Obraz zawierający tekst

Opis wygenerowany automatycznie

**Computer Science**

**Informatyka**

Data Science

**Szymon Wujec**

S20431

**“Data analysis in terms of resource usage based on “Bike Sharing” dataset”**

Master Thesis

Thesis supervisor

Dr. habil. Grzegorz Marcin Wójcik

Warsaw, ------------, 2024

Table of contents

[1 Introduction 4](#_Toc127208227)

[1.1 Thesis structure description 4](#_Toc127208228)

[1.2 Goals of the thesis 4](#_Toc127208229)

[1.3 Main thesis issues 4](#_Toc127208230)

[1.3.1 URL address 4](#_Toc127208231)

[1.3.2 Web aplication 4](#_Toc127208232)

[1.3.3 Bot 5](#_Toc127208233)

[1.3.4 Web Scraping 5](#_Toc127208234)

[1.3.5 Framework 5](#_Toc127208235)

[1.3.6 Refactoring 6](#_Toc127208236)

[1.3.7 Markup language 7](#_Toc127208237)

[2 Description of application. 11](#_Toc127208238)

[2.1 Description introduction 11](#_Toc127208239)

[2.2 Front-end description 11](#_Toc127208240)

[2.3 Back-end description 11](#_Toc127208241)

[2.4 Input and output of data file 12](#_Toc127208242)

[2.5 Tools 14](#_Toc127208243)

[2.5.1 IntelliJ IDEA 14](#_Toc127208244)

[2.5.2 Excel 16](#_Toc127208245)

[2.5.3 Python 16](#_Toc127208246)

[2.5.4 Selenium 18](#_Toc127208247)

[2.5.5 Pandas 19](#_Toc127208248)

[2.5.6 Openpyxl 19](#_Toc127208249)

[2.5.7 HTML 20](#_Toc127208250)

[2.5.8 Google Maps 21](#_Toc127208251)

[2.5.9 Materialize 23](#_Toc127208252)

[3 Functionality process assumptions 25](#_Toc127208253)

[4 Implementation. 30](#_Toc127208254)

[4.1 Front-end code 30](#_Toc127208255)

[4.1.1 Home page 30](#_Toc127208256)

[4.1.2 Summary page 34](#_Toc127208257)

[4.2 Back-end code 35](#_Toc127208258)

[4.2.1 Initialization classes 35](#_Toc127208259)

[4.2.2 Callable classes 41](#_Toc127208260)

[5 Summary 51](#_Toc127208261)

[5.1 Application outcome 51](#_Toc127208262)

[5.2 Acknowledgments 51](#_Toc127208263)

[6 Bibliography 52](#_Toc127208264)

[7 Attachment 53](#_Toc127208265)

[7.1 Table of Figures 53](#_Toc127208266)

[7.2 Table of Pictures 53](#_Toc127208267)

[7.3 Table of Code 54](#_Toc127208268)

[7.4 What CD contains 55](#_Toc127208269)

# Introduction

## Thesis structure description

Głównym celem tej pracy jest analiza przygotowanego zbioru danych przy użyciu różnych technik lub dostępnych narzędzi aby lepiej zrozumieć czym kierują się docelowi klienci firm wypożyczających rowery miejskie lub jakie warunki sprzyjają lepszemu rozwojowi biznesu z tej branży usługowej. Do napisania takiej pracy skłoniły mnie własne doświadczenia, sam w wolnym czasie bardzo często korzystam z takiego środka komunikacji jakim jest rower. Za każdym razem gdy znajduje się przed miejska stacja rowerów publicznych i zamierzam wypożyczyć rower aby dostać się z punktu „A” do punktu „B” ciekawi mnie w jaki sposób firma dostarczająca rowery do użytku publicznego, rozplanowuje kiedy i gdzie powinna dostarczyć zwiększoną ilość tych pojazdów, a gdzie może sobie pozwolić aby było ich mniej. Aby taki proces oraz biznes funkcjonował poprawnie oraz aby był wydajny, wymagana jest dogłębna analiza danych użytkowania, którą mam zamiar przeprowadzić w tej pracy magisterskiej.

## Goals of the thesis

Głównym celem analizy zbioru danych „Bike sharing dataset” dotyczącym wypożyczania rowerów jest zrozumienie wzorców wypożyczeń, optymalizacji oferty oraz identyfikacji trendów czasowych i zachowan użytkowników. Umożliwia to prognozowanie popytu, segmentacje użytkowników, ocenę wydajności oraz wpływu czynników zewnętrznych, a co najważniejsze opłacalności biznesu.

## Main thesis issues

### Bike - Sharing systems

Na swiecie istnieje wiele programów potocznie zwanych „miejskimi rowerami”. Ta potoczna nazwa ma wiele wspólnego ze stanem rzeczywistym ponieważ znaczaca czesc organizacji swiadczacych usługi z branzy wypożyczalni rowerowych to wlasnie zarządy miejskie opłacane z pieniędzy podatnikow. Oczywiście, na przestrzeni lat gdy ten rynek znacząco zaczal się rozwijać, przez zwiększony ruch uliczny, zanieczyszczenie powietrza, a nawet powiększanie się problemu jakim jest globalne ocieplenie, zaczely pojawiać się na rynku firmy prywatne oferujące takie usługi. Lokalizacja rowerow do wypożyczenia, a także występowanie oraz możliwość skorzystania z programów bike-sharing stosowane jest zwykle w dużych aglomeracjach miejskich. Z początku wystepowaly w scislym centrum takich miast, natomiast aktualnie możemy je zauwazyc w lokalizacjach, gdzie ruch publiczny jest znacznie bardziej nasilony. Do takich miejsc naleza miedzy innymi stacje metra, ponieważ metro to bez watpienia miejski srodek transportu który posiada najwieksza ilość przetransportowanych pasazerow, kolejnymi popularnymi lokalizacjami sa obiekty gdzie dojazd samochodem jest znacznie utrudniony. Wyobraźmy sobie, ze w pewnym miescie aktualnie odbywa się mecz pilki nożnej, drużyn posiadających ogromna liczbe fanow. Oczywistym jest, ze stadion wyprzedałby wszystkie dostępne bilety na długo przed rozpoczęciem meczy, przez zwiększone zainteresowanie tym wydarzeniem. Najwiekszy stadion na swiecie jest w stanie pomiescic 114 000 osob na widowni, gdyby każdy chciał przyjechać własnym samochodem to przestrzen przeznaczona na parking musiałaby zajmować 137 hektarow, czyli prawie siedem razy więcej niż zajmuje sam stadion, przy zalozeniu, ze srednia powierzchnia potrzebna do zaparkowania samochodu osobowego to 12 metrow kwadratowych. W takich sytuacjach oraz miejscach swietnie sprawdzają się stacje „rowerow miejskich”, generujące znacznie mniej miejsca, nie wspominając o emisji spalin.



**Picture 1 Bike-sharing station**

Systemy Bike – Sharing mozemy podzielić na pięć generacji tych programów.

#### Generation zero

Stacje, które posiadaja obsluge przy wypożyczeniu roweru naleza do generacji zerowej Bike - Sharing systems. Charakteryzuje je to, ze takie lokalizacji lub stacje nie sa zautomatyzowane lecz prowadzone przez pracowników lub wolontariuszy. Jest to bez watpienia najstarszy model wypożyczalni dostępny na swiecie, aktualnie najbardziej rozpoznawane wypożyczalnie korzystające z tego modelu to siec sklepow z wyposażeniem o charakterystyce sportowej „Decathlon”



**Picture 2 Bike-Sharing generatoin 0**

#### First generation

Pierwszy przypadek bezobsługowej wypożyczalni miał swoje miejsce w Amsterdamie w 1965 roku. Holenderski projektant przemyslowy Luud Schimmelpennink, postanowil wraz ze swoimi znajomymi zebrac 50 rowerow, przemalować je na biało, a następnie rozstawić w mieście do darmowego użytku publicznego. Ten program został nazwany „White Bicycle Plan”. Niestety większość z tych rowerow została skradziona.

**Picture 3 Left side, Luud Schimmelpennink Picture 4 Right side, "White Bicycle Plan"**

#### Second generation

Autorami tego modelu sa Morten Sadolina i Ole Wessung. Opracowali oni model wypożyczalni rowerowej, który zakladal darmowe wypożyczenie pojazdu rowerowego w zamian za kaucje. Depozyt ten był w postaci monet, które następnie odblokowywaly dostep do rowerow, jest to podobny system do tego który znamy z hipermarketów oraz ich wózków sklepowych. Pierwsza pula dostepnych rowerow była zlokalizowana w Danii oraz liczyła 26 jednostek transportowych wraz z 4 stacjami w latach 1991 – 1993. Kolejnym dużym etapem w roku 1995 było wprowadzenie do obiegu 800 jednostek transportowych w Kopenhadze, a ten system nosil nazwe „Bycyklen”.



**Picture 5 Bike-Sharing second generation**

#### Third generation

Generacja trzecia sklada sie ze stacji dokujących gdzie można wypożyczyć pojazd rowerowy, a następnie odstawić go w dowolnej stacji należącej do tego samego Bike-Sharing system. Stacje sa wyposażone w stojaki, które posiadaja mechanizmy zwalniające oraz blokujące pozostawione tam rowery tylko i wyłącznie systemem komputerowym. Wypożyczenie roweru polega na identyfikacji zarejestrowanej osoby karta programu członkowskiego. Taki model wypożyczania został opracowany przez Hellmut Slachta i Paul Brandstätter w latach 1990 – 1992 jako „Public Velo”, natomiast po raz pierwszy został wdrożony pod nazwa „Bikeabout” w roku 1996 przez Uniwersytet Portsmouth i Radę Miasta Portsmouth w Anglii.



**Picture 6 Bike-Sharing Third generation**

#### Fourth generation

Generacja o ktorej mowa posiada swoje dwa warianty. Pierwszy z nich to w istocie rowery gotowe do wypożyczenia generacji czwartej czyli takiej gdzie rowery posiadaja swoje stacje dokujące tak jak w generacji drugiej lub trzeciej, natomiast pojazdy rowerowe sa wyposażone w blokady które umozliwiaja samodzielne podjecie decyzji przez użytkownika czy pozostawić rower w dedykowanej stacji dokującej lub zaparkowanie w dowolnym miejscu przez skorzystanie z wyposażonej blokady. Natomiast wspomniany wcześniej, drugi wariant generacji czwartej to rowery objęte programami typu bike sharing noszące miano generacji piatej. Roznica polega w tym, ze wariant nazwany generacja piata nie posiada dedykowanych stacji dokujących lecz posiada tylko i wyłącznie blokady, znane już z pierwszego wariantu generacji czwartej. Cala generacja czwarta została opracowana przez niemieckie przedsiębiorstwo kolejowe i logistyczne, „Deutsche Bahn” w 1998 roku w celu wykorzystania automatycznie wygenerowanych cyfrowych kodow uwierzytelniających służących do zautomatyzowanego blokowania i odblokowywania rowerow. Ich autorski system w roku 2000 nosil nazwe „Call a Bike”, polegal on na odblokowaniu roweru przy uzyciu wiadomości SMS lub polaczenia telefonicznego, aby w późniejszych latach zrestrukturyzować oraz uruchomić w pełni sprawna aplikacje do zainstalowania na telefon. W Polsce pierwsze programy posiadające w swojej flocie rowery pochodzące z generacji czwartej, pojawily się w roku 2015 w Krakowie pod nazwa „Wavelo” oraz w roku 2017 w Warszawie o nazwie „Acro-bike”



**Picture 7 Bike-Sharing Fourth generation**

(Bicycle-sharing system, 2024)

### Analysis

Analizą mozemy nazwac proces kiedy rozbijamy zlozony podmiot na czynniki pierwsze lub mniejsze części aby docelowo wyciagnac konkretne wnioski oraz uzyskac lepsze zrozumienie omawianej problematyki. Analiza to slowo pochodzące z okolo 1500 do 300 roku przed narodzeniem Chrystusa z jezyka Ancient Greek, ktore brzmialo „analusis”, a oznaczalo „breaking up” lub „an untying”. Przez ten zakres na osi czasu mogliśmy zaobserwować podzial na cztery okresy; „Mycenaean” Greek który trwal w przedziale 1500–1200 rokiem przed Chrystusem, „Dark Ages” którego czas przypadal na lata 1200–800 przed Chrystusem, the „Archaic or Epic period” obowiazujacy w latach 800–500 przed Chrystusem, aby finalnie zakonczyc okresem the Classical period który trwal najkrócej, rozpoczynając od  500 do 300 roku przed narodzeniem Chrystusa. Technika ta oficjalnie i pod swoja nazwa była wykorzystywana do nauki matematyki, rozwiazywania problemów logicznych, natomiast proces ten był wykorzystywany znacznie wcześniej lecz pod bliżej nieokreślona forma przez jakiekolwiek myślące istoty żyjące we wszechświecie. Przykladem jest polowanie lub obrona przed zostaniem upolowanym, proces wyboru odpowiedniego miejsca w którym dany osobnik powinien się znaleźć aby upolować ofiare, lub wybor bezpiecznego i ustronnego miejsca to również proces strategicznej analizy. Natomiast w literaturze i sztuce bardzo czesto pojawia sie pojecie „deconstructive analysis” lub „critical analysis”. Stwierdzenia te polegaja na poszukiwaniu znaczen docelowo ukrytych przez autorow lub artystow. Taki rodzaj analizy przeswietla dzielo sztuki w celu zrozumienia glebszego sensu lub znaczenia ukrytego pod oslona przenosni, cenzury, lub wielu innych podobnych sposobow maskowania glebszego znaczenia tworczosci artysty. Istnieje wiele rodzaji lub odmian analizy, wszystko zalezy od tego co poddajemy omawianemu procesowi analizy.

(Analysis, 2024)

#### Data Analysis

Termin „Data Analysis”, jak sama nazwa wskazuje odnosi sie do przetrwarzania danych w celu ich lepszego zrozumienia oraz na ich podstawie wyciagniecia celnych wnioskow. Do elementow wykorzystywanych w tym procesie mozemy zaliczyc inspekcje, transformacje, modelowanie, a nawet czyszczenie danych. Przeprowadzanie tego typu analizy pomaga w podejmowaniu decyzji, do podjecia ktorych niezbezne jest wziecie pod uwage wiele kryterii. Aby wykonac taka analize niezbedne jest wykonanie kilku krokow, ktore sa scisle egzekwowane na przyklad przez firmy swiadczace uslugi analityczne. Pierwszym etapem majacym wplyw na efekt koncowy jakim jest wynik przeprowadzonej analizy, to ustalenie wymagan potrzebych do skompletowania wymaganego zakresu danych tworzacych pozniej zbior danych. Przykladowo jezeli uzytkownik koncowy, czyli osoba zlecajaca analize, oczekuje konkretnych wniskow dotyczacych tematyki branzy rowerowej to w zbiorze danych mozemy oczekiwac pojawienie sie danych takich jak modele rowerow, ich wagi, rozmiarow i wiele innych. Natomiast nie powinnismy oczekiwac wystepowania informacji niedotyczacych zlecanej analizy, w tym przypadku moglyby to byc przykladowo dane medyczne, ktore maja malo wspolnego z tematyka rowerowa. Po ustaleniu wymagan dotyczacych jakie informacje powinien zawierac zbior danych, nastepuje kolekcjonowanie danych w oparciu o wymagania ustalone w poprzednim kroku. Dane powinny byc kolekcjonowane z wielu mozliwie dostepnych zrodel, w celu unikniecia z gory zalozonej stronniczosci wyniku analizy danych. Dane moga przybierac rozne formy, od tekstowych, przechodzac przez numeryczne dane, a konczac na danych graficznych. Kolejnych krokiem jest etap nazwany „Data processing”. Data processing polega na ulozeniu zebranych danych wynikajacych z poprzednich etapow, w taki sposob aby umozliwic przeprowadzenie analizy. Przykladowo, jezeli analiza dotyczy danych numerycznych, nalezy umiejscowic dane w wierszach i kolumnach w tabeli, wtedy mowimy o ustrukturyzowanych danych. Strukturyzacja danych zazwyczaj nastepuje w programach do przechowywania danych, stworzonych do celow analitycznych lub statystycznych, miedzy innymi takich jak program formy „Microsoft” z pakietu „Office” o nazwie „Excel”. Czyszczenie danych to kolejny bardzo wazny punkt przygotowywania danych do analizy, poniewaz zebrane dane moga zawierac liczne duplikaty, a nawet bledy, ktore w pozniejszym etapie maga miec ogromny wplyw na wynik analizy, uniemozliwajac dotarcie do slusznych wnioskow z przeprowadzanej analizy. Przykladowo, w momencie zestawienia danych finansowych, potrzebnych do przeprowadzenia analizy portfela klienta zlecajacego analize, moga wystapic duplikaty posiadanych dobr, a w rzeczywistosci istnieja one w pojedynczej ilosci. Taki blad moze skutkowac falszywie zawyzona informacja finansowa o posiadanym majatku, a nastepnie prowadzic do problemow prawnych na przyklad zwiazanych z kwestiami podatkowymi. Finalny etap analizy może nastapic za pomocą modelowania oraz algorytmow które zazwyczaj sa matematycznymi równaniami, które pomagają w lepszym zrozumieniu problematyki oraz stawianej tezy w ramach zebranego zbioru danych. Natomiast przy aktualnej technologii najczęściej korzysta się z programów wykorzystujących uczenie maszynowe aby lepiej przeswietlic zgromadzony zbior danych, a następnie przedstawić celne wnioski na podstawie przeprowadzonej analizy.

(Data analysis, 2024)

### Machine learning

Uczenie maszynowe, czesto nazwyane za pomocą skrótu „ML” to gałąź specjalizacji jaka jest artificial intelligence, która również bardzo często jest nazywana przy uzyciu skrótu „AI”. ML ma za zadanie wykonać proces „myslowy” przy pomocy roznych algorytmow oraz AI, który jest wytrenowany na podstawie dostarczonego zbioru danych w postaci tekstowej, numerycznej lub graficznej. W tradycyjnym programowaniu człowiek pisze wszystkie zasady, według których komputer ma działać. W uczeniu maszynowym komputer sam tworzy te zasady, analizując przykłady. Na początku aby wytrenować model, wymagany będzie zbior danych zawierający dane związane z problemem który chcemy rozwiazac. Przykladem może być model przy pomocy którego chcemy rozpoznawać koty, w takim przypadku należy przygotować zestawienie dużej ilości zdjęć kotow oraz innych zwierzat. Kolejnym etapem jest trenowanie modelu który się uczy na podstawie dostarczonych danych. W trakcie trenowania, model analizuje dane i stara się znaleźć wzorce, które pomagają rozpoznać, co jest przedstawine na zdjęciach. Po procesie trenowania, można przystapic do testowania. Testowanie odbywa się na nowych danych których model wcześniej nie widział, aby sprawdzić jak dobrze odnajduje się w rozpoznawaniu zalozonych obiektow przy uzyciu znalezonych wzorcow. Procesz uczenia można przyrownac do nauki dziecka przez pokazywanie mu przedmiotow takich jak banan, jabłko, pomarańcz, podkreslajac nazwe przedstawianej rzeczy. Dziecko powinno znaleźć konkretne wzorce, takiej jak to, ze banany najczęściej sa żółte oraz podłużne lub jabłka sa czerwone i okrągłe. Problem może się pojawić w momencie jeżeli caly czas będziemy przedstawiać czerwone jabłko, a w momencie testu pokazemy jabłko o kolorze zielonym. Dlatego w procesie kompletowania zbioru danych jest bardzo ważne to aby dane były rozne, jeżeli przedstawiamy zdjęcia kotow, to nie może się tam znajdować tylko ten sam kot. W momencie gdy model jest wytrenowany oraz przetestowany, a wyniki sa zadowalające, model jest gotowy do uzycia w praktyce.

(Oxford, 2019)

#### History of Machine Learning

Osoba którą możemy nazwac inicjatorem sztucznej inteligencji to Alan Turing. W roku 1950 Turing zaproponowal test Turinga jako sposób sprawdzenia maszyny czy potrafi myslec jak człowiek. Test polega na pewnego rodzaju grze nazywanej „grze w imitacje”. W eksperymencie biora udział 3 postaci, człowiek, maszyna oraz sedzia. Sedzia to osoba która inicjuje oraz prowadzi test zadając pytania za pomocą interfejsu którym może być przykładowo czat tekstowy. Interfejs jest niezbędny w tym eksperymancie ponieważ sedzia nie wie która postac z która rozmawia jest człowiekiem, a która maszyna. Po zadaniu pytania oraz otrzymaniu odpowiedzi, sedzia analizuje odpowiedzi, a po wykonanej serii pytan nastepuje wynik testu Turinga, czyli moment kiedy sedzia wskazuje jednoznacznie która postac to człowiek, a która to maszyna. Natomiast jeżeli sedzia nie będzie w stanie jednoznacznie wskazać, maszyna zdaje test Turinga. Celem takiego testu była ocena zdolności maszyn do wykazywania się inteligencja. Turing uwazal, ze jeżeli maszyna potrafi prowadzic rozmowe w sposób nie możliwy do odróżniania od rozmowy z człowiekiem to można uznac, ze maszyna jest inteligentna.

(Alan Turing, 2024)

**Picture 8 Left side, Alan Truing Picture 9 Right side, Working Alan Turing**

W roku 1957 Frank Rosenblatt stworzył perceptron, pierwszy algorytm uczenia maszynowego inspirowany neuronami w ludzkim mózgu. Był to wczesny model sieci neuronowej. Praca Rosenblatta nad perceptronem była przełomowa, ponieważ po raz pierwszy pokazała, że maszyny mogą się uczyć i adaptować na podstawie doświadczenia. Mimo że klasyczny perceptron miał swoje ograniczenia, takie jak niemożność rozwiązania problemów nieliniowych, przykladem jest problem XOR, który jest operacja logiczna zwracajaca wynik prawdziwy tylko wtedy kiedy jedno z dwóch wejść jest prawdziwe, niestety Perceptron Rosenblatta, czyli prosty model sieci neuronowej, działa dobrze tylko w przypadku problemów, które można rozwiązać za pomocą jednej prostej linii. Ponieważ problem XOR nie jest liniowo separowalny, perceptron nie potrafi go poprawnie rozwiązać. Mimo to klasyczny perceptron stał się fundamentem dla rozwoju bardziej zaawansowanych architektur sieci neuronowych. Wkład Franka Rosenblatta w rozwój sztucznej inteligencji jest kluczowy, a jego badania nad perceptronem uformowaly fundamenty pod rozwój nowoczesnych metod uczenia maszynowego, które rewolucjonizują nasze podejście do rozwiązywania złożonych problemów w różnych dziedzinach nauki i technologii.



**Picture 10 Frank Rosenblatt**

Jednym z pierwszych odnotowanych programów komputerowych, które potrafią się uczyc na swoich doświadczeniach to program do gry w warcaby autorstwa Arthur Samuel. Arthur to Amerykanski pionier inżynieryjny z dziedziny technologii komputerowej oraz wykorzystywania AI. Wspomniany program jego autorstwa, opracowal w 1959 roku. Program używał techniki zwanej "self-play”, grając tysiące partii przeciwko sobie. Dzięki temu mógł analizować różne strategie i uczyć się, które ruchy są najlepsze w różnych sytuacjach. Samuel wprowadził algorytmy i funkcje oceny, które pozwalały programowi ocenić, jak dobra jest dana pozycja na planszy. Program używał tych ocen do podejmowania lepszych decyzji podczas gry. Jego koncepcje dotyczące samodzielnego uczenia się komputerów były rewolucyjne i zainspirowały wielu przyszłych badaczy. Arthur Samuel jest często nazywany jednym z ojców uczenia maszynowego. Jego innowacyjne podejście do samouczenia się komputerów i gry w warcaby miało ogromny wpływ na rozwój AI. Dziś jego prace są nadal cytowane i studiowane jako podstawowe teksty w dziedzinie sztucznej inteligencji i uczenia maszynowego.

“As a result of these experiments one can say with some certainty that it is now possible to devise learning schemes which will greatly outperform an average person and that such learning schemes may eventually be economically feasible as applied to real-life problems.”

(Samuel, 1959)



**Picture 11 Arthur Samuel**

Wszystkie wyzej wymienione postaci oraz ich dokonania lub odkrycia prowadza nas do czasow w ktorych zyjemy aktualnie. Na codzien uczenie maszynowe jest szeroko stosowane w roznych dziedzinach, od medycyny po marketing. Co chwile wiadomosci obiega informacja o udostepnieniu do uzytku publicznego nowego modelu powszechnie znanego „Chat GPT” autorstwa firmy „OpenAI”. Kazde z osiagniec przybliza ludzkosc do zaawansowanej sztucznej inteligencji.

Historia uczenia maszynowego to historia stopniowego rozwoju, algorytmow oraz co za tym idzie, technologii. Przez lata gromadzone dane, moga byc analizowane automatycznie oraz prowadzic do nowych wnioskow oraz odkryc naukowych. Od wczesnych prac Alana Turing i Franka Rosenblatt, przez rozwoj sieci neuronowych, az po wspolczesne techniki deep learning, uczenie maszynowe nieustannie sie rozwija i zyskuje na znaczeniu w naszym codziennym zyciu.

(Foote, 2021)

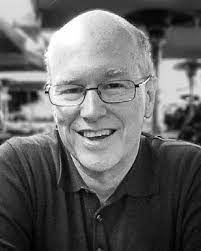
### Refactoring

Refactoring is the name of a technique involving the restructuring of an existing body of code, interfering with its structure, without changing the operation and functionality of the script. This procedure is very popular among programmers in order not to duplicate repeatedly occurring parts of the code, but to call a specific one method using variables, thus making it universal. As a result, the script takes the form of a clear, readable, and transparent code, which takes up much less code than before the refactoring procedure. The first known use of the term refactoring took place in 1990 in the published literature. The authors of the article were William Opdyke and Ralph Johnson. Martin Flower is a British software developer. His experience in the industry is passed on to the next generation in many public statements, publications, and books. One of those books is a book titled "Refactoring: Improving the design of Existing Code," published in 2002 in collaboration with the aforementioned William Opdyke.

(Code Refactoring, 2023)

“Help in understanding the code also helps me spot bugs. I admit I'm not terribly good at finding bugs. Some people can read a lump of code and see bugs, I cannot. However, I find that if I refactor code, I work deeply on understanding what the code does, and I put that new understanding right back into the code. By clarifying the structure of the program, I clarify certain assumptions I've made, to the point at which even I can't avoid spotting the bugs. 49 It reminds me of a statement Kent Beck often makes about himself, "I'm not a great programmer; I'm just a good programmer with great habits." Refactoring helps me be much more effective at writing robust code.”

(Flower, Beck, Opdyke, & Brant, 2002)

**Picture 12 Left side, Martin Flower Picture 13 Right side, William Opdyke**

# Description of analysis.

## Description introduction

Analiza zbioru danych “Bike sharing” została przeprowadzona w dwóch podejściach. Pierwszym podejściem było przeswietlenie zbioru danych „Bike sharing” manualnie, przy pomocy narzedzi takich jak program „Excel” pochodzący z pakietu „Office” autorstwa firmy „Microsoft”. Następnie za pomocą dostępnych w tym programie formuł analitycznych umożliwiających transformacje, formatowanie, modelowanie danych, prowadzących do ciekawych wyników, zostały wyciągnięte wnioski końcowe. Natomiast drugie podejście to pozostawienie analizy danych zawartych w zbiorze danych „Bike sharing” programowi z dziedziny Machine Learning, który generuje wykresy oraz wypisuje wszystkie interesujące dane wynikowe w zrozumialy dla użytkownika sposób. Te analizy pozwalają lepiej zrozumieć, kiedy i dlaczego rowery są wypożyczane, co może być przydatne dla zarządzających wypożyczalniami rowerów do lepszego planowania i zarządzania zasobami.

## Manual analysis description

Analiza danych z pliku „day.csv” zostanie podzielona na kilka etapów, z krótkim opisem każdej analizy, metodą wizualizacji oraz potencjalnymi wnioskami, jakie można wyciągnąć.

### Seasonal analysis

Na początek porównana zostanie liczba wypożyczeń rowerów w różnych porach roku. Przygotowany wykres słupkowy pokaże, ile razy rowery były wypożyczane wiosną, latem, jesienią i zimą. Będzie można wtedy zauważyć, w których miesiącach wypożyczenia są najczęstsze, co może wynikać z lepszej pogody w pewnych porach roku.

### Annual analysis

Porównane zostaną wypożyczenia w dwóch różnych latach: 2011 i 2012. Wykres słupkowy pokaże całkowitą liczbę wypożyczeń dla każdego roku. Będzie można zobaczyć, czy popularność wypożyczalni rowerów rośnie, czy maleje. Jeśli liczba wypożyczeń w 2012 roku jest większa, można przypuszczać, że usługa zyskuje na popularności.

### Impact of working days and holidays

Zbadane zostanie, jak liczba wypożyczeń różni się między dniami roboczymi a weekendami i świętami. Wykres słupkowy pokaże różnice. Można spodziewać się, że w dni robocze wypożyczeń jest mniej, ponieważ ludzie są w pracy, natomiast w weekendy i święta wypożyczeni prawdopodobnie jest więcej z powodu większej ilości wolnego czasu.

### Weather influence

Porównana zostanie liczba wypożyczeń rowerów przy różnych warunkach pogodowych. Wykres słupkowy pokaże, jak różne typy pogody wpływają na wypożyczenia. Będzie można zauważyć czy w dni słoneczne i bezchmurne jest więcej lub mniej wypożyczeń niż w dni deszczowe, co pozwoli zrozumieć wpływ pogody na zachowania użytkowników.

### Monthly analysis

Zbadane zostanie, jak liczba wypożyczeń zmienia się w poszczególnych miesiącach roku. Wykres liniowy pokaże tę zmianę. Może się okazać, że latem i wiosną wypożyczeń jest więcej niż zimą, co może być związane z lepszą pogodą i wyższymi temperaturami.

### Influence of humidity and wind speed

Zbadane zostanie, jak wilgotność i prędkość wiatru wpływają na liczbę wypożyczeń. Dwa wykresy punktowe, jeden dla wilgotności i drugi dla prędkości wiatru, pokażą te zależności. Może się okazać, że wysoka wilgotność i silny wiatr zniechęcają do wypożyczania rowerów.

### Analysis by days of the week

Przeanalizowane zostanie, jak liczba wypożyczeń zmienia się w zależności od dnia tygodnia. Wykres słupkowy pokaże, ile razy rowery były wypożyczane w każdy dzień tygodnia na przestrzeni dwóch lat. Można będzie zauważyć czy wpływ na ilość wypożyczeń może mieć na przykład fakt dojazdu użytkowników do pracy jeżeli wypozyczen byłoby więcej wciągu pracowniczego tygodnia niż dni weekendowych.

### Comparison of registered and casual users

Porównana zostanie liczba wypożyczeń przez użytkowników zarejestrowanych i okazjonalnych. Wykres słupkowy pokaże różnice między tymi grupami. Można będzie zauważyć czy zarejestrowani użytkownicy wypożyczają rowery częściej, co mogłoby wynikać z niższych kosztów za wynajem lub innych korzyści dla takich użytkowników.

### The impact of holidays on rentals

Pod analizę postawiona zostanie teza, czy liczba wypożyczeń różni się w dni świąteczne. Wykres słupkowy pokaże różnice. Może okazać się, że w święta jest mniej wypożyczeń, bo ludzie spędzają czas z rodziną, albo tych wypożyczeń będzie więcej, bo mają więcej wolnego czasu na rekreację.

## Automated analysis description

Analiza danych z wykorzystaniem Machine Learning to zaprzęgniecie obecnie dostępnej technologii do wykonywania złożonej analizy w stosunkowo krótkim czasie, porównując do analizy wykonanej manualnie przez człowieka. Oczywiście trzeba mieć na względzie poziom trudności oraz zawiłości wykonywanej analizy, ponieważ istnieją sytuacje w których to człowiek dominuje nad maszyną i technologią. Natomiast w przypadku analizowania zbioru danych „Bike sharing”, który zawiera informacje rozciągnięte na przestrzeni dwóch lat, technologia sprawia, ze analiza ta nabiera kolorow, a czasem nawet głębszego znaczenia. Proces takiej zautomatyzowanej analizy przewiduje kilka kluczowych etapow, a każdy etap jest niezbedy do uzyskania finalnego kroku jakim sa wyciagniete wnioski przeprowadzonej analizy.

### Data upload

Dane są wczytywane z pliku o rozszerzeniu „CSV” do programu zapisanego w jezyku Python. Dane dotyczą wynajmu rowerów i zawierają informacje o różnych cechach, takich jak temperatura, wilgotność, dzień tygodnia i tym podobne.

### Data preparation

Dane są przygotowywane do analizy. Skalowane są wartości kolumn, takich jak temperatura, aby były one w odpowiednich jednostkach. Następnie usuwane są kolumny, które nie są potrzebne do analizy, na przykład kolumna z datą.

### Modeling

W tym etapie wykorzystywana jest regresja liniowa. Jest ona stosowana, aby określić, jak różne cechy, takie jak temperatura, wilgotność, dzień tygodnia i tym podobne, wpływają na liczbę wynajętych rowerów. Model jest trenowany na całym zestawie danych, co umozliwia dokładniejsze oszacowanie wpływu poszczególnych cech na liczbe wynajętych rowerow. Co więcej takie podejście przez trenowanie modelu na całym zbiorze danych umozliwia lepsze zrozumienie wszystkich relacji miedzy cechami.

### Model results

Po trenowaniu modelu analizowane są współczynniki regresji. Wartości te wskazują, jak silnie każda cecha wpływa na liczbę wynajętych rowerów. Na przykład wysoki współczynnik dla temperatury oznacza, że temperatura ma duży wpływ na wynajem rowerów. Dotatkowo wyniki prezentowane sa w formie liczb dodatnich, a tazke ujemnych, co pozwala wyznaczyć zależność zmiany wartości cech z ich wpływem progresyjnym lub regresyjnym na liczbe wynajętych rowerow. Przykladowo, wzrost wspolczynnika mowiacego o pogorszeniu się warunków pogodowych prawdopodobnie uzyska ujemny wynik wspolczynnika regresji. Należy te informacje traktować w sposób taki, ze pogorszenie się warunków pogodowych zmniejsza liczbe wynajętych rowerow, a im wieksza wartość bezwzgledna takigo wspolczynnika regresji tym cecha przypisana do tego wspolczynnika, znacznie bardziej wpływa, wręcz dyktuje ilość wynajętych rowerow.

### Visualization of results

Tworzone są wykresy w celu ułatwienia interpretacji wyników. Macierz korelacji pokazuje, jak różne cechy są ze sobą powiązane. „Heatmap” wizualizuje te zależności, co pozwala zobaczyć, które cechy są ze sobą silnie powiązane, a które mniej. Dodatkowe wykresy takie jak słupkowe lub punktowe, najciesciej sa wykorzystywane do wizualizacji korelacji jednej z cech z cecha interpretowana jaka jest liczba wynajętych rowerow.

### Chart analysis

Ostatni etap, dojście do którego nie byłoby możliwe bez wykonania poprzednich, to analiza wygenerowanych wykresow. Wnioski z takiej analizy to wazny moment każdej analizy, ponieważ bledne zrozumienie wartości wynikowych lub popełnienie bledow na etapach przygotowywania lub modelowania danych, może nieść ze sobą poważne konsekwencje. Przykladowo gdyby wynik takiej analizy miał wplywac na restrukturyzacje strategii lub nawet przebranżowienie firmy zlecającej analizę, niosłoby to ze sobą ogromne, zbędnie generowane koszty. W celu zestawieniea wynikow obrazujących szersze spojrzenie na relacje miedzy cechami znajdującymi się w zbiorze danych, dobrze sprawdza się macierz korelacji. Pozwala zobaczyć, jak różne czynniki wpływają na siebie nawzajem. Na przykład można jednocześnie zauważyć, że liczba wynajętych rowerów jest silnie powiązana z temperaturą, a temperatura jest silnie powiązana z porą roku. Natomiast wykresy punktowe Pokazują relację między konkretną cechą, a liczbą wynajętych rowerów. Przy ich pomocy można wykazać czy w cieplejsze dni rowery są częściej wynajmowane.

## Dataset

Zbiór danych dotyczy systemu wypożyczania rowerów „Capital Bikeshare” w Waszyngtonie D.C., USA, i obejmuje dane historyczne z lat 2011-2012. Systemy wypożyczania rowerów umożliwiają automatyczne wypożyczanie i zwracanie rowerów, a dane z tych systemów mogą być używane do analizy mobilności w mieście. Dodatkowo, dane te mogą być pomocne w badaniach nad wpływem różnych czynników środowiskowych na wypożyczanie rowerów.



**Picture 14 Capital Bikeshare rental bicycle**

Dane dotyczace wynajmowanych rowerow firmy „Capital Bikeshare” z lat 2011-2012 znajduja się w pliku korzystającego z rozszerzenia „csv” pod nazwa „day.csv”. Każdy wpis w tym zestawie danych reprezentuje kolejny dzień z zakresu dwuletniego. W efekcie korzystamy ze zbioru danych posiadającego 731 wpisow, a każdy wpis posiada 16 wartosci. Wartosci te zostały nazwane w pierwszym wierszu.

- *instant*: record index  
- *dteday*: date  
- *season*: season (1: spring, 2: summer, 3: fall, 4: winter)  
- *yr*: year (0: 2011, 1: 2012)  
- *mnth*: month (1 to 12)  
- *holiday*: whether the day is a holiday or not (1: yes, 0: no)  
- *weekday*: day of the week (0: Sunday, 1: Monday, ..., 6: Saturday)  
- *workingday*: whether the day is a working day (1: yes, 0: no, meaning it's a weekend or holiday)  
- *weathersit*: weather situation:  
1: Clear, Few clouds, Partly cloudy  
2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds  
3: Light Snow, Light Rain + Thunderstorm + Scattered clouds  
4: Heavy Rain + Ice Pellets + Thunderstorm + Mist, Snow + Fog  
- *temp*: normalized temperature in Celsius (values divided by 41)  
- *atemp*: normalized feeling temperature in Celsius (values divided by 50)  
- *hum*: normalized humidity (values divided by 100)  
- *windspeed*: normalized wind speed (values divided by 67)  
- *casual*: count of casual users  
- *registered*: count of registered users  
- *cnt*: count of total rental bikes including both casual and registered users

**Figure 1 List of attributes in dataset “Bike sharing”**

Autorem zbioru danych jest Hadi Fanaee-T, natomiast właścicielem jest “Laboratory of Artificial Intelligence and Decision Support (LIAAD), University of Porto, INESC Porto”. Jest to jednostka badawcza powiązana z Uniwersytetem w Porto, jednym z największych i najbardziej prestiżowych uniwersytetów w Portugalii. „LIAAD” jest znane ze swoich badań w dziedzinie sztucznej inteligencji oraz systemów wspomagania decyzji. Systemy wspomagania decyzji to miedzy innymi analiza decyzji i zarządzanie ryzykiem, inteligencja biznesowa, techniki optymalizacyjne oraz symulacja i modelowanie. Celem „LIAAD” jest dążenie do zaawansowania społecznej wiedzy oraz technologii w dziedzinach sztucznej inteligencji oraz systemow wspomagiania decyzji.

(LIAAD, 2024)



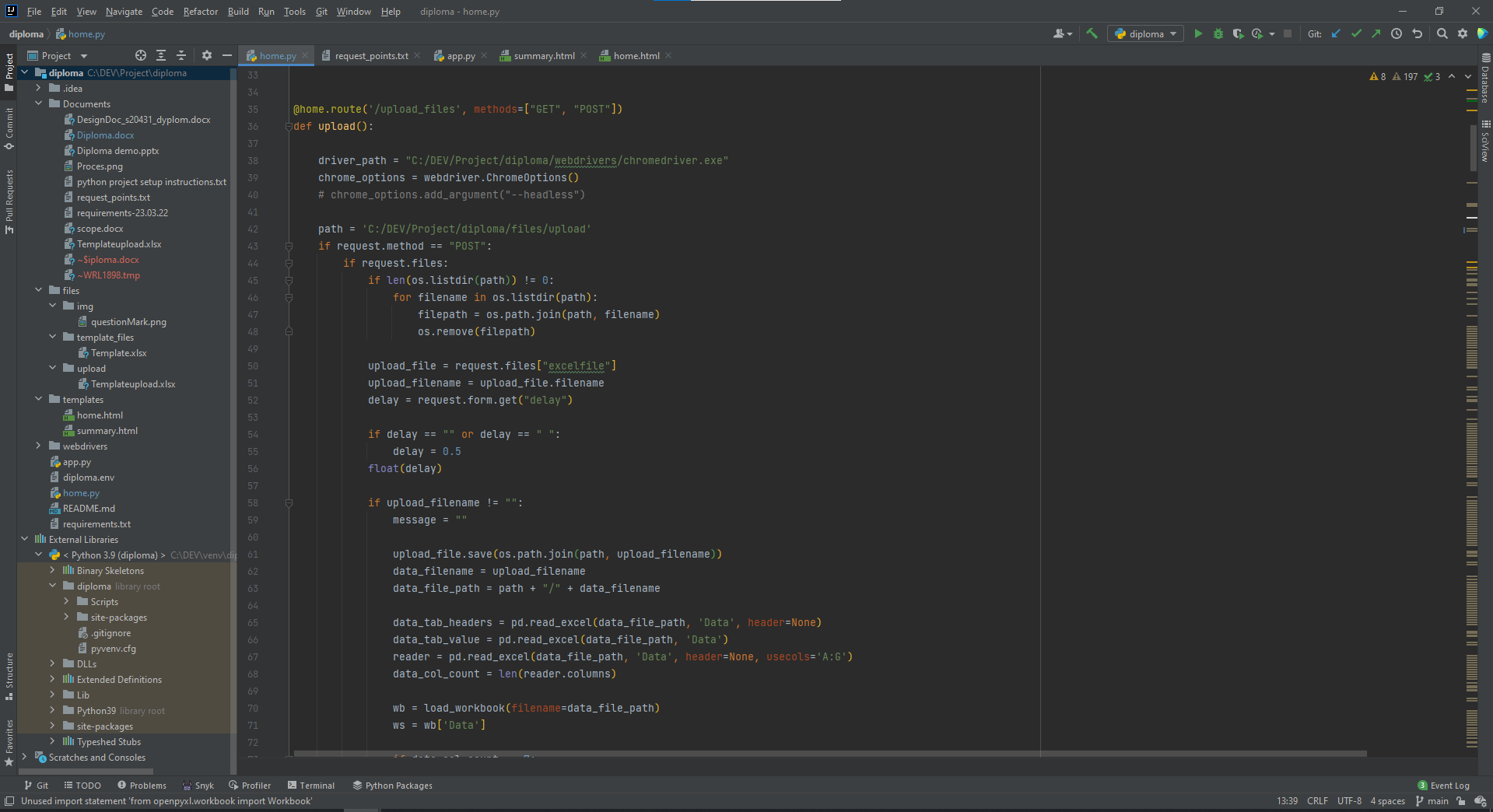
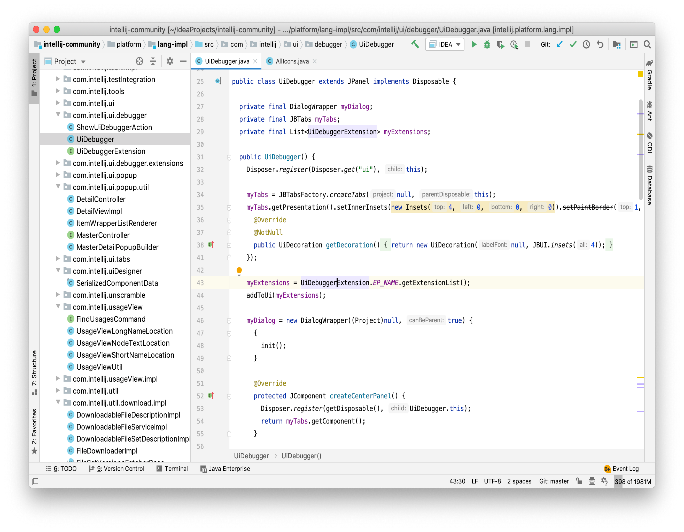
**Picture 15 Hadi Fanaee-T**

Omawiany zbior danych posiada wytyczne aby w razie wykorzystywania go w publikacjach musi zostać cytowane zgodnie z poniżej prezentowana publikacją.  
„Fanaee-T, Hadi, and Gama, Joao, "Event labeling combining ensemble detectors and background knowledge", Progress in Artificial Intelligence (2013): pp. 1-15, Springer Berlin Heidelberg, doi:10.1007/s13748-013-0040-3.”

## Tools

### IntelliJ IDEA

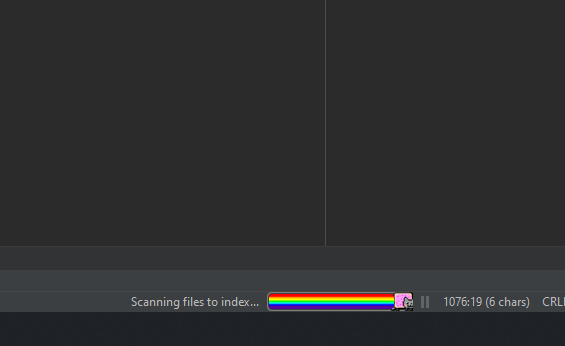
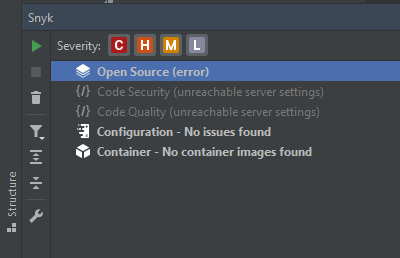
Aby stworzyć zaawansowane lub proste programy IT, które są potrzebne programiście lub organizacji, konieczny jest edytor kodu. Jednym z najlepszych edytorów jest IntelliJ IDEA, komercyjne środowisko programistyczne, które może kompilować napisany kod. Ten edytor został stworzony przez firmę "JetBrains" w języku Java. Pierwsza wersja została wydana na początku 2001 roku i zawierała narzędzia do refaktoryzacji kodu. Teraz IntelliJ IDEA oferuje wiele przydatnych wskazówek i skrótów klawiszowych, które ułatwiają pracę programisty. IntelliJ IDEA obsługuje 19 języków programowania i automatyzacji, takich jak "Python", "Java", "Scala" oraz języki do tworzenia stron internetowych, jak "HTML" i "CSS". Dzięki temu programiści mogą pracować w różnych językach w jednym, spójnym środowisku. IntelliJ IDEA współpracuje również z wieloma innymi narzędziami i środowiskami open source, takimi jak "GIT", "SVN", "CVS", "Apache Maven", "Apache Ant" i "JUnit", co znacznie ułatwia zarządzanie projektami i integrację z innymi systemami. Dla wielu fanów IT interesujący jest także ciemny motyw interfejsu tego edytora, który zmniejsza zmęczenie oczu podczas długotrwałej pracy. IntelliJ IDEA jest nie tylko narzędziem do pisania kodu, ale także wsparciem dla programisty dzięki funkcjom takim jak inteligentne podpowiedzi, automatyczne uzupełnianie kodu, analiza błędów w czasie rzeczywistym i wszechstronne opcje debugowania. To wszystko sprawia, że jest to narzędzie cenione zarówno przez początkujących, jak i doświadczonych programistów. Edytor ten jest również regularnie aktualizowany, co zapewnia dostęp do najnowszych technologii i najlepszych praktyk w programowaniu. Dzięki aktywnej społeczności użytkowników i szerokiej bazie wiedzy dostępnej online, wsparcie i rozwój umiejętności są na wyciągnięcie ręki. IntelliJ IDEA to inwestycja, która przynosi korzyści w postaci zwiększonej produktywności i jakości kodu.

**Picture 16 Left side, IntelliJ dark theme Picture 17 Right side, IntelliJ default light theme**

Aby rozszerzyć funkcjonalność podstawowej wersji IntelliJ IDEA, dostępnych jest wiele wtyczek. Przykładem jest "Rainbow Brackets" autorstwa "izhangzhihao", która poprawia czytelność kodu poprzez kolorowanie nakładających się nawiasów okrągłych, kwadratowych i klamrowych. Inna wtyczka, "Snyk Security - Code, Open Source, Container, IaC Configurations", analizuje kod napisany przez programistę i generuje raport na temat bezpieczeństwa aplikacji. Raporty są bardzo czytelne, nawet dla początkujących programistów, co ułatwia zrozumienie potencjalnych problemów. "Snyk" oferuje również możliwe rozwiązania dla wykrytych luk bezpieczeństwa. Istnieją również wtyczki rekreacyjne, które mają na celu poprawić humor użytkownikom IntelliJ IDEA. Jedną z nich jest "Nyan Progress Bar" stworzona przez "Dimitry Batkovich", która zamienia pasek ładowania kompilatora na tęczowy ogon kultowego bohatera społeczności programistycznej, jakim jest "Nyan Cat". Dzięki takim dodatkom, praca z edytorem staje się nie tylko efektywna, ale również przyjemna. Rozbudowa możliwości IntelliJ IDEA za pomocą takich wtyczek sprawia, że edytor ten jest jeszcze bardziej wszechstronny i dostosowany do indywidualnych potrzeb użytkowników. Niezależnie od tego, czy chodzi o poprawę czytelności kodu, zwiększenie bezpieczeństwa aplikacji czy wprowadzenie elementu zabawy, dostępne wtyczki mogą znacznie ułatwić i uprzyjemnić codzienną pracę programistów.

(Jetbrains, 2024)

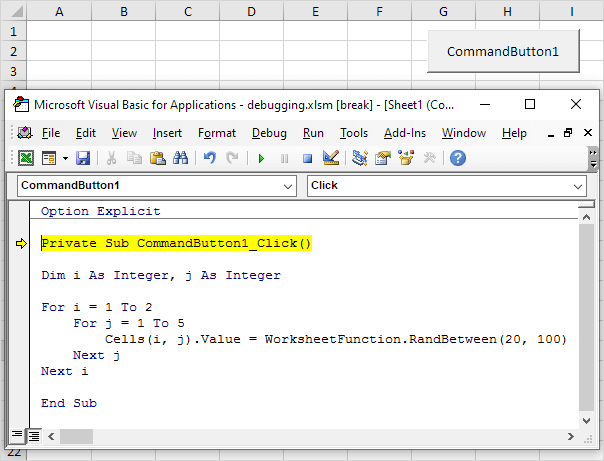
 

**Picture 18 Left side, Nyan Progress Bar Picture 19 Right side, Snyk add-in**

### Excel

Microsoft Excel to popularny arkusz kalkulacyjny dostępny na systemy Windows, macOS, Android i iOS. Umożliwia użytkownikom wykonywanie różnych obliczeń matematycznych oraz prezentowanie danych za pomocą wykresów i tabel przestawnych. Dzięki funkcji Makr, można także automatyzować powtarzalne zadania. Makra tworzy się za pomocą języka Visual Basic for Applications (VBA). Każda nowa wersja Excel wprowadza szereg nowych funkcji i usprawnień, które czynią program bardziej przyjaznym dla użytkownika. Oprócz standardowych funkcji, dostępne są również dodatki, które rozszerzają możliwości programu. Te dodatki, często opracowane przez niezależnych programistów, oferują dodatkowe narzędzia i funkcje, które nie są zawarte w oficjalnej wersji programu. Dzięki tym rozszerzeniom, Microsoft Excel staje się jeszcze bardziej elastycznym i potężnym narzędziem, idealnym do zaawansowanej analizy danych i automatyzacji różnych procesów, co znacząco zwiększa jego wartość zarówno w zastosowaniach biznesowych, jak i codziennych.

(Microsoft, 2024)

**Picture 20 Left side, Excel Worksheet Picture 21 Right side, Visual Basic for Applications**

### Python

Python to język programowania, który oferuje wiele możliwości dzięki licznym narzędziom i bibliotekom. Jego główną zaletą jest przejrzystość i prostota kodu, co sprawia, że jest łatwy do nauki i użycia. Python pozwala na pisanie kodu w sposób jasny i zwięzły. W Pythonie nie trzeba deklarować typów zmiennych, takich jak integer czy string, co ułatwia pisanie kodu. Dzięki swojej popularności, Python działa na wielu systemach operacyjnych. Jego rozwój jest prowadzony przez "Python Software Foundation", organizację non-profit. Standardowa wersja Pythona, "CPython", jest napisana w języku C. Istnieją też inne wersje, takie jak JPython, napisany w Javie, i IronPython dla .NET. Python powstał na początku lat 90., a jego twórcą jest Guido van Rossum. Język został stworzony jako następca języka ABC w Centrum Matematyki i Informatyki w Amsterdamie (CWI - Centrum voor Wiskunde en Informatica). Ciekawostka jest to, że nazwa Python, którą autor nazwal ten jezyk programowania, nie pochodzi od nazwy gatunku węża. Autor wybierając nazwe dla tego jezyka programistycznego był wielkim fanem serialu komediowego emitowanego przez brytyjska telewizje na antenie „BBC” pod tytułem "Monty Python's Flying Circus". Python jest używany w wielu dziedzinach, takich jak analiza danych, sztuczna inteligencja, tworzenie stron internetowych, automatyzacja zadań i tworzenie gier. Frameworki takie jak Django i Flask pomagają w tworzeniu stron internetowych, a biblioteki takie jak NumPy, pandas i TensorFlow są używane w analizie danych i uczeniu maszynowym. Dzięki dużej społeczności i wielu dostępnych narzędziom, Python jest idealnym wyborem zarówno dla początkujących, jak i doświadczonych programistów. Jest to język, który można łatwo dostosować do różnych projektów i zadań, co czyni go bardzo wszechstronnym narzędziem w programowaniu.

(Python Software Foundation, 2024)

**Picture 22 Left Side, CWI building Picture 23 Right side, Guido van Rossum**

### Pandas

Pandas to biblioteka do jezyka programistycznego Python, stworzona do obsługi i analizy różnorodnych danych, zwłaszcza tych numerycznych używanych w zadaniach analizy danych. Została stworzona i wydana na licencji "BSD" w 2008 roku przez Wes McKinney, który wtedy pracował w "AQR Capital Management". Motywacja do stworzenia tej biblioteki wynikała z potrzeby posiadania potężnego narzędzia do efektywnej analizy danych finansowych. Nazwa "Pandas" pochodzi od terminu "panel data", który odnosi się do zbiorów danych zawierających obserwacje z różnych okresów czasowych w ekonometrii.

(Pandas, 2024)



**Picture 24 Wes McKinney**

### Openpyxl

When the program or application that the programmer designs requires the use of "Excel" spreadsheets from the "Microsoft Office" package, the best solution for automated handling of this program is the "Openpyxl" library. "Openpyxl "is a well-developed library containing many useful spreadsheet control tools. The main reason for the creation of this package was the lack of existence of similar solutions capable of opening and editing files saved in formats with extensions "xls", "xlsx", "xlsm", "xltx and "xltm" in a simple, clear and logical way. Unfortunately, in the world of computer technology there are many threats in the form of viruses or spyware, which can often be encountered very well hidden in Excel spreadsheets coming from an unknown source. According to the official website of the manufacturer, the library "Openpyxl" does not offer any support in this regard. The only possible option suggested by the owners of the software is to install an additional package that is "defusedxml". The authors of the "Openpyxl" software are Eric Gazoni and Charlie Clark. The first version available for public use was version 1.7.0 released in late 2013.

(Openpyxl, 2022)

**Picture 14 Left side, Eric Gazoni Picture 15 Right side, Charlie Clark**

### HTML

The language "Hyper Text Markup Language (HTML)" belongs to the category of markup languages. The predominant variety of text formatting occurring in the "HTML" language is descriptive markup. Distinguish these varieties of markup of text fragments or transmitted data, which define the role of fulfilling the extracted information stored in the document. "HTML" was created for the creation of web pages "World Wide Web (WWW)", it allows to describe the structure of the information displayed inside a web page. The main tools available are the formulation of hyperlinks, lists, tables, headings, paragraphs and the possibility of embedding multimedia or file objects in the text of the document. There are also non-database elements, such as interactive data forms. The beginning of the language "HTML" can be marked in 1980, when physicist Tim Berners-Lee created a prototype of a hypertext information system called "ENQUIRE". The purpose of this system was to organize and make available documents related to scientific research. Users could then access the shared documentation located in a completely different, remote location. In 1989 Tim Berners-Lee competed with Robert Cailliau. The object of their competition was a parallel proposal for hypertext information systems based on the Internet. This was the beginning of an adventure leading to the development of an amazing technology, which today's Internet users know very well, because in 1990 Tim and Robert joined forces and presented a jointly prepared proposal for the "World Wide Web" project. They achieved success as their proposal was accepted by CERN. The first publicly available documentation of the language "HTML," then called "HTML Tags," was issued in 1991. The original structure of this language contained 22 tags adapted to mark the elements contained in a document. "HTML", although it was written on the basis of the "SGML" language, did not have in it a formal definition of the original markup language for easy storage of text documents with the possibility of simple transfer, display and printing in various electronic data transmission systems.

(HTML, 2023)

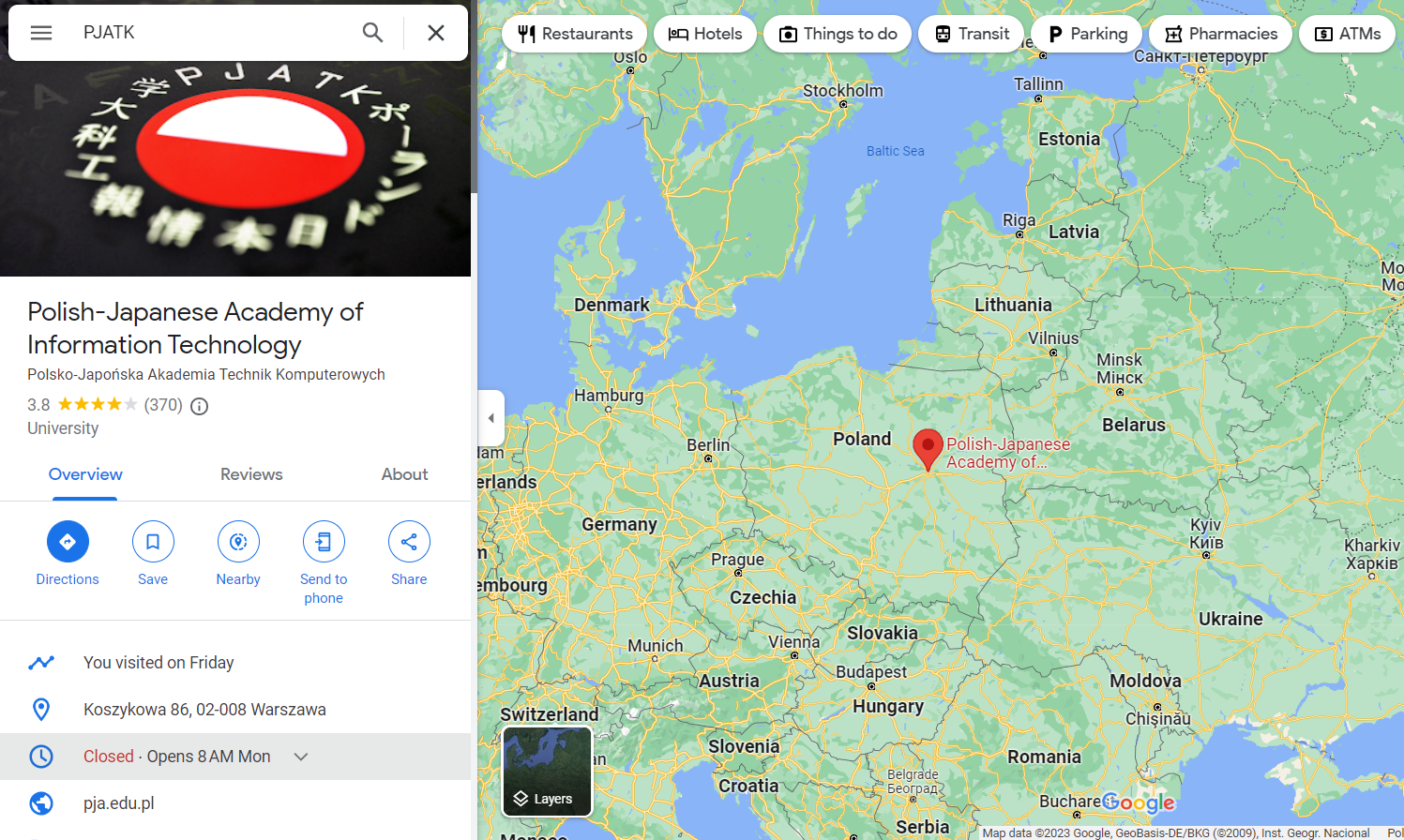
 

**Picture 16 Left side, Timothy Berners-Lee Picture 17 Right side, Robert Cailliau**

### Google Maps

The website, necessary for carrying out the main functionality of the application "GeoFinderUI" is "Google Maps". The company "Google", which owns the website in question, has a huge amount of data stored in databases, new items of which are added every second. The application "Google Maps" uses this data, which has in it information about the location located on the globe, and then presents it graphically in the form of marked points on the world map. In order to make it easier for the user to find a location of interest, the "Google Maps" application has been equipped with a search engine. After typing a phrase and finding it, the user is shown a list of the best matching search results. Unfortunately, such a search is not always consistent with the user's intentions. For example, the database on which the application operates is so large that there is a very high probability of finding two or even several locations having the same name. In order to identify the correctness of the search results, it is necessary to have a person or a program with analytical skills. Such scenarios do not occur every time, but every day there are many new business establishments, companies, green parks, streets, entertainment or tourist facilities, whose owners want their location to be available in the databases of the company "Google". This is the main problem that the user may encounter while using the application. Against this phenomenon, "Google Maps" is in constant development and improvement so that the user can get the best quality result of his search. However, the development is not only limited to the search terms entered by the user and the delivery of the results, the developers of "Google" are constantly creating new functionalities, which are made available to the public. Since the first version of this application, available to the public, released in 2005, many things have changed, for example, a very useful function "Street View" has been added, allowing the user to appear in the center of a designated location, available for this function, and view this place in the form of a virtual reality created with the help of interlocked images from a camera capable of taking so-called "360 photos." "360 photos" are combined projections acquired from at least two camera lenses, resulting in an expanded angle of view of up to 360°. Such images are usually saved in special formats that allow reading and displaying the file in computer programs dedicated to this technology and giving the user the opportunity to observe the selected frame, rather than the frame imposed by the author of the photograph.

(Google Maps, 2022)

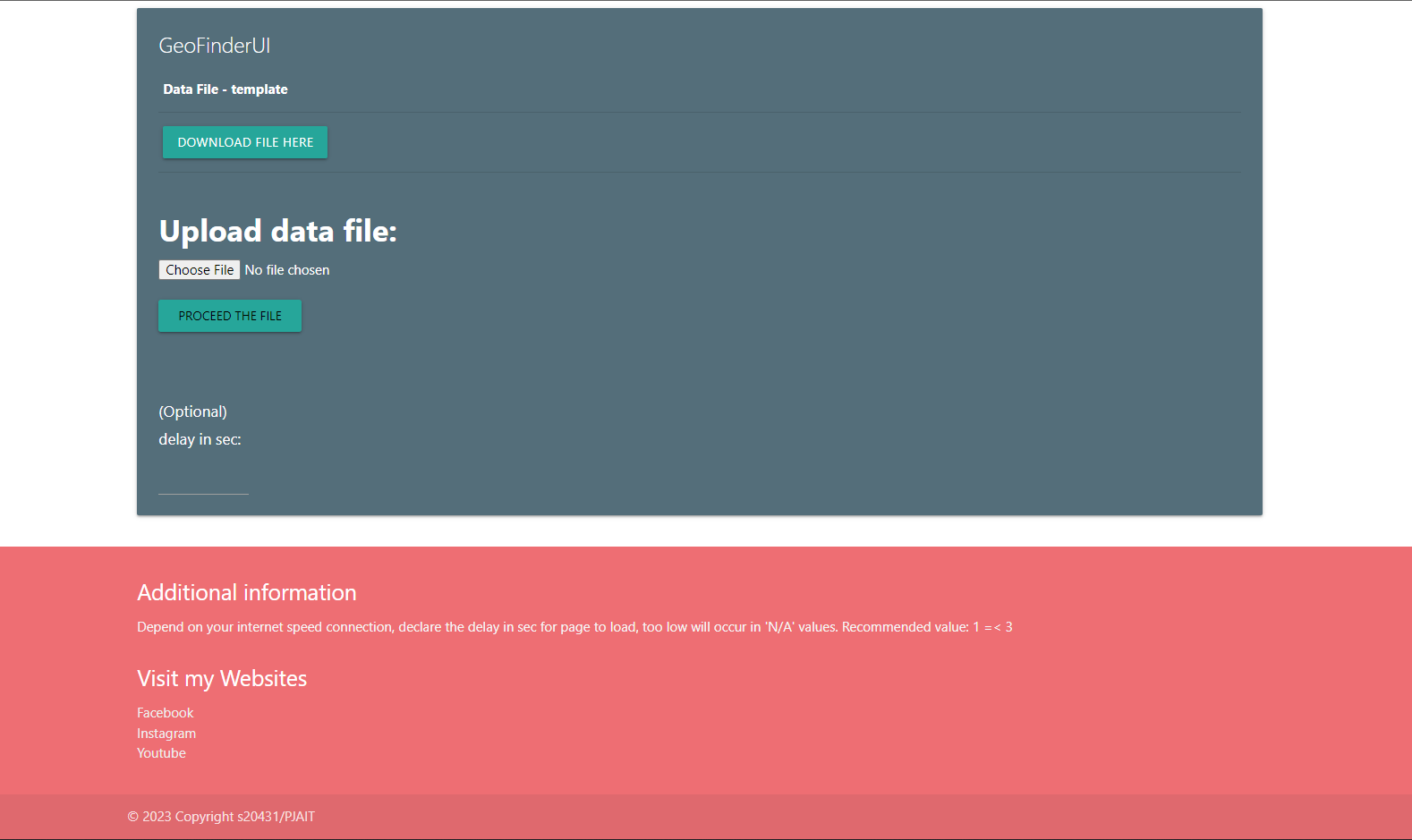


**Figure 2 Google Maps search result**

### Materialize

In order to better present your site and save a huge amount of time in the process, it is worth using ready-made "CSS" and "JavaScript" style packages prepared by the portal "materializecss.com". "Materialize" is a website that offers many categories of style packages for "HTML" pages. On the start page, the user is informed about the steps required before taking advantage of the benefits of building your own website and giving it a visual appearance through specially prepared tags of the "Materialize" portal. Each step is presented in a precise and logical way so that anyone would be able to understand and use this website. In order for the style packages to be available for use in the "HTML" document, you must add in the "<head> </head>"section two lines of text containing links to the page where the style packages are placed. The first is a link to the "CSS" packages, and the second is a link to the "JavaScript" packages. From the left side of the "Materialize" portal, the user can search for the package he is interested in from among many categories. In each category there are many examples with different appearance of the style package for a particular object. After finding the package of interest, it is necessary to copy the "HTML" code visible to the user under the example of the appearance of the styling of the "HTML" object, then place it in your document responsible for the layout of the web page and the content therein, written in "Hyper Text Markup Language".

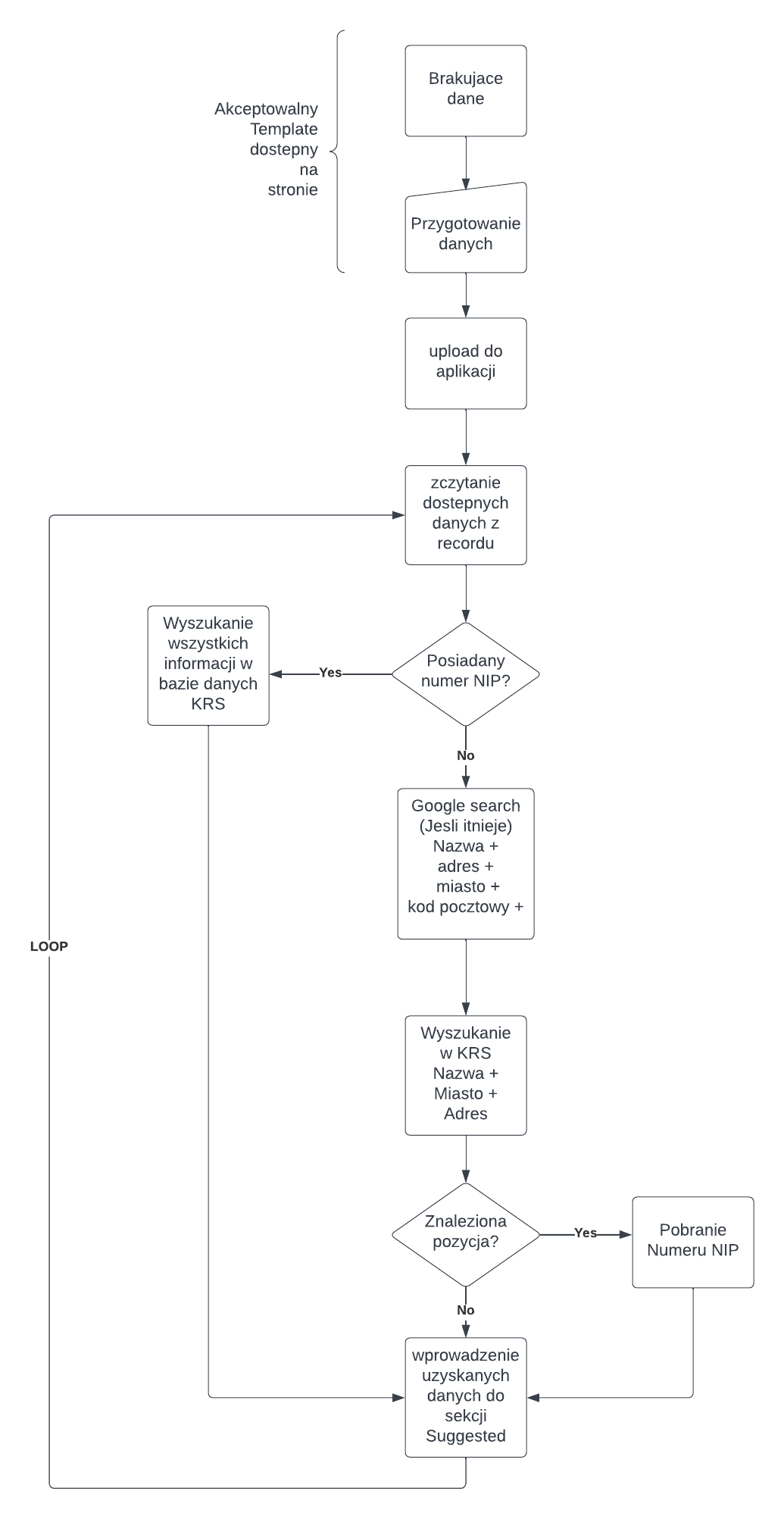
(Materialize, 2023)



**Figure 3 GeoFinderUI application home page with Materialized styling packages**

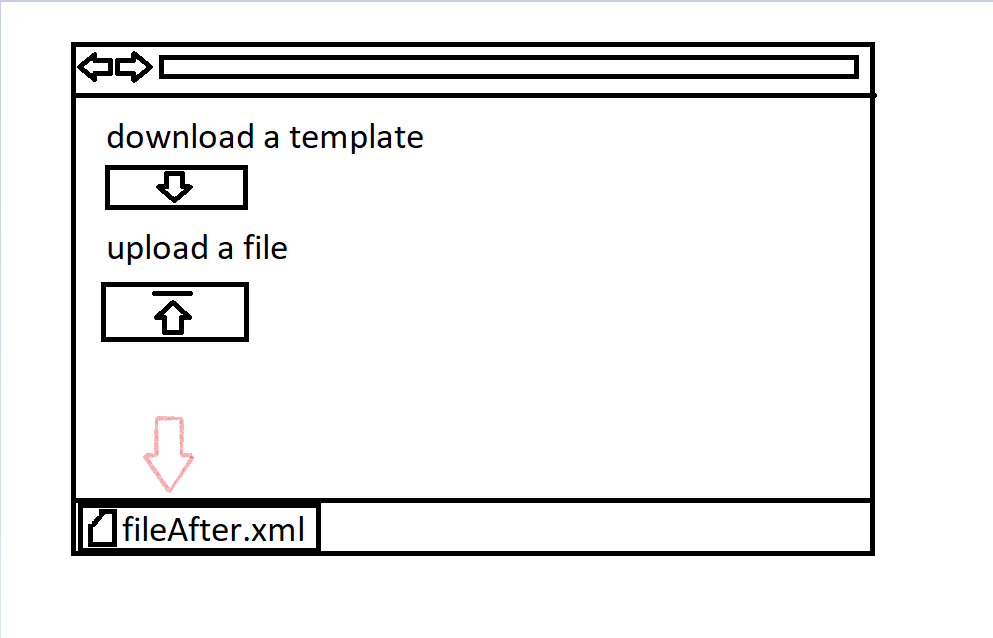
# Functionality process assumptions

The first stage of this work was the graphical formulation of the web application process. This process was mainly based on the entered "NIP" number or its search for a specific location. Such information is generally available on the National Court Register website "(KRS)". Unfortunately, these assumptions also included the use of web scraping technology. Without exception, the "KRS" website blocks access to the data on it after detecting automated movement through the site using a bot. Web scraping technology is the creation of such a bot capable of reading and searching the source code of a website.

****

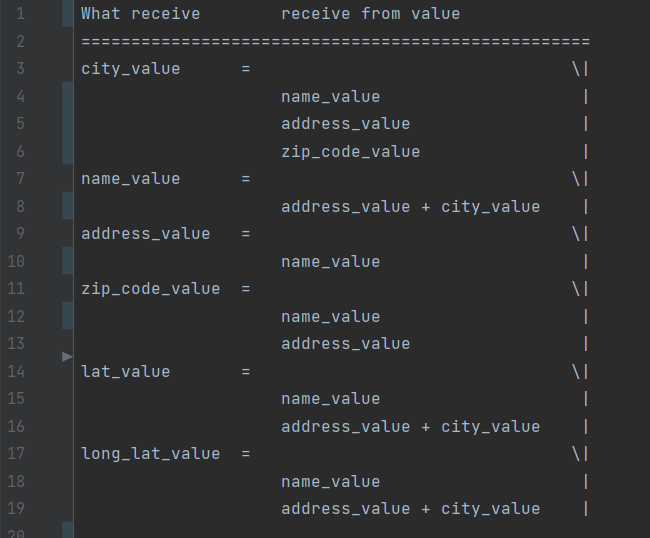
**Figure 4 the original diagrammatic process of the operation of the GeofinderUI application**

Such an obstacle required a restructuring of the functioning of the application. Since then, the main source of information acquisition became a portal having access to the largest geolocation database, which is "Google Maps". Before starting work on the programming of the application "GeoFinderUI", it was necessary to find the right tools that would help in obtaining the full functionality of the program and provide the best quality result to the user. The first requirements were to store and provide data for palliation in an understandable and simple way for the user. For this purpose, the choice fell on the spreadsheet "Excel". Due to the decision to use ".xls" files, the program, which was to be written in the programming language "Python", required appropriate libraries capable of handling spreadsheets. Library "Pandas" and library "Openpyxl" are well suited for such tasks. With their help "GeoFinderUI" reads and writes found data from the Internet in spreadsheets. The next step was to select a suitable tool for operating websites. The tool should make it possible to enter phrases into the search window of a browser or a web page, and then, among the search results, find an object in the source code of a web page, containing the searched information. The "Selenium" library meets the above requirements and is the most important part of the functionality of the "GeoFinderUI" program. The web application is not just a back-end script, for this we needed to visualize the layout of the page. A simple but succinct document, outlining the purpose to which the application was to aspire from the beginning, was prepared in the program "Paint" available in the operating system "Windows". This program is an easy to use and very limited in its capabilities graphical editor. However, it was sufficient to create a sample visualization of the appearance of the site.



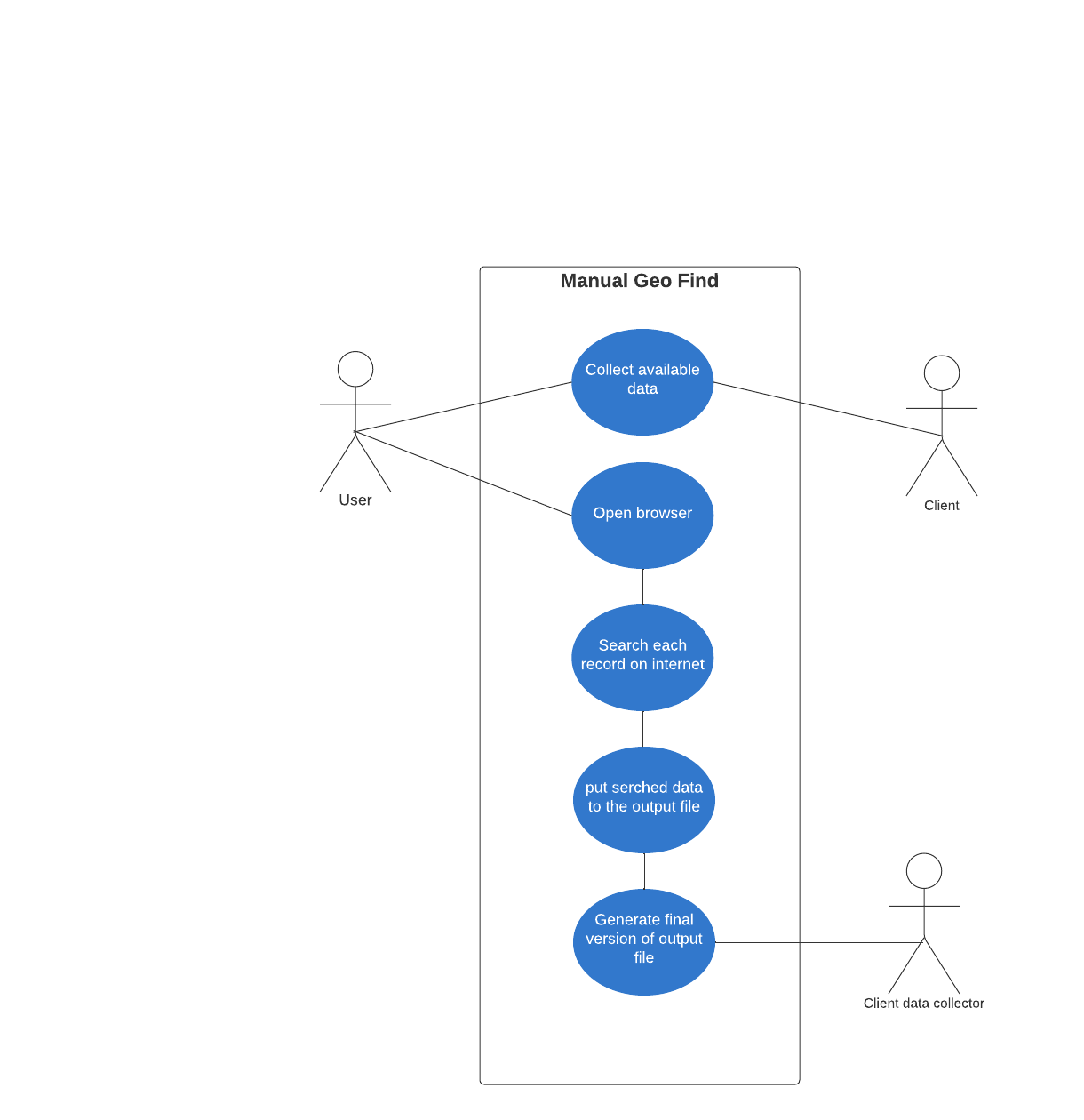
**Figure 5 sample visualization of the GeoFinderUI start page**

The next necessary step was to determine the possibility of receiving data. The purpose of this was to maximize the use of user-submitted data in order to receive the most available information about a given location in the final results prepared by the "GeoFinderUI" application.



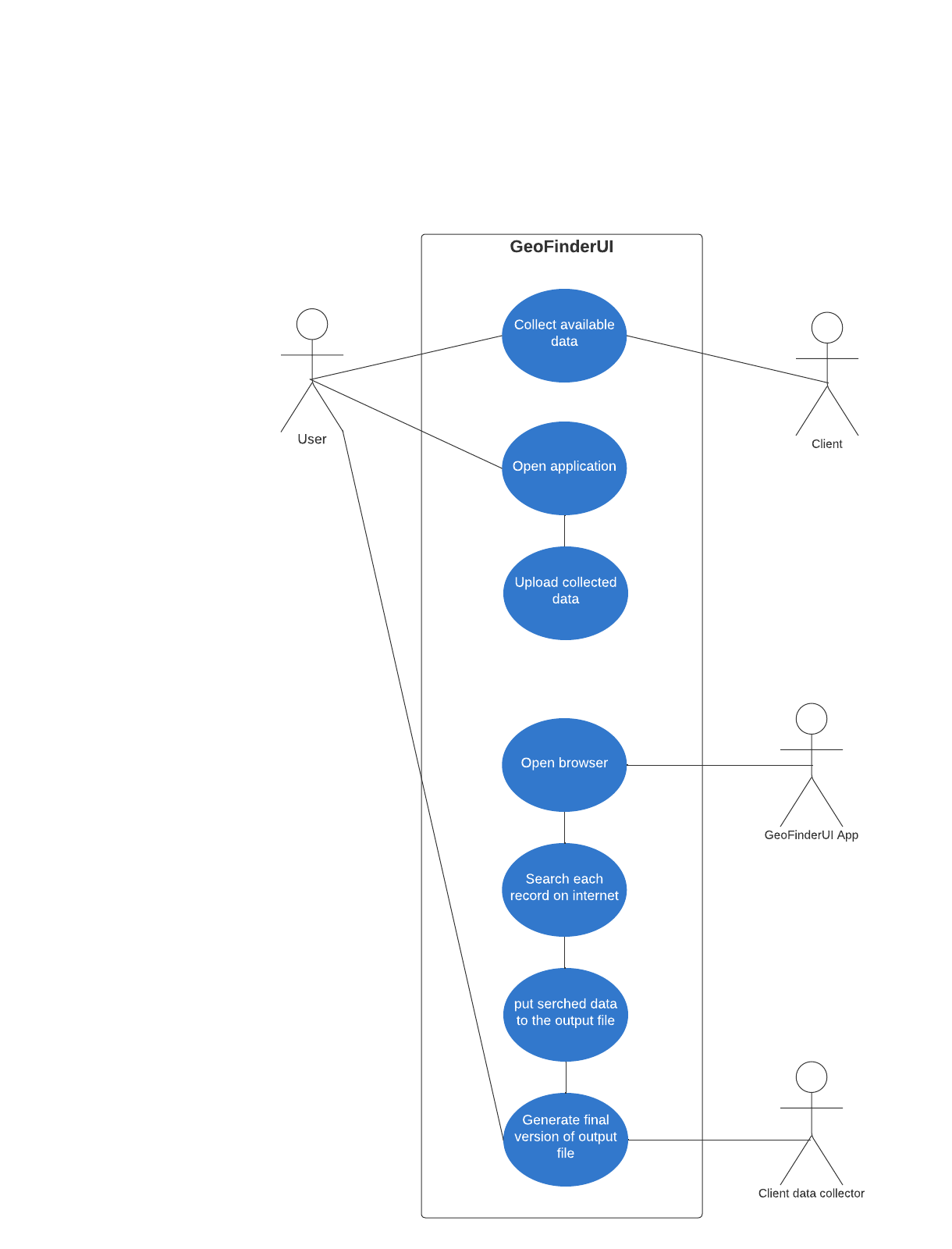
**Figure 6 data acquisition points by GeoFinderUI application**

After collecting such extensive information and documents that deepen the problems of the assumed task, the process of programming the application can begin.



**Figure 7 Manual geo finder use case diagram**

The process of searching for geolocation data carried out manually would be a time-consuming and exhausting process for the person responsible for collecting the data. All elements of the full process, by which we obtain geolocation data, would have to be performed by the user. Starting with the joint collection of input data with the customer. The next steps are up to the user only. These elements are finding a suitable location on the Internet on the basis of the input data provided, and then finding the missing information about the individual place whose data we entered for the search.



**Figure 8 GeoFinderUI use case diagram**

Unlike the manual process of obtaining data on the basis of selected input data about a specific location, the "GeoFinderUI" application independently and automatically searches, collects information and delivers it to a file for suggestive result data.

# Implementation.

## Front-end code

Front-end code is the part of the code responsible for all the elements visible on the page or modelling it.

### Home page

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>GeoFinderUI</title>  
 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">  
 <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>

<style>  
 html {  
 height: 100%;  
 }  
 body {  
 min-height: 100%;  
 display: flex;  
 flex-direction: column;  
 }  
 .content {  
 flex: 1;  
 }  
 footer {  
 margin-top: auto;  
 display: block;  
 bottom: 0;  
 width:100%;  
 height: auto;  
 }  
</style>

</head>

**Code 1 The <Head> section with the style part needed for positioning the content of the GeoFinderUI start page**

The entire user-visible part of the application was written in the "HTML" language. A large role in its appearance is played by styling packages provided by the portal "Materialize". With their help the site acquired a professional and user-friendly appearance. This would not be possible without the addition of 2 lines of code, which are references to the location of the saved styling packages "CSS" and "JS". The references were placed at the top of the source code of the website in the "<head> </head>" section. In this section you can also see the subsection "<style> </style>". The style section, prepared in this way, allows you to arrange the elements on the web page in a universal way with respect to the size of the window in which the page is displayed. When the size of the screen displaying the page is so small that the elements of the page overlap, a scroll bar is displayed on the right side of the window, making it possible to read any information on the page. The "<footer> </footer>" subsection has been given stylistic attributes, forcing this subsection to be displayed in the lowest possible position on the page, but not further away than required by the space provided for the proper display of all elements of the website. The same set of settings can be found on each of the existing subsections of the "GeoFinderUI" application.

<table class="table">  
 <thead>  
 <tr>  
 <th>Data File - template</th>  
 </tr>  
 </thead>  
 <tbody>  
 <tr>  
 <td>  
 <a href="/download\_template"><button class="waves-effect waves-light btn black-text">Download file here</button></a>  
 </td>  
 </tr>  
 </tbody>  
</table>

**Code 2 Part of the code that allows the user to download the official version of the data template acceptable by the application**

In order to provide the best services of the application, a link has been placed on the site that allows downloading the official template file in effect when the user submits data to the application. The link initiates the launch of a dedicated script for downloading files located on the site, written in the "Python" programming language.

<form enctype="multipart/form-data" action="/upload\_files" method="post">  
 <div>  
 <h4> <b>Upload data file: </b></h4>  
 <p><input type="file" name="excelfile" accept=".xlsx, .xls"/></p>  
 </div><br>  
 <div>  
 <input type="submit" id="proceed-xlsx-file" class="btn waves-effect waves-light" value="Proceed the file">  
 </div><br>  
 <div>  
 <h6>(Optional)</h6>  
 <h6>delay in sec:</h6>  
 <p><input class="col s1" type="number" step="0.5" min="1" name="delay"/></p>  
 </div>  
</form>

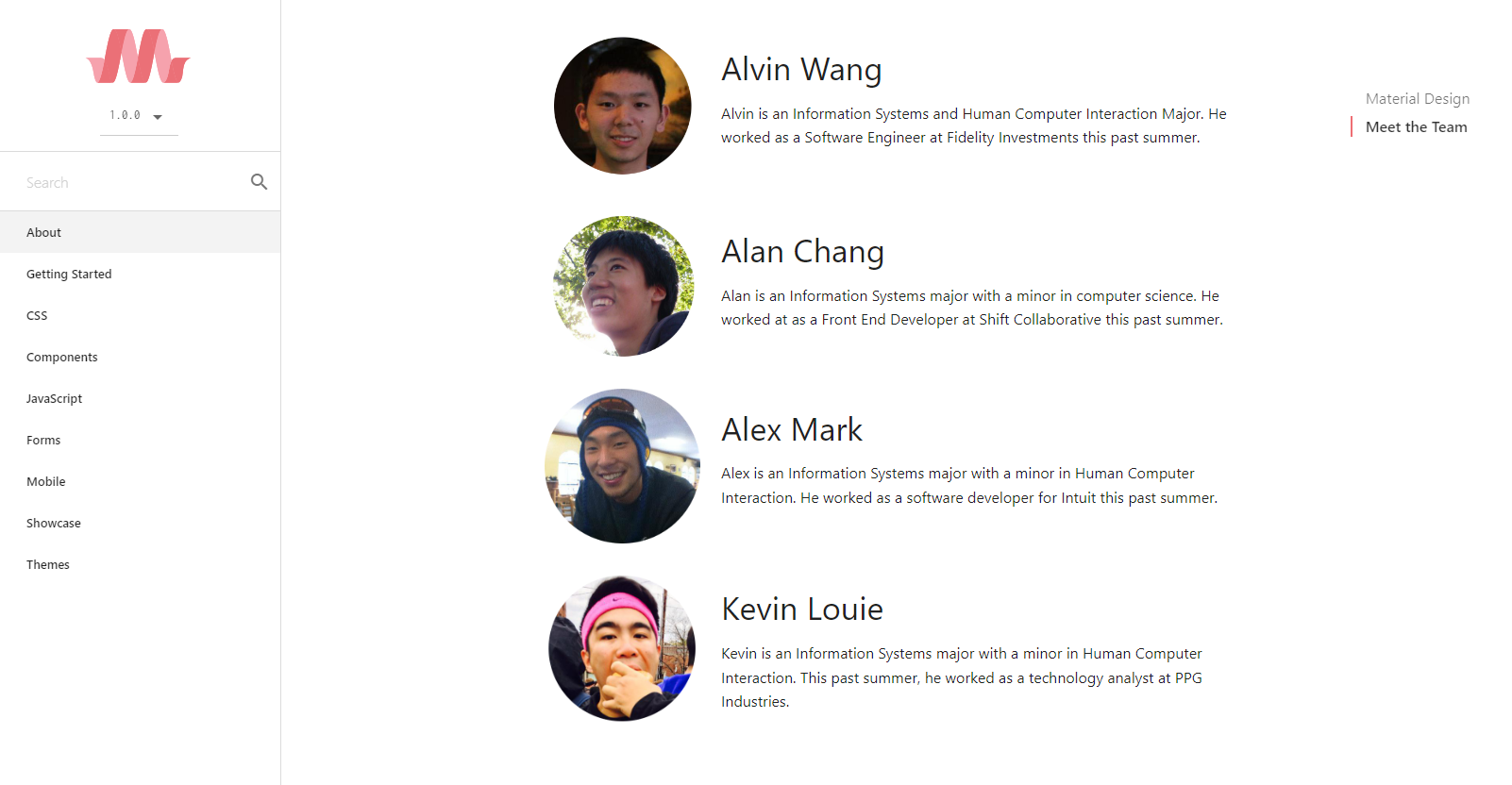
**Code 3 The form section needed for receiving and sending files and information to the back-end code**

In order to easily receive and process files or optionally provided information about the additional time needed to load the page, and then transfer them to the back-end code, the application uses the "<form> </form>" sections. With the help of this section, websites easily manage to provide the necessary information used in programming languages as variables. The "form" section contains 3 "<input>" fields, the first of which is used to upload the selected file by the user, the second approves the selections and starts the main functionality of the application, and the last is an unnecessary field informing the program about the delay in obtaining data in order to load the full pages.

<footer class="page-footer">  
 <div class="container">  
 <div class="row">  
 <div class="col s12 m12">  
 <h5 class="white-text">Additional information</h5>  
 <p class="grey-text text-lighten-4">Depend on your internet speed connection, &#013;declare the delay in sec for page to load, &#013;too low will occur in 'N/A' values. Recommended value: 1 =< 3</p>  
 </div>  
 <div class="col s12">  
 <h5 class="white-text">Visit my Websites</h5>  
 <ul>  
 <li><a class="grey-text text-lighten-3" href="https://www.facebook.com/m3mo.wujec/">Facebook</a></li>  
 <li><a class="grey-text text-lighten-3" href="https://www.instagram.com/m3mo\_w/">Instagram</a></li>  
 <li><a class="grey-text text-lighten-3" href="https://www.youtube.com/@m3mo\_w">Youtube</a></li>  
 </ul>  
 </div>  
 </div>  
 </div>  
 <div class="footer-copyright">  
 <div class="container">  
 © 2023 Copyright s20431/PJAIT  
 </div>  
 </div>  
 </footer>  
</body>  
</html>

**Code 4 The part of the code responsible for the footer visible on each page of the "GeoFinderUI" web application**

Each page displayed within the application "GeoFinderUI" has in it a part of the code responsible for generating a page footer. The footer located on the main page has additional information that cannot be found on any other page. This information is intended to explain to the user what the "delay in sec:" field is used for. Many sites available on the Internet have information about the author of the site or web application. The application "GeoFinderUI" also has information about the author in the form of external links to the author's community portals where information about the author's hobbies, interests and also information about the author's professional experience is published.



**Figure 9 Information about the authors of materializecss.com**

### Summary page

<div class="container">  
 <div class="row">  
 <div class="col s12 m12">  
 <div class="card blue-grey darken-1">  
 <div class="card-content white-text">  
 {% if message %}  
 <span class="card-title">{{ message }}</span>  
 {% endif %}  
 <p>As GeoFinderUI developers, we are extremely happy that you are using our tool.</p>  
 </div>  
 <div class="card-action">  
 {% if check %}  
 <a href="/download\_file"><button class="btn btn-default">Download file here</button></a>  
 {% endif %}  
 <a href="/home"><button class="btn btn-default">return to home page</button></a>  
 </div>  
 </div>  
 </div>  
 </div>  
</div>

**Code 5 The part of the code that contains the information and its layout on the summary page**

The page devoted to the summary of the procedure performed by the "GeoFinderUI" application, has elements that can display other information depending on the result of the procedure performed. The values of the variables are passed to the display via a back-end script as parameters added to the "render\_template()" function from the toolkit that constitutes the "flask" library. The page is able to change its layout at the time of a positive result of data file generation or at the time of a negative result. When the result is negative, the information about will be displayed and the button allowing the user to download the file generated by a script written in the "Python" programming language is not available.

## Back-end code

### Initialization classes

import os  
import time  
  
from flask import Flask, Blueprint, render\_template, *request*, send\_file  
from selenium import webdriver  
from selenium.webdriver.common.keys import Keys  
from selenium.webdriver.common.by import By  
from selenium.common.exceptions import NoSuchElementException  
from selenium.common.exceptions import TimeoutException  
from selenium.webdriver.support.ui import WebDriverWait  
from selenium.webdriver.support import expected\_conditions as *EC*import pandas as *pd*from openpyxl.workbook import Workbook  
from openpyxl import load\_workbook

**Code 6 Part of the code containing the import of the necessary tools used in the "GeoFinderUI" application**

The "home.py" file begins with a section of code devoted to importing libraries and the various tools needed to run the application.

@*home*.route('/')  
def index():  
 return render\_template('home.html')  
  
  
def download(*filepath*):  
 print(*filepath*)  
 return send\_file(*filepath*, as\_attachment=True)  
  
  
@*home*.route('/download\_template')  
def download\_template():  
 *path* = os.path.realpath('files/template\_files/Template.xlsx')  
 return download(*path*)  
  
  
@*home*.route('/download\_file')  
def download\_file():  
 *path* = os.path.realpath('files/download/S\_DataValues.xlsx')  
 return download(*path*)

**Code 7 Side methods in "home.py"**

The first methods found in the "home.py" file are side methods, used to initialize the start page after entering the corresponding "URL" address, and methods that allow downloading files such as "Template.xlsx" and a spreadsheet file holding the end results of the main application process "GeoFinderUI".

@*home*.route('/upload\_files', methods=["GET", "POST"])  
def upload():  
  
 *driver\_path* = os.path.realpath("webdrivers/chromedriver.exe")  
 *chrome\_options* = webdriver.ChromeOptions()  
 *chrome\_options*.add\_argument("headless")  
 *chrome\_options*.add\_argument("disable-gpu")  
  
 *path* = os.path.realpath('files/upload')  
 *path\_download* = os.path.realpath('files/download')

**Code 8 The beginning of the main method responsible for the functionality of the application**

The beginning of the main method responsible for the functionality of the application is the definition of variables necessary for the use of the script. There are functions that return the absolute path of a given location based on its fragment, and a variable that initializes the driver for the "Chrome" browser from "Google". Arguments have also been added to this variable to control the browser in the background so that the automated process of obtaining data from the browser does not appear on the user's screen.

if *request*.files:  
 if len(os.listdir(*path*)) != 0:  
 for *filename* in os.listdir(*path*):  
 *filepath* = os.path.join(*path*, *filename*)  
 os.remove(*filepath*)  
 if len(os.listdir(*path\_download*)) != 0:  
 for *filename* in os.listdir(*path\_download*):  
 *filepath* = os.path.join(*path\_download*, *filename*)  
 os.remove(*filepath*)  
  
 *upload\_file* = *request*.files["excelfile"]  
 *upload\_filename* = *upload\_file*.filename  
 *delay* = *request*.form.get("delay")  
  
 if *delay* == "" or *delay* == " ":  
 *delay* = 0.5  
 float(*delay*)

**Code 9 Part of the code responsible for preparing and cleaning the location for storing data files**

Before starting the process of searching the web pages in order to collect data forming the final result, the program analyses the target space for storing data files and when any file is found it is deleted, such procedure minimizes the risk of an error occurrence, also the location of such a procedure at the beginning of the code has a great impact on negating the potential occurrence of an error, because if the previous user had not brought the data acquisition process to an end, and the code fragment cleaning the location was located at the end of the procedure, the current user could have received results based on the data provided by its predecessor.

*data\_tab\_value* = pd.read\_excel(*data\_file\_path*, 'Data')  
*reader* = pd.read\_excel(*data\_file\_path*, 'Data', header=None, usecols="A:G")  
  
*wb* = load\_workbook(filename=*data\_file\_path*)  
*ws* = *wb*['Data']

*ws*['F1'] = "S\_Name"  
*ws*['G1'] = "S\_Address"  
*ws*['H1'] = "S\_City"  
*ws*['I1'] = "S\_Zip-code"  
  
*driver\_chrome* = webdriver.Chrome(executable\_path=*driver\_path*, options=*chrome\_options*)  
*driver\_chrome*.get("https://www.google.com/maps/")

**Code 10 Part of the code responsible for preparing the data file and launching the automated browser**

To ensure that the user is well informed about the values being entered into the spreadsheet, the program assigns header names that are a combination of the "S" corresponding to the word "Suggested" and the type of data being transmitted, such as address, city or zip-code.

for *record* in range(len(*data\_tab\_value*.index)):  
 *name\_value* = (*data\_tab\_value*.iat[*record*, 0])  
 *address\_value* = (*data\_tab\_value*.iat[*record*, 1])  
 *city\_value* = (*data\_tab\_value*.iat[*record*, 2])  
 *zip\_code\_value* = (*data\_tab\_value*.iat[*record*, 3])

**Code 11 A loop procedure that goes through all the records entered in the data file**

The whole application is based on the "for" loop, the program at this point collects data provided by the user, each iteration is another line in the spreadsheet.

if pd.isna(*name\_value*):  
 if *address\_value* and *city\_value*:  
 *s\_name* = name\_f\_address\_city(*driver\_chrome*, *search*, *delay*, *address\_value*, *city\_value*)  
 if pd.isna(*zip\_code\_value*):  
 if *s\_name* != "N/A":  
 *s\_zip\_code* = zipcode\_f\_name(*driver\_chrome*, *search*, *delay*, *s\_name*)  
 else:  
 *s\_zip\_code* = zipcode\_f\_address(*driver\_chrome*, *search*, *delay*, *address\_value*)  
  
 elif *address\_value* and *zip\_code*:  
 *s\_city* = city\_f\_zipcode(*driver\_chrome*, *search*, *delay*, *zip\_code\_value*)  
 if *s\_city* != "N/A":  
 *s\_name* = name\_f\_address\_city(*driver\_chrome*, *search*, *delay*, *address\_value*, *s\_city*)  
  
if pd.isna(*address\_value*):  
 if *name\_value* and (*s\_address* == '' or *s\_address* == "N/A"):  
 *s\_address* = address\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*)  
 if pd.isna(*city\_value*) and *s\_address* != "N/A" and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_address(*driver\_chrome*, *search*, *delay*, *s\_address*)  
 elif pd.isna(*city\_value*) and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*)  
 if pd.isna(*zip\_code\_value*) and *s\_address* != "N/A" and (*s\_zip\_code* == '' or *s\_zip\_code* == "N/A"):  
 *s\_zip\_code* = zipcode\_f\_address(*driver\_chrome*, *search*, *delay*, *s\_address*)  
 elif pd.isna(*zip\_code\_value*) and (*s\_zip\_code* == '' or *s\_zip\_code* == "N/A"):  
 *s\_zip\_code* = zipcode\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*)  
  
if pd.isna(*city\_value*):  
  
 if *name\_value* and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*)  
 if pd.isna(*address\_value*) and (*s\_address* == '' or *s\_address* == "N/A"):  
 *s\_address* = address\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*)  
 if pd.isna(*zip\_code\_value*) and *s\_address* != "N/A" and (*s\_zip\_code* == '' or *s\_zip\_code* == "N/A"):  
 *s\_zip\_code* = zipcode\_f\_address(*driver\_chrome*, *search*, *delay*, *s\_address*)  
  
 elif *address\_value* and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_address(*driver\_chrome*, *search*, *delay*, *address\_value*)  
 if pd.isna(*name\_value*) and *s\_city* != "N/A" and (*s\_name* == '' or *s\_name* == "N/A"):  
 *s\_name* = name\_f\_address\_city(*driver\_chrome*, *search*, *delay*, *address\_value*, *s\_city*)  
 elif *zip\_code\_value* and *s\_city* == "N/A" and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_zipcode(*driver\_chrome*, *search*, *delay*, *zip\_code\_value*)  
 if *s\_name* == '' or *s\_name* == "N/A":  
 *s\_name* = name\_f\_address\_city(*driver\_chrome*, *search*, *delay*, *address\_value*, *s\_city*)  
 if pd.isna(*zip\_code\_value*) and *s\_address* != "N/A" and (*s\_zip\_code* == '' or *s\_zip\_code* == "N/A"):  
 *s\_zip\_code* = zipcode\_f\_address(*driver\_chrome*, *search*, *delay*, *address\_value*)  
  
 elif *zip\_code\_value* and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_zipcode(*driver\_chrome*, *search*, *delay*, *zip\_code\_value*)  
  
if pd.isna(*zip\_code\_value*):  
 if *name\_value* and (*s\_zip\_code* == '' or *s\_zip\_code* == "N/A"):  
 *s\_zip\_code* = zipcode\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*)  
 if pd.isna(*city\_value*) and *s\_zip\_code* != "N/A" and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_zipcode(*driver\_chrome*, *search*, *delay*, *s\_zip\_code*)  
 elif pd.isna(*city\_value*) and pd.isna(*address\_value*) is not True and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_address(*driver\_chrome*, *search*, *delay*, *address\_value*)  
 if pd.isna(*address\_value*) and (*s\_address* == '' or *s\_address* == "N/A"):  
 *s\_address* = address\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*)  
  
 elif *address\_value* and (*s\_zip\_code* == '' or *s\_zip\_code* == "N/A"):  
 *s\_zip\_code* = zipcode\_f\_address(*driver\_chrome*, *search*, *delay*, *address\_value*)  
 if pd.isna(*city\_value*) and *s\_zip\_code* != "N/A" and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_zipcode(*driver\_chrome*, *search*, *delay*, *s\_zip\_code*)  
 elif pd.isna(*city\_value*) and (*s\_city* == '' or *s\_city* == "N/A"):  
 *s\_city* = city\_f\_address(*driver\_chrome*, *search*, *delay*, *address\_value*)  
 if pd.isna(*name\_value*) and *s\_city* != "N/A" and (*s\_name* == '' or *s\_name* == "N/A"):  
 *s\_name* = name\_f\_address\_city(*driver\_chrome*, *search*, *delay*, *address\_value*, *s\_city*)

**Code 12 Part of code with nested "if" statements that returns final data values**

The program has multiple nested conditional procedures in order to obtain the searched data in the most efficient way. Each data acquisition path is precisely prepared for all possible scenarios of data arrangement. The program is designed and written in such a way that there are 9 ways to get the next information about a given location from each of up to 4 pieces of information provided by the user.

data\_to\_file(*name\_value*, *s\_name*, *address\_value*, *s\_address*, *city\_value*, *s\_city*, *zip\_code\_value*, *s\_zip\_code*, *ws*, *record*)  
  
 *message* = "Success! - file is ready"  
 *check* = 'pass'  
  
 *driver\_chrome*.quit()  
 *wb*.save(*download\_file\_path*)  
 else:  
 *message* = "Please use the official version of the template file available on main page"  
 else:  
 *message* = "Please use the official version of the template file available on main page"  
 else:  
 *message* = "please reload page and make sure to choose the data file"  
 else:  
 *message* = "please reload page and make sure to choose the data file"  
else:  
 *message* = "upload filed!"  
return render\_template('summary.html', message=*message*, check=*check*)

**Code 13 Last part of main class code**

The last part of the main module of the application, enters the data collected by the program into a spreadsheet file and saves it under the name "S\_DataValues.xlsx". The script assigns to the variable "message" the final information about the executed process. The variable "check" is a tag used to generate a page containing a summary. If this variable has the value "pass", the user has access to a button that initiates the download of a file having the final data stored in a spreadsheet named "S\_DataValues.xlsx".

### Callable classes

def cookies(*driver\_chrome*):  
 *delay\_cookie* = 10  
 try:  
 myElem = WebDriverWait(*driver\_chrome*, *delay\_cookie*).until(EC.visibility\_of\_element\_located((By.XPATH, '/html/body/c-wiz/div/div/div/div[2]/div[1]/div[3]/div[1]/div[1]/form[2]/div/div/button')))  
 except TimeoutException:  
 print("Loading took too much time!")  
 try:  
 *driver\_chrome*.find\_element(by=By.XPATH, value="/html/body/div[2]/div[2]/div[3]/span/div/div/div/div[3]/button[2]/div").click()  
 except Exception:  
 pass  
 try:  
 *driver\_chrome*.find\_element(by=By.XPATH, value="/html/body/c-wiz/div/div/div/div[2]/div[1]/div[3]/div[1]/div[1]/form[2]/div/div/button").click()  
 except Exception:  
 pass

**Code 14 Class dedicated to accept cookies on website**

A class thanks to which a browser can approve a pop-up window that requires accepting the storage of cookies emitted by a website visited by the program. With the help of web scraping, the application locates the approval button and then it is remotely clicked.

def address\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*):  
 try:  
 *driver\_chrome*.find\_element(by=By.NAME, value='q').clear()  
 except Exception:  
 pass  
  
 *search*.send\_keys(*name\_value*)  
 *search*.send\_keys(Keys.RETURN)  
  
 page\_to\_load(*driver\_chrome*, *delay*)  
 time.sleep(1)  
 try:  
 *s\_address* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[7]/div[1]/button/div[1]/div[2]').text  
 except NoSuchElementException:  
 *s\_address*="N/A"  
  
 if *s\_address*=="N/A" or *s\_address*=="":  
 try:  
 *s\_address* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Copy address"]').text  
 except NoSuchElementException:  
 *s\_address*="N/A"  
  
 if *s\_address*=="N/A" or *s\_address*=="":  
 try:  
 *s\_address* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Kopiuj adres"]').text  
 except NoSuchElementException:  
 *s\_address*="N/A"

return *s\_address*

**Code 15 A class that returns address value from name value**

Acquiring an address from the place name specified by the user starts with entering the phrase into the search box and then validating it. When the correct value is not found, the program looks for the corresponding information in other locations on the page containing the search results. Then it returns the value assigned to the "s\_address" variable.

def city\_f\_zipcode(*driver\_chrome*, *search*, *delay*, *zip\_code\_value*):  
 try:  
 *driver\_chrome*.find\_element(by=By.NAME, value='q').clear()  
 except Exception:  
 pass  
  
 *search*.send\_keys(*zip\_code\_value*)  
 *search*.send\_keys(Keys.RETURN)  
  
 page\_to\_load(*driver\_chrome*, *delay*)  
 time.sleep(1)  
 try:  
 *s\_city* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[2]/div[1]/div[1]/div[1]/h1/span[1]').text  
 except NoSuchElementException:  
 *s\_city*="N/A"  
  
 try:  
 if *s\_city* != "N/A":  
 *temp\_city* = *s\_city  
 s\_city* = *temp\_city*.split()[1]  
 except Exception:  
 print("too fast")  
  
 if *s\_city*=="":  
 *s\_city*="N/A"  
 return *s\_city*

**Code 16 A class that returns city value from zip-code value**

Getting the city from the postal code is a very accurate way to determine the city in which the location is located. The value obtained from the search consists of the combination of the postal code and the city, for this purpose the program must split the retrieved text and assign to the variable "s\_city" the second part of the text, which has an index of 1.

def name\_f\_address\_city(*driver\_chrome*, *search*, *delay*, *address\_value*, *city\_value*):  
 try:  
 *driver\_chrome*.find\_element(by=By.NAME, value='q').clear()  
 except Exception:  
 pass  
  
 *search*.send\_keys(*address\_value* + " " + *city\_value*)  
 *search*.send\_keys(Keys.RETURN)  
  
 page\_to\_load(*driver\_chrome*, *delay*)  
 time.sleep(1)  
 try:  
 *s\_name* = *driver\_chrome*.find\_element(by=By.CLASS\_NAME, value="bfdHYd Ppzolf")  
 *s\_name* = *s\_name*.get\_attribute("aria-label")  
 except NoSuchElementException:  
 *s\_name*="N/A"  
  
 if *s\_name*=="N/A" or *s\_name*.upper()==*address\_value*.upper():  
 try:  
 *s\_name* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[2]/div[1]/div[1]/div[1]/h1').text  
 except NoSuchElementException:  
 *s\_name*="N/A"  
  
 if *s\_name*=="N/A" or *s\_name*.upper()==*address\_value*.upper():  
 try:  
 *s\_name* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[15]/div/div[2]/div[2]/div[1]/div/div/div/div[1]/div/span').text  
 except NoSuchElementException:  
 *s\_name*="N/A"  
  
 if *s\_name*=="N/A" or *s\_name*.upper()==*address\_value*.upper():  
 try:  
 *s\_name* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[17]/div/div[2]/div[2]/div[1]/div/div/div/div[1]').text  
 except NoSuchElementException:  
 *s\_name*="N/A"  
  
 if *s\_name*=="N/A" or *s\_name*.upper()==*address\_value*.upper():  
 try:  
 *s\_name* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[17]/div/div[2]/div[2]/div[1]/div/div/div/div[1]').text  
 except NoSuchElementException:  
 *s\_name*="N/A"  
  
 if *s\_name*=="N/A" or *s\_name*.upper()==*address\_value*.upper():  
 try:  
 *s\_name* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[19]/div/div[2]/div[2]/div[1]/div/div/div/div[1]/div').text  
 except NoSuchElementException:  
 *s\_name*="N/A"  
  
 if *s\_name*=="N/A" or *s\_name*.upper()==*address\_value*.upper():  
 try:  
 *s\_name* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[15]/div/div[2]/div[2]/div[1]/div/div/div/div[1]/div').text  
 except NoSuchElementException:  
 *s\_name*="N/A"  
  
 if *s\_name*=="N/A" or *s\_name*.upper()==*address\_value*.upper():  
 try:  
 *s\_name* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[27]/div/div[2]/div[2]/div[1]/div/div/div/div[1]/div').text  
 except NoSuchElementException:  
 *s\_name*="N/A"  
  
 if *s\_name*=="N/A" or *s\_name*.upper()==*address\_value*.upper():  
 try:  
 *s\_name* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[18]/div/div[2]/div[2]/div[1]/div/div/div/div[1]/div').text  
 except NoSuchElementException:  
 *s\_name*="N/A"  
  
 if *s\_name*=="":  
 *s\_name*="N/A"  
 return *s\_name*

**Code 17 A class that returns name value from address value and city value**

The class that returns the name of the location is one of the most extensive classes used in the "GeoFinderUI" application. Its extensiveness results from several possible positions of the information sought on the page subjected to the technique of searching the source code, web scraping. The source code of the "Google Maps" pages does not have unified tags for the elements found on the search results page. This is problematic, so the program must refer to the location on the page using "X-Path", which is the path of sequential tags such as "<div>" or "<span>" counted from the beginning of the source code of the page.

def city\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*):  
 try:  
 *driver\_chrome*.find\_element(by=By.NAME, value='q').clear()  
 except Exception:  
 pass  
 *exception\_check* = 0  
  
 *search*.send\_keys(*name\_value*)  
 *search*.send\_keys(Keys.RETURN)  
  
 page\_to\_load(*driver\_chrome*, *delay*)  
 time.sleep(1)  
 try:  
 *s\_city* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[2]/div[1]/div[1]/h2[2]/span').text  
 *exception\_check* = 1  
 except NoSuchElementException:  
 *s\_city*="N/A"  
  
 if *s\_city*=="N/A" or *s\_city*=="":  
 try:  
 *s\_city* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Copy address"]').text  
 except NoSuchElementException:  
 *s\_city*="N/A"  
  
 if *s\_city*=="N/A" or *s\_city*=="":  
 try:  
 *s\_city* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Kopiuj adres"]').text  
 except NoSuchElementException:  
 *s\_city*="N/A"  
  
 try:  
 if *s\_city*!="N/A" or *s\_city*!="":  
 *temp\_city* = *s\_city* if *exception\_check* == 0:  
 *temp\_city* = *temp\_city*.split(sep=",")[1]  
 *s\_city* = *temp\_city*.split()[1]  
 except Exception:  
 *s\_city*="N/A"  
  
 if *s\_city*=="":  
 *s\_city*="N/A"  
 return *s\_city*

**Code 18 A class that returns city value from name value**

The class that assigns and returns the value of the "s\_city" variable obtained by entering the location name in the search field is mainly based on finding the full address of that location. In this case, finding this information on the page is easier, because next to each search result for a field having an address value, there is a shortcut for users to copy the full address to the clipboard in the device's memory. Due to the possible occurrence of the source code and the elements contained therein in Polish or English, the criteria for finding the appropriate item on the page are repeated, but having the word value "Copy address" in both languages. The next step is to split the received value of the full address and extract from it information about the city in which the location is located.

def zipcode\_f\_name(*driver\_chrome*, *search*, *delay*, *name\_value*):  
 try:  
 *driver\_chrome*.find\_element(by=By.NAME, value='q').clear()  
 except Exception:  
 pass  
 *exception\_check* = 0  
  
 *search*.send\_keys(*name\_value*)  
 *search*.send\_keys(Keys.RETURN)  
  
 page\_to\_load(*driver\_chrome*, *delay*)  
 time.sleep(1)  
 try:  
 *s\_zip\_code* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[2]/div[1]/div[1]/h2[2]/span').text  
 *exception\_check* = 1  
 except NoSuchElementException:  
 *s\_zip\_code*="N/A"  
  
 if *s\_zip\_code*=="N/A" or *s\_zip\_code*=="":  
 try:  
 *s\_zip\_code* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Copy address"]').text  
 except NoSuchElementException:  
 *s\_zip\_code*="N/A"  
  
 if *s\_zip\_code*=="N/A" or *s\_zip\_code*=="":  
 try:  
 *s\_zip\_code* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Kopiuj adres"]').text  
 except NoSuchElementException:  
 *s\_zip\_code*="N/A"

try:  
 if *s\_zip\_code*!="N/A" or *s\_zip\_code*!="":  
 *temp\_zip\_code* = *s\_zip\_code* if *exception\_check* == 0:  
 *temp\_zip\_code* = *temp\_zip\_code*.split(sep=",")[1]  
 *s\_zip\_code1* = *temp\_zip\_code*.split()[0]  
 *s\_zip\_code* = *s\_zip\_code1* except Exception:  
 *s\_zip\_code*="N/A"  
  
 if *s\_zip\_code*=="":  
 *s\_zip\_code*="N/A"  
 return *s\_zip\_code*

**Code 19 A class that returns zip-code value from name value**

Similar to the class that obtains the city in which a location is located from the name of that location, it uses the location's full address to then split it and assign under the "s\_zip\_code" variable a value corresponding to the location's zip code contained in the full address of the searched location.

def city\_f\_address(*driver\_chrome*, *search*, *delay*, *address\_value*):  
 try:  
 *driver\_chrome*.find\_element(by=By.NAME, value='q').clear()  
 except Exception:  
 pass  
  
 *search*.send\_keys(*address\_value*)  
 *search*.send\_keys(Keys.RETURN)  
  
 page\_to\_load(*driver\_chrome*, *delay*)  
 time.sleep(1)  
 try:  
 *s\_city* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[2]/div[1]/div[1]/div/div[2]/div[2]/div[1]/div/div/div/div[4]/div').text  
 except NoSuchElementException:  
 *s\_city*="N/A"  
  
 if *s\_city*=="N/A" or *s\_city*=="" or *s\_city*.upper() == *address\_value*.upper():  
 try:  
 *s\_city* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Copy address"]').text  
 except NoSuchElementException:  
 *s\_city*="N/A"  
  
 if *s\_city*=="N/A" or *s\_city*=="" or *s\_city*.upper() == *address\_value*.upper():  
 try:  
 *s\_city* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Kopiuj adres"]').text  
 except NoSuchElementException:  
 *s\_city*="N/A"

try:  
 if *s\_city*!="N/A" or *s\_city*!="":  
 *temp\_city* = *s\_city  
 s\_city* = *temp\_city*.split(",")[1]  
 *temp\_city* = *s\_city  
 s\_city* = *temp\_city*.split()[1]  
 except Exception:  
 *s\_city*="N/A"  
  
 if *s\_city*=="":  
 *s\_city*="N/A"  
  
 return *s\_city*

**Code 20 A class that returns city value from address value**

The class which obtains the name of the city in which a given location is located on the basis of its address, uses the same way of obtaining information as in previous classes, by full address of the location, and then splitting the text value and extracting from it the searched value, which in this case is the name of the city in which the location is located.

def zipcode\_f\_address(*driver\_chrome*, *search*, *delay*, *address\_value*):  
 try:  
 *driver\_chrome*.find\_element(by=By.NAME, value='q').clear()  
 except Exception:  
 pass  
  
 *search*.send\_keys(*address\_value*)  
 *search*.send\_keys(Keys.RETURN)  
  
 page\_to\_load(*driver\_chrome*, *delay*)  
 time.sleep(1)  
 try:  
 *s\_zip\_code* = *driver\_chrome*.find\_element(by=By.XPATH, value='//\*[@id="QA0Szd"]/div/div/div[1]/div[2]/div/div[1]/div/div/div[2]/div[1]/div[1]/div/div[2]/div[2]/div[1]/div/div/div/div[4]/div').text  
 except NoSuchElementException:  
 *s\_zip\_code*="N/A"  
  
 if *s\_zip\_code*=="N/A" or *s\_zip\_code*=="" or *s\_zip\_code*.upper() == *address\_value*.upper():  
 try:  
 *s\_zip\_code* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Copy address"]').text  
 except NoSuchElementException:  
 *s\_zip\_code*="N/A"  
  
 if *s\_zip\_code*=="N/A" or *s\_zip\_code*=="" or *s\_zip\_code*.upper() == *address\_value*.upper():  
 try:  
 *s\_zip\_code* = *driver\_chrome*.find\_element(by=By.CSS\_SELECTOR, value='[data-tooltip="Kopiuj adres"]').text  
 except NoSuchElementException:  
 *s\_zip\_code*="N/A"  
  
 try:  
 if *s\_zip\_code*!="N/A" or *s\_zip\_code*!="" or *s\_zip\_code*.upper() != *address\_value*.upper():  
 *temp\_zip\_code* = *s\_zip\_code  
 s\_zip\_code* = *temp\_zip\_code*.split(",")[1]  
 *temp\_zip\_code* = *s\_zip\_code  
 s\_zip\_code* = *temp\_zip\_code*.split()[0]  
 except Exception:  
 *s\_zip\_code*="N/A"  
  
 if *s\_zip\_code* == "" or *s\_zip\_code*.upper() == *address\_value*.upper():  
 *s\_zip\_code*="N/A"  
  
 return *s\_zip\_code*

**Code 21 A class that returns zip-code value from address value**

To get the postal code of a given location using the given address of that location, the program uses the same way as the previous described class. The way is to get the full address, and then split it and extract a piece of text corresponding to the postal code of the location.

# Summary

## Application outcome

The application "GeoFinderUI" is a program that returns to the user suggestive data about a given location, which were obtained from the minimum input data provided by the user. All search results are subject to the risk of incorrect data, this is due to the program finding the first, most matching search result to the input data entered. The idea behind this application is to reduce as much as possible the time needed to obtain geolocation data required in a defined business environment, which requires obtaining specific output data. The application was successfully formulated and written, thus achieving a fully functional program. Thanks to this project, I have achieved a much broader perspective on IT topics, mostly related to programming. My knowledge has been enriched with a lot of new information, skills and experience.

## Acknowledgments

The most interesting and rewarding part of the application development process was the achievement of the set goals. Thanks are due to Maksymilian Kubica, M.Sc. for his professional supervision of the project and for providing many valuable tips during the process of writing the thesis. The knowledge and experience he imparted are invaluable, and his management of the project allowed it to develop properly. For providing a place to work, help during brainstorming, and support, thanks are due to all those close to me - parents and friends.

# Bibliography

* *Aplikacja internetowa*. (2022, December). From Wikipedia: https://pl.wikipedia.org/wiki/Aplikacja\_internetowa
* *Bot program*. (2022, May). From Wikipedia: https://pl.wikipedia.org/wiki/Bot\_(program)
* *Code Refactoring*. (2023, January). From Refactoring: https://refactoring.com/
* *Discover Intellij Idea*. (2023, January). From jetbrains: https://www.jetbrains.com/help/idea/discover-intellij-idea.html
* Flower, M., Beck, K., Opdyke, W., & Brant, J. (2002). Refactoring Helps You Find Bugs. In M. Flower, *Refactoring: Improving the design of Existing Code* (pp. 48-49). Addison-Wesley.
* *Framework*. (2022, October). From Wikipedia: https://pl.wikipedia.org/wiki/Framework
* *Google Maps*. (2022, May). From https://www.google.com/maps/about/#!/
* *HTML*. (2023, January). From https://html.spec.whatwg.org/
* *Markup language*. (2023, January). From Wikipedia: https://en.wikipedia.org/wiki/Markup\_language
* *Materialize*. (2023, January). From https://materializecss.com/
* *Microsoft Excel*. (2023, January). From Wikipedia: https://en.wikipedia.org/wiki/Microsoft\_Excel
* *Openpyxl*. (2022, May). From https://openpyxl.readthedocs.io/en/stable/
* *Pandas software*. (2023, January). From Wikipedia: https://en.wikipedia.org/wiki/Pandas\_(software)
* *Python*. (2023, January). From https://docs.python.org/pl/3/
* *Selenium*. (2022, December). From https://www.selenium.dev/
* *Uniform Resource Locator*. (2022, December). From Wikipedia: https://pl.wikipedia.org/wiki/Uniform\_Resource\_Locator
* *Web scraping*. (2023, January). From Wikipedia: https://en.wikipedia.org/wiki/Web\_scraping

# Attachment

## Table of Figures

[**Figure 1 Presentation markup example** 8](#_Toc127208177)

[**Figure 2 Procedural markup example** 9](#_Toc127208178)

[**Figure 3 Descriptive markup example** 10](#_Toc127208179)

[**Figure 4 File Template.xlsx** 13](#_Toc127208180)

[**Figure 5 Input file with test data** 13](#_Toc127208181)

[**Figure 6 Output file with test data** 14](#_Toc127208182)

[**Figure 7 Google Maps search result** 23](#_Toc127208183)

[**Figure 8 GeoFinderUI application home page with Materialized styling packages**.......................................................................................................................24](#_Toc127208184)

[**Figure 9 the original diagrammatic process of the operation of the GeofinderUI application** 26](#_Toc127208185)

[**Figure 10 sample visualization of the GeoFinderUI start page** 27](#_Toc127208186)

[**Figure 11 data acquisition points by GeoFinderUI application** 27](#_Toc127208187)

[**Figure 12 Manual geo finder use case diagram** 28](#_Toc127208188)

[**Figure 13 GeoFinderUI use case diagram** 29](#_Toc127208189)

[**Figure 14 Information about the authors of materializecss.com** 34](#_Toc127208190)

## Table of Pictures

[**Picture 1 Left side, Martin Flower Picture 2 Right side, William Opdyke** 7](#_Toc126793112)

[**Picture 3 Left side, IntelliJ dark theme Picture 4 Right side, IntelliJ default light theme** 13](#_Toc126793113)

[**Picture 5 Left side, Nyan Progress Bar Picture 6 Right side, Snyk add-in** 14](#_Toc126793114)

[**Picture 7 Left side, Excel Worksheet Picture 8 Right side, Visual Basic for Applications** 15](#_Toc126793115)

[**Picture 9 Left Side, CWI building Picture 10 Right side, Guido van Rossum** 16](#_Toc126793116)

[**Picture 11 Jason Huggins** 17](#_Toc126793117)

[**Picture 12 Wes McKinney** 18](#_Toc126793118)

[**Picture 13 Left side, Eric Gazoni Picture 14 Right side, Charlie Clark** 19](#_Toc126793119)

[**Picture 15 Left side, Timothy Berners-Lee Picture 16 Right side, Robert Cailliau**.........................................................................................................................20](#_Toc126793120)

## Table of Code

[**Code 1 The <Head> section with the style part needed for positioning the content of the GeoFinderUI start page** 26](#_Toc126793275)

[**Code 2 Part of the code that allows the user to download the official version of the data template acceptable by the application** 27](#_Toc126793276)

[**Code 3 The form section needed for receiving and sending files and information to the back-end code** 28](#_Toc126793277)

[**Code 4 The part of the code responsible for the footer visible on each page of the "GeoFinderUI" web application** 29](#_Toc126793278)

[**Code 5 The part of the code that contains the information and its layout on the summary page** 30](#_Toc126793279)

[**Code 6 Part of the code containing the import of the necessary tools used in the "GeoFinderUI" application** 31](#_Toc126793280)

[**Code 7 Side methods in "home.py"** 32](#_Toc126793281)

[**Code 8 The beginning of the main method responsible for the functionality of the application** 32](#_Toc126793282)

[**Code 9 Part of the code responsible for preparing and cleaning the location for storing data files** 33](#_Toc126793283)

[**Code 10 Part of the code responsible for preparing the data file and launching the automated browser** 34](#_Toc126793284)

[**Code 11 A loop procedure that goes through all the records entered in the data file** 34](#_Toc126793285)

[**Code 12 Part of code with nested "if" statements that returns final data values** 36](#_Toc126793286)

[**Code 13 Last part of main class code** 37](#_Toc126793287)

[**Code 14 Class dedicated to accept cookies on website** 38](#_Toc126793288)

[**Code 15 A class that returns address value from name value** 39](#_Toc126793289)

[**Code 16 A class that returns city value from zip-code value** 39](#_Toc126793290)

[**Code 17 A class that returns name value from address value and city value** 41](#_Toc126793291)

[**Code 18 A class that returns city value from name value** 43](#_Toc126793292)

[**Code 19 A class that returns zip-code value from name value** 44](#_Toc126793293)

[**Code 20 A class that returns city value from address value** 45](#_Toc126793294)

[**Code 21 A class that returns zip-code value from address value** 47](#_Toc126793295)

## What CD contains

* **\Szymon\_Wujec\_s20431\_DiplomaThesis\_EPG**
  + Electronic version of engineering thesis
* **\GeoFinderUI**
  + All files necessary for the functioning of the web application