***KATEDRA****: Wstęp pracy (nie numerowany) powinien składać się z czterech części (które nie są wydzielane jako osobne podrozdziały): zakresu pracy, celu, analizy i porównania istniejących rozwiązań oraz przeglądu literatury, oraz opisu zawartości pracy.*

***SEMINARIUM:***

Wstęp:

Zakres

- Jakie typu zagadnienie

- Jakie mechanizmy jakie algorytmy

Cel

- Założenia funkcjonalne

- Założenia poza funkcjonalne

Zawartość pracy

**Wstęp**

W obecnych czasach strony internetowe są niezwykle dynamiczne, a treści wyświetlane na nich zmieniają się bardzo często, nieustanna zmiana jest symbol dzisiejszego Internetu. W sieci jest ponad 1,7 mld\* takich stron, niemal każda firma, a coraz częściej także niemal każda osoba ma już swoją własną stronę internetową. Śledzenie zmian na nich zachodzących lub pojawiających się na nich konkretnych treści, wymaga częstego odświeżenia strony, wraz z przeładowaniem plików zapamiętanych przez przeglądarkę w pamięci podręcznej. Jest to zadanie żmudne, do którego potrzeba spostrzegawczości i dobrej pamięci, aby zauważyć zmiany treści na stronie.

Biorąc pod uwagę dynamikę stron internetowych oraz ich ogromną liczbę, można założyć, że zwykły użytkownik Internetu spotyka się z potrzebą uzyskania natychmiastowego komunikatu o zmianie na stronie dość często. Może to nastąpić w ramach konkurencji z innymi klientami np. w wyścigu o uzyskanie najlepszych biletów na wydarzenia kulturowe, które właśnie zostały wypuszczone do sprzedaży lub uzyskanie produktu, który właśnie został przeceniony. Studenci, często niecierpliwie czekający na wyniki egzaminów, również chcą uzyskać o nich informację, gdy tylko te się pojawią. Czasami, dla użytkownika Internetu nie ma znaczenia szybkość informacji o pojawieniu się nowej treści, lecz tylko sam fakt jej wystąpienia, na przykład nowy wpis na ulubionym blogu, czy nowy artykułu na ulubionej stronie popularnonaukowej.

Aplikacja webowa *WebsiteMonitor*, będąca tematem tej pracy, jest rozwiązaniem, które zapewni nieustanne i w pełni automatyczne monitorowanie zmian pojawiających się na stronach internetowych, całych lub na wybranych ich fragmentach. Aplikacja będzie informować o nich w wybranym przez użytkownika czasie, np. raz w tygodniu, raz dziennie lub gdy tylko nowe treści się pojawią. Informacja będzie przekazywana za pomocą różnych środków komunikacji: mail, powiadomienia w przeglądarce lub za pomocą protokołu WebSocket, a historia zmian będzie prezentowana w łatwy do przejrzenia sposób, wraz z możliwością wygenerowania pliku pdf.

Zakres aplikacji obejmuje przede wszystkim cykliczne pobieranie *pliku html* i *css* strony internetowej, jednak nie tak jak robi to polecenie *curl* lub *wget* w systemie operacyjnym *Linux*, gdyż obecnie dużo stron ładuje swoje treści dynamicznie, za pomocą skryptów napisanych w *javascriptcie*.   
Serwer aplikacji będzie więc imitował działająca przeglądarkę, powodując rozpoczęcie działania zawartych na stronie skryptów, a następnie pobierze już gotowy plik html z użytymi w nim innymi plikami jak zdjęcia, czy pliki css. Potem wyświetli użytkownikowi pobraną stronę oraz doda do niego odpowiedni skrypt umożliwiający wybranie konkretnych jej elementów do obserwacji. W wybranym przez użytkownika czasie i za pomocą preferowanych przez niego środków poinformuje go o zmianie, a także umożliwi mu łatwe przedstawienie historii zmian na stronie.

Zadania, które aplikacja może pomóc realizować nie ograniczają się jedynie do zadań prywatnych, znajdzie ona również zastosowanie w zadaniach biznesowych. Przykładowo zmiany na stronie konkurencji, która właśnie wypuściła nowy produkt lub nowe zmiany prawne na stronie rządowej, mające wpływ na działanie biznesu. Przyszli pracownicy, mogą w ten sposób szukać nowych ofert pracy bezpośrednio na stronie firmy, w której chcą się zatrudnić. Właściciel przedsiębiorstwa może śledzić fora internetowe, w oczekiwaniu aż nazwa jego firmy zostanie wspomniana i dzięki temu natychmiast na taki wpis odpowiedzieć. Kontrolowanie wielu stron staje się również coraz bardziej potrzebne, dzięki rozporoszonym modelom biznesowym. Przykładowo firmy opierające swój marketing na marketingu afiliacyjnym, czyli takim, który jest tworzony przez sieć niezależnych sprzedawców reklamujących produkt, tak jak ma to miejsce w firmach ubezpieczeniowych, muszą sprawdzać czy na stronach agentów oferta przez nich przedstawiona nie zmienia się i ciągle spełnia założenia firmy zatrudniającej go.

Również właściciele strony internetowych mogą odnieść wiele korzyści, mając pewność, że ich strona jest nieustannie monitorowana oraz, że w przypadku wystąpienia na niej jakiekolwiek zmiany zostaną natychmiast o tym fakcie poinformowani. Może pomóc to, we wczesnym wykryciu ataku na stronę, przede wszystkim w atakach biorących na cel nie serwer, lecz klienta, pobierającego kod strony internetowej. Po takim ataku dostaje on stronę z niewidocznymi pod względem wyglądu strony zmianami, zawartymi często tylko w kodzie strony. Na przykład przy ataku XSS (*cross-site scripting*), polegającym na wstrzyknięciu złośliwego kodu JavaScript do plików pobieranych przez odwiedzającego stronę lub przy ataku DNS-spoofing, w którym zaatakowany serwer DNS będzie przekierowywał klientów na stronę atakującego, pomimo wprowadzenia w przeglądarce poprawnej nazwy domeny. Każdy tego typu ataku, będzie łatwiejszy do wykrycia, jeśli właściciel strony dostanie natychmiast powiadomienie o zmianie na stronie, której sam nie wykonał. Będzie to dla niego wtedy jasny sygnał, że na stronie dzieje się coś niepokojącego i należy się temu przyjrzeć.

Także w przypadkach stron, w których treści mogą być dodawane przez użytkowników (np. fora internetowe), administrator może takie treści na bieżąco moderować np. sprawdzać czy nie łamią one żadnego punktu regulaminu lub na nie niemal natychmiast odpowiadać, co w obecnych czasach może być dużą korzyścią dla popularności strony lub biznesu, która ona reprezentuje.

Celem pracy jest zaprojektowanie i implementacja aplikacji webowej umożliwiającej śledzenie zmian lub występowania słów kluczowych na stronach internetowych. Aplikacja ta powinna nieść ze sobą następujące funkcjonalności:

* Rejestracja wraz z utworzeniem konta, które będzie umożliwiać dostęp do niżej wymienionych akcji
* 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
* Monitorowanie łatwe do podzielenia się z innymi, nawet niezarejestrowanymi osobami, po przez udostępnienie im linku do tzw. ‘pokoju obserwacji’, w którym będą mogli przeglądać wyniki przeprowadzonego monitorowania strony, ale nie będą mogli zmieniać żądnych 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

W Internecie można znaleźć wiele firm oferujących aplikacje webowe do monitorowania zmian pojawiających się na stronach internetowych, zwykle oferują one ubogie w funkcje, ale bezpłatne wersje podstawowe oraz wersje zaawansowany, z wieloma dodatkowymi funkcjami, ale płatne. Żadnej firmie nie udało się jak dotąd zająć pozycji lidera na rynku, w różnych rankingach pojedyncze strony zajmują różne pozycje, te bardziej popularne to na pewno:

<https://changetower.com>  
<https://www.wachete.com/>  
https://visualping.io/

Wszystkie posiadają podstawowe dla tego typu stron funkcje monitorowania zmian na wybranej stronie, z możliwością wyboru tylko pewnego elementu lub wszystkich treści na stronie, oraz wysyłania powiadomień na maila.

*Visualping* dodatkowo oferuje serie akcji, które możemy wykonać na stronie, są to: wciśnięcie wybranego przycisku, wybranie opcji z wyświetlonej listy opcji, usunięcie elementu ze strony, odświeżenie strony, dodanie ciasteczek, wpisanie tekstu w pole, a także odczekanie podanej liczby sekund. Umożliwia także zintegrowanie ze *Slackiem* oraz przeprowadzanie skanów z różnych lokalizacji na świecie.

*Wachete* również oferuje przeprowadzenie na pobranej stronie różnych akcji, jak klikanie przycisków czy wpisywanie wartości w wybrane pola oraz pobieranie strony z różnych lokalizacji na świecie, poza *Slackiem* informacje o zmianie możemy otrzymywać za pomocą aplikacji *MS Teams, Discord, Zapier*, a nawet kanału *RSS*. Dla programistów udostępnione jest natomiast wygodne API. Zmiany tekstowe mogą być filtrowane za pomocą wyrażeń regularnych. Zmiany wartości numerycznych mogą zostać zaprezentowane, na automatycznie tworzonych wykresach.

*ChangeTower* podobnie jak pozostałe firmy oferuje serie akcji do wykonania na stronie internetowej, powiadomienia możemy zintegrować tylko z aplikacją *Zapier*, ale za to możemy wysyłać notyfikacje wiele osób z zespołu. Pozwala ustawić wiele warunków po których spełnieniu wysyłanie powiadomienie, może to być na przykład wystąpienie jednego z wielu słów kluczowych, sprawdzane w różnych miejscach strony. Dodatkowo oferowane jest, nie tylko zapisywanie zmian zachodzących na stronie, ale również całej strony i przechowywania jej poprzednich wersji aż do 12 lat.

Wszystkie trzy powyższe rozwiązania oferują monitorowanie zmian na stronie internetowej lub jej części, wraz z łatwą integrację z różnymi popularnymi w obecnych czasach internetowymi komunikatorami, co stanowi ich niewątpliwą zaletę. Dodatkowo oferowana przez nie możliwość wykonywania zautomatyzowanych akcji, typu: ‘w pole wpisz wartość’ lub ‘wciśnij przycisk’, daję możliwość skanowania stron, dostępnych dopiero po zalogowaniu. Pod znakiem zapytania pozostaje jednak bezpieczeństwo takiego rozwiązania oraz to, czy użytkownicy będą mieli wystarczające zaufanie, aby podawać swój loginy i hasła do serwisów, podmiotom trzecim.

Niewątpliwą wadą powyższych rozwiązań jest brak jakiekolwiek możliwości łatwego podzielenia się utworzonym żądaniem monitorowania strony z innymi niezarejestrowanymi osobami, tak, aby mogły one przejrzeć historie zmian na śledzonej stronie. Nie ma też możliwości wygenerowania takiej historii w postaci pliku pdf, a automatyczne stworzenia wykresów zmiany wskazanych danych liczbowych na stronie jest dostępne tylko w aplikacji *Wachate*. Przy ustawieniach monitoringu strony w aplikacjach *Visualping*  oraz *Wachete* brakuje parametru zakończenia przeprowadzanych skanów strony, które nie zmuszał by użytkownika do ręcznego usunięcia ustawionego monitoringu. Z całej trójki tylko *Wachete* udostępnia API dla programistów, którzy chcieliby sami obrobić dane za pomocą własnych skryptów. Dodając do tego fakt, iż wszystkie powyższe usługi w wersjach rozszerzonych są płatne, można dojść do wniosku, iż na rynku wciąż nie ma narzędzia zaspokającego potrzeby wszystkich klientów. Aplikacja webowa, która jest tematem tej pracy, wszystkie wymienione wyżej braki wypełni i będzie sprawdzać się wszędzie tam, gdzie łatwość udostępniania wyników monitoringów, ich łatwa prezentacja oraz wygodne API będą kluczowe.

**opis zawartości pracy**

Praca składa się z czterech rozdziałów ….

\*(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**

*W tym rozdziale należy przedstawić analizę zagadnienia, które podlega informatyzacji. Należy zidentyfikować i opisać obiekty składowe rozważanego wycinka rzeczywistości i ich wzajemne relacje (np. użytkowników*

*systemu i ich role). Należy szczegółowo omówić procesy jakie zachodzą w systemie i które będą informatyzowane, takie jak np. przepływ dokumentów. Należy sprecyzować i wypunktować założenia funkcjonalne i*

*poza funkcjonalne dla projektowanego systemu. Jeśli istnieją aplikacje realizujące dowolny podzbiór zadanych funkcjonalności realizowanego systemu należy przeprowadzić ich analizę porównawczą, wskazując na*

*różnice bądź innowacyjne elementy, które projektowany w pracy system informatyczny będzie zawierał. Należy odnieść się do uwarunkowań prawnych związanych z procesami przetwarzania danych w projektowanym*

*systemie. Jeśli zachodzi konieczność, należy wprowadzić i omówić model matematyczny elementów systemu*

*na odpowiednim poziomie abstrakcji.*

*W niniejszym rozdziale omówiono koncepcję architektury programowej systemu . . . . W szczególny sposób . . . . Omówiono założenia funkcjonalne i niefunkcjonalne podsystemów . . . . Przedstawiono mechanizmy*

*. . . . Sklasyfikowano systemy ze względu na . . . . Omówiono istniejące rozwiązania informatyczne o podobnej*

*funkcjonalności . . . (zobacz [2]).*

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

2. Funkcje wraz z ich dokładnym opisem

3. innowacyjność jak we wstepie