

tytul

()

Agnieszka Pawicka

Praca licencjacka

Promotor: dr Jan Otop

Uniwersytet Wrocławski
Wydział Matematyki i Informatyki
Instytut Informatyki

Wrocław 2021

Streszczenie

streszczenie

Spis treści

1. Wprowadzenie	7
1.1. Motywacja	7
1.2. Gotowe rozwiązania	8
1.3. Cel pracy	8
2. Środowisko	9
2.1. Google Apps script	9
2.2. Node.js	10
2.3. Python	11
2.4. Bootstrap	11
3. Techniczny opis programu	13
3.1. to co w google	13
3.2. html, bootstrap requesty do mojego serwera	13
3.3. używane biblioteczki po stronie serwera mojego i cały jego opis	13
3.4. zdjęcia	13
3.4.1. Alternatywne rozwiązanie	13
3.5. instalator i uruchamianie	13
4. Instrukcja użytkownika	15
4.1. uruchomienie i instalacja	15
4.2. Schemat pliku kodującego (JSON)	15
4.3. Obsługa narzędzia	15
5. Podsumowanie	19

5.1. słowo końcowe	19
5.2. rozwój	19

Rozdział 1.

Wprowadzenie

1.1. Motywacja

Przeprowadzanie testów i ankiet jest wpisane w życie akademickie oraz szkolne. Jedną ze szczególnych metod przeprowadzania testu lub ankiety jest sprawdzanie czy zdobywanie wiedzy zdalne. Aby to jednak było wartościowe, należy spełnić warunki takie jak:

- czytelność formy
- możliwość weryfikowania, kto wysłał odpowiedź
- kontrolowanie czasu wysyłanych odpowiedzi
- zapamiętywanie przesłanych odpowiedzi

Oprócz powyższych kryteriów możliwym byłoby wymienienie jeszcze wielu kwestii, których implementacja znacznie ułatwiłaby pracę ze zdalnymi testami - zwłaszcza dla strony przeprowadzającej test, jak np.:

- automatyczne generowanie testu z popularnego formatu,
- możliwość wstawiania symboli matematycznych,
- automatyczne ocenianie (dla testów) - być może w nietypowej skali,
- generowanie losowych testów z pytań z pewnej puli,
- udostępnianie wyników w wygodnym formacie

Zupełnie nowe narzędzie - możliwie blisko ideału - byłoby więc ambitnym przedsięwzięciem dla zespołu programistów. Tu z pomocą przychodzi możliwość rozszerzania istniejących rozwiązań.

1.2. Gotowe rozwiązania

Do najpopularniejszych gotowych rozwiązań należą formularze Googla, oraz Microsoftu - udostępniają one już bardzo dużo wymienionych w poprzednim rozdziale możliwości - są czytelne, zapewniają weryfikowanie tożsamości osób przysyłających odpowiedzi (sprawdzają zalogowanego kontem mailowym użytkownika, w Google Forms dzieje się to przez OAuth).

Wymienione narzędzia wciąż jednak nie są idealne. Głównym problemem w przeprowadzaniu większych testów/ankiet jest konieczność ręcznego „przeklikiwania się” przez interfejs do tworzenia formularzy, a w dziedzinach ścisłych również brak wbudowanej interpretacji symboli matematycznych - chociaż własności, które byłyby pomocne jest znacznie więcej. Udostępnione możliwości oceniania automatycznego nie pozwalają na ustawienie nietypowej (jak np. wykładniczej) skali oceniania. Google Forms w czystej formie nie pozwalają też na automatyczne ustawienie ram czasowych akceptacji odpowiedzi - czasu początku i końca „testu”. Format odpowiedzi również nie jest najprostszy do przetwarzania.

Jest jednak ważna zaleta gotowych rozwiązań - można do nich dobudowywać dalsze usprawnienia. Niniejsza praca polega na usprawnieniu gotowego rozwiązania jakim są Google Forms o parę nowych możliwości.

1.3. Cel pracy

Poprzez wykorzystanie Google Apps Script i rozwiązań serwerowych bibliotek JavaScriptu powstało niewielkie usprawnienie działania znanego już rozwiązania - formularzy google. W prezentowanej pracy zostało zaimplementowane automatyczne generowanie formularzy z plików w formacie JSON, renderowanie wstawek napisanych w latex’u oraz możliwość prostego włączania oraz wyłączania opcji, mówiącej czy formularz przyjmuje w danym momencie zgłoszenia. Niektóre opcje formularzy można przekazać już na podstawie kodowania JSONowego - w konstruowanym pliku.

Rozdział 2.

Środowisko

2.1. Google Apps script

Nabudowywanie na gotowym narzędziu wymaga dostępu do niego. Google udostępnia API operujące na całym szeregu klas i metod oferowanych usług. Kod przypisany jest do danego konta google, możliwe jest ustawienie parametrów, takich jak: kto ma dostęp do nowotworzonej aplikacji internetowej (właściciel, zalogowany użytkownik danej organizacji, dowolny zalogowany użytkownik, każdy) pod którym kontem jest ona uruchamiana (właściciela/współwłaściciela, czy też zalogowanego użytkownika).

Framework Udostępnione API w pewnym stopniu wymusza na użytkownikach, aby kod wykonywany na infrastrukturze Google'a był napisany w javascriptcie, we framework'u „Google Apps Script”. Program jest wykonywany po stronie serwera. Pozwala on na stosunkowo łatwe manipulowanie działaniem produktów Google takich jak formularze, arkusze, dysk i inne. Framework powstał w 2009 roku, w JavaScript 1.6, jest jednak regularnie ulepszany.

Komunikacja Apps Script udostępnia komunikację przez HTTP - jeśli projekt zawiera funkcje doGet(e) / doPost(e) - odpowiednie rządania wykonują kod z ciała tych metod. Zwracane wartości prowadzą do przekierowań zapytań -

API Google Apps Script udostępnia szereg klas i metod związanych z poszczególnymi narzędziami. Szczegółowa dokumentacja narzędzia znajduje się tutaj: Forms Service. Główna klasa - FormApp - jest odpowiedzialna za zarządzanie formularzami - m.in. tworzenie nowych. Każdy typ pytania i element formularza ma odpowiednią klasę (jak np. CheckboxItem czy SectionHeaderItem). Poprzez klasę From można zmieniać główne ustawienia formularzy - jak na przykład dodawanie właścicieli, two-

rzenie pytań, automatyczne ocenianie, manipulowanie tytułem, ustawienie, czy formularz jest „aktywny” (czy przyjmuje odpowiedzi).

2.2. Node.js

Praca wykorzystuje kilka bibliotek, przede wszystkim korzysta jednak z możliwości serwerowych Node.js.

http Interfejs silnie związany z „sercem” Node.js - udostępnia narzędzia do komunikacji poprzez HTTP zarówno ze strony serwerowej jak i klienckiej.

Dokumentacja: [http](http://nodejs.org/api/http.html)

express „Szybki (...), minimalistyczny framework webowy dla Node.js” - narzędzie pozwala w przystępny sposób postawić serwer (korzysta z biblioteki http). Udostępnia cztery klasy:

- **application** - odpowiada aplikacji serwerowej
- **request** - zarządza odwołaniami do serwera (głównie parametrami)
- **response** - odpowiada za odpowiedzi serwera
- **router** - zajmuje się routingiem, może być używane jako oprogramowanie pośredniczące

Dokumentacja znajduje się tutaj: [expres.js](http://expressjs.com/)

cors Node.js’owy moduł pozwalający na odpowiednie ustawienia CORS (Cross-Origin Request Sharing) w rozwiązaniach typu express.

Github modułu: [cors](https://github.com/expressjs/cors)

child_process Moduł Node.js’owy pozwalający na uruchamianie podprocesów.

Dokumentacja znajduje się tutaj: [child_process](http://nodejs.org/api/child_process.html)

jsonschema Nowy (zaledwie dziesięćmięczny, wciąż w wersji Beta) moduł pozwalający na walidację formatu json zgodnie z podanym schematem.

Oficjalna strona: [jsonschema](http://jsonschema.com/)

2.3. Python

Rozwiązania pythonowe zostały wykorzystane w celu konwersji wstawek matematycznych (latex) do zdjęć. Poniżej krótki opis wykorzystanych bibliotek.

tex2pix Biblioteka pozwalająca na konwertowanie formatu .tex do różnych formatów zdjęciowych. Metoda konwertująca format .tex na format .png zaczyna od konwersji .tex do .pdf, stąd w pracy używana jest konwersja do pdf z tej biblioteki, a dalsze manipulowanie formatem używa innych - subiektywnie prostszych w użytkowaniu - bibliotek.

Oficjalna strona: [tex2pix](#).

pdf2image Biblioteka umożliwiająca konwersję formatu pdf do formatów zdjęciowych.

Oficjalna strona: [pdf2image](#).

opencv Biblioteka pozwalająca na manipulację obrazami. Udostępnia znacznie więcej możliwości niż te użyte w pracy. W szczególności pozwala na przycinanie obrazów względem ich kolorystyki - co pozwala na automatyczne przycięcie strony pdf do rozmiarów napisanego na niej tekstu. Więcej informacji na temat biblioteki: [opencv](#)

base64 Biblioteka pozwala na konwersję obrazu do formatu Base64 - używanego w pracy do przesyłu obrazów pomiędzy serwerami node'owym a google'owym.

2.4. Bootstrap

Popularna biblioteka CSS, ułatwiająca budowanie interfejsów graficznych stron internetowych pisanych w HTML. Oficjalna strona [bootstrap](#).

Rozdział 3.

Techniczny opis programu

3.1. to co w google

3.2. html, bootstrap requesty do mojego serwera

3.3. używane biblioteczki po stronie serwera mojego i
cały jego opis

3.4. zdjęcia

3.4.1. Alternatywne rozwiązanie

3.5. instalator i uruchamianie

Rozdział 4.

Instrukcja użytkownika

4.1. Początki pracy z narzędziem

Narzędzie pozwala na wygenerowanie formularza Google Forms zakodowanego w formacie JSON, automatyczne generowanie zdjęć na podstawie wstawek w latex'u oraz zarządzanie informacjami o tym, czy dany formularz przyjmuje przesyłane odpowiedzi. Na początek należy jednak pobrać i zainstalować zależności wykorzystywane w pracy.

4.1.1. Instalacja

Na początku należy sklonować repozytorium projektu. Następnie wejść w folder **source** oraz uruchomić plik o nazwie „install.sh”. Pozwoli to na zainstalowanie potrzebnych bibliotek.

4.1.2. Uruchomienie

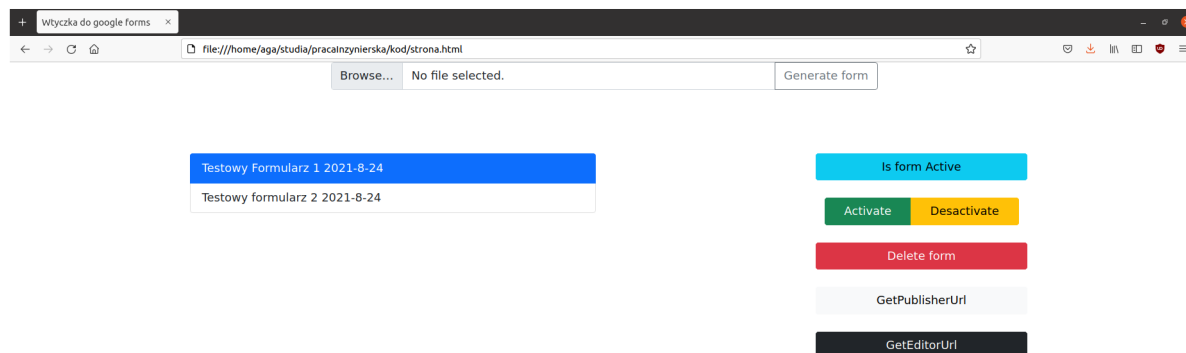
Aby uruchomić narzędzie należy uruchomić plik o nazwie „run.sh”. Uruchomi on lokalny serwer Node.js'owy oraz stronę internetową z interfejsem użytkownika.

4.2. Schemat pliku kodującego (JSON)

4.3. Obsługa narzędzia

Po uruchomieniu użytkownik widzi stronę w przeglądarce, jak na zdjęciu poniżej:

Widoczne u góry pole do wgrywania plików przyjmuje formaty .txt oraz .json. W pliku powinien znajdować się zakodowany formularz (podrozdział 4.2. Schemat pliku



Rysunek 4.1: Interfejs aplikacji

kodującego (JSON)).

Poniżej po lewej stronie znajduje się lista utworzonych już formularzy, po prawej znajdują się przyciski służące do operowania na już istniejących formularzach.

Przycisk „Generate form” wgrywa podany plik wykonuje na nim kolejne operacje. Poprawne wykonanie powinno przechodzić przez kolejne etapy:

- Sprawdzany jest format - pliku. Jeśli zawartość jest obiektem typu JSON, dane przekazywane są do lokalnego serwera, w przeciwnym przypadku strona wyświetli alert informujący o niepoprawnym formacie.
- Serwer lokalny sprawdza zgodność pliku ze schematem (JSON schema). Po tym etapie poniżej pola do wgrywania plików powinna pojawić się jedna z poniższych informacji:
 - Validation succeded
 - Wrong JSON format
- Jeśli JSON jest zgodny ze schematem, następuje konwertowanie pytań zakodowanych jako „tex” na format zdjęciowy.
- Po zakończonej konwersji z lokalnego serwera wysyłany jest POST request do serwera po stronie Google, gdzie trwa konwersja pliku na formularz. Zdalny serwer odsyła informację po zakończonej pracy do serwera lokalnego.
- Lokalny serwer dodaje nowy formularz do listy formularzy.

- Wyświetla się komunikat **New form has been created. Please reload page** z prośbą o odświeżenie strony.

Zachowania poszczególnych przycisków - za wyjątkiem „Generate Form” - dotyczą zawsze wybranego formularza z listy (podświetlonego w danym momencie na niebiesko). Aby wybrać formularz należy kliknąć na niego w liście formularzy.

Is Form Active przycisk zwraca wartość **Form is active** jeśli formularz przyjmuje odpowiedzi oraz **Form is inactive** w przeciwnym przypadku.

Activate Wysyła do serwera po stronie Google’a zapytanie, jeśli aktywacja formularza przebiegła pomyślnie, zwracana jest wiadomość **Form activated**.

Deactivate Wysyła do serwera po stronie Google’a zapytanie, jeśli dezaktywacja formularza przebiegła pomyślnie, zwracana jest wiadomość **Form deactivated**.

Delete Form Wysyła do serwera po stronie Google’a zapytanie o dezaktywację formularza, następnie usuwa z pliku z danymi o formularzach wpis dotyczący wybranego formularza oraz w komunikacie **Form deactivated, please reload page** prosi o odświeżenie strony

Get Publisher Url Wysyła do serwera po stronie Google’a zapytanie o adres url dla respondentów wybranego formularza. W komunikacie pojawia się odpowiedni link.

Get Editor Url Wysyła do serwera po stronie Google’a zapytanie o adres url dla edytorów wybranego formularza. W komunikacie pojawia się odpowiedni link.

Rozdział 5.

Podsumowanie

5.1. słowo końcowe

5.2. rozwój