorowanie całej strony lub jedynie wybranych jej fragmentów**
* wraz z utworzonym monitorowaniem powstanie tzw. ‘**pokoju obserwacji’**, z którym klient nawiąże komunikację za pomocą protokołu *Websocket*, co pozwoli mu na obserwowanie wyników przeprowadzonych skanów strony w czasie rzeczywistym
* możliwość **udostępnienia „pokoju obserwacj**i” innym, nawet niezarejestrowanym użytkownikom za pomocą długiego, trudnego do zgadnięcia linku. Użytkownicy nie będący właścicielami pokoju będą mogli przeglądać wyniki przeprowadzonego monitorowania strony, ale nie będą mogli zmieniać żadnych ustawień dotyczących monitoringu
* **natychmiastowe powiadomienie użytkownika** o zmianie na stronie drogą mailową lub notyfikacją w przeglądarce internetowej, zarówno mobilnej, jak i używanej na komputerach stacjonarnych
* wyeksportowanie raportu o zmianach w postaci pliku **pdf**
* automatyczne **generowanie wykresów zmian wybranych wartości liczbowych na stronie**
* udostępnienie **API** dla aplikacji zewnętrznych

**3.**

Jedną z kluczowych funkcjonalności aplikacji, jest wybranie konkretnego elementu, do obserwacji na podanej przez klienta stronie. Wymagającym zagadnieniem jest sposób w jaki należy wyświetlić w aplikacji stronę podaną przez klienta np. *www.pwr.pl,* tak jak widzi on ją w swojej przeglądarce i umożliwić mu wygodne wybrane elementu ‘do obserwacji’ (np. poprzez kliknięcie lub zaznaczenie go) , mając równocześnie na uwadze obostrzenie ze strony SOP.

Podstawowym mechanizmem obronnym nowoczesnych przeglądarek jest Same-Origin Policy.

[Same-Origin Policy (SOP)](https://en.wikipedia.org/wiki/Same-origin_policy):

* Protokół (inaczej – Schemat)
* Host (sprawdzany rygorystycznie – to znaczy subdomena nie jest tożsama z domeną!)
* Port

dwie aplikacje charakteryzujące się różnymi Originami nie mogą używać (ściągać, osadzać, odpytywać) swoich wzajemnych elementów.

Oznacza to na przykład:

- Nie można zapisywać i odczytywać ciasteczek Originu A, będąc na stronie innego Originu B

**- Nie można wywoływać zapytań HTTP z Originu A do Originu B (na przykład element** **<form> który jest wysyłany pod inny adres)**

wybranie konkretnego elementu:

Idealnym do tego elementu html iframe.   
Wyswitlenie w iframie, bezposrednio na stronie, X-frame-option false na to nie pozwoli

Poza tym SOP zablokuje skrypt próbujący modyfikować elementy iframe, jeżeli strona posiadajaca iframe i strona na nim wyswietlana maja inny origin.

SOP zablookuje tez dynamiczne pobieranie elementow strony z innych serwerow bo będzie zrodlo jako moja domena ustawione.

Rozwiazanie:

Sciagnie na serwer, już wyrendowowanej stron tak jakbyśmy wchodzili przez przegladrke.

I wstrzykniecie skryput, te na backendzie bo inaczej to by było blokowane przez sop

**4.**

- w jaki sposob porownac pliki ? Tak zeby było to ergonomiczne i czytelne ?

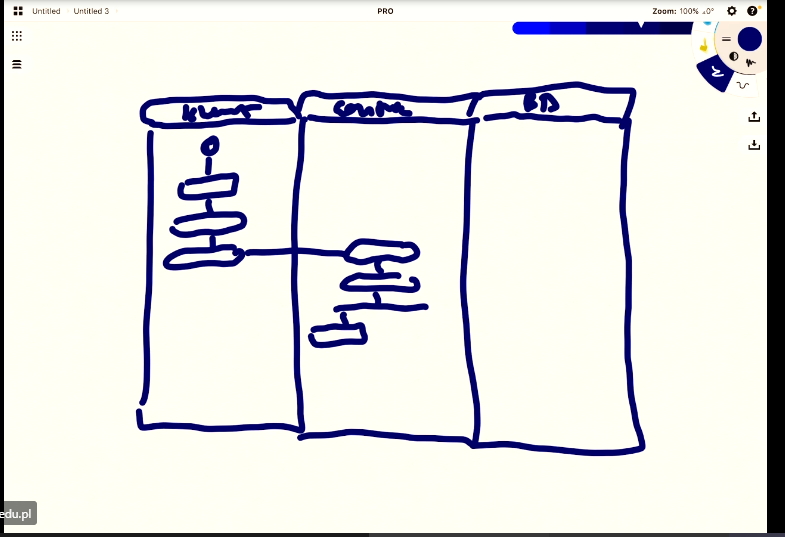
difflib

Nie chodzi tylko o to żeby powiedzieć ze 2 pliki sa rozne albo ze roznia się ilomas słowami,

Zależy nam żeby odnosci się do każdej linijki poprzedniej wersji i informować czy została usunieta lub zmieniona,

te „śmieciowe” elementy to takie, które są w pewnym sensie nieinteresujące, takie jak puste linie lub spacje Ta sama idea jest następnie stosowana rekurencyjnie do fragmentów sekwencji po lewej i po prawej stronie pasującego podciągu. Nie daje to minimalnej liczby sekwencji edycji, ale zwykle daje dopasowania, które „wyglądają dobrze” dla ludzi.

Na następnych diagram aktywności: cos ala:

kolumny to :

server api

frontend

baza danych

**Uwagi do prezentacji:**

Slajd 1-2 : powinny być 2/3 osobne : cel, zakres, funkcjonalność

Może polaczyć cel i zakres ? Zakres to jaki obejmuje rejon informatyki parsowanie / kontrolowanie /porównywanie …. Cel to projekt, implementacja, wdrożenie

Funkcjnoalnosci polaczyc w mniej, i bardziej opisywac co one oznaczają….

Do tego większe lliterki

I nie powtarzac zakresu

Slajd 4 zczaczac od tego po co w mojej stronie to jest i jak sop ma się do tego

Slajd 5 opis algorytmu jakiś obrazek albo schemat blokowy

Slajd 6 wieksze napisy, unikac ang nazw, wystlumaczyc po polsku co robi load balancer itp

Slajd 7 wieksze napisy