

**WYDZIAŁ INŻYNIERII METALI I INFORMATYKI PRZEMYSŁOWEJ**

**Praca dyplomowa**

**Opracowanie i Implementacja strony internetowej do obliczania i nauki wybranych metod numerycznych**

***Development and implementation of a website for calculating and learning selected numerical methods***

Autor: Szymon talar

Kierunek studiów: Inżynieria Obliczeniowa

Opiekun pracy: dr. Hab. Danuta Szeliga

Kraków, 2025

Uprzedzony o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.): „ Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystyczne wykonanie albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.”, a także uprzedzony o odpowiedzialności dyscyplinarnej na podstawie art. 211 ust. 1 ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (t.j. Dz. U. z 2012 r. poz. 572, z późn. zm.) „Za naruszenie przepisów obowiązujących w uczelni oraz za czyny uchybiające godności studenta student ponosi odpowiedzialność dyscyplinarną przed komisją dyscyplinarną albo przed sądem koleżeńskim samorządu studenckiego, zwanym dalej „sądem koleżeńskim”, oświadczam, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i że nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

………………………

**Spis treści**

1. Wstęp………………………………………………………………………………………..…4
2. Cel i założenia projektu…………………………………………………………………….….6
   1. Opis wykorzystanych metod ……………………………………………………………..6
   2. Omówienie dostępnych rozwiązań……………………………………………..................8
   3. Wykorzystane technologie……...………………………………………………………..14
3. Implementacja………………………………………………………………………………..17
   1. Napisanie i optymalizacja algorytmów………………………………………………….17
   2. Komunikacja front-back…………….…………………………………………….…..…21
   3. Stworzenie GUI………………………………………………………………………….24
   4. Dodanie sekcji pomocniczych …………………………………………………………..25
4. Testy aplikacji………………………………………………………………………………...30
5. Podsumowanie………………………………………………………………………………..33
6. Źródła…………………………………………………………………………….…………...34
7. **Wstęp**

Rozwój technologii informacyjnych i dynamiczny postęp w dziedzinie programowania otwierają nowe możliwości w nauce oraz dydaktyce. Współczesne systemy edukacyjne coraz częściej wykorzystują narzędzia internetowe, które nie tylko ułatwiają dostęp do wiedzy, ale także wspierają aktywny proces nauczania i uczenia się. Jednym z obszarów, który może szczególnie skorzystać z tego rodzaju rozwiązań, są metody numeryczne – dziedzina matematyki stosowanej, znajdująca szerokie zastosowanie w inżynierii, fizyce, ekonomii oraz informatyce. Metody numeryczne pozwalają na rozwiązywanie złożonych problemów matematycznych, których analityczne rozwiązanie jest trudne lub wręcz niemożliwe.

Celem niniejszej pracy inżynierskiej jest opracowanie i implementacja strony internetowej, która będzie pełniła dwie zasadnicze funkcje: edukacyjną oraz narzędziową. Platforma ta umożliwi użytkownikom zarówno zrozumienie teoretycznych podstaw wybranych metod numerycznych, jak i ich praktyczne zastosowanie poprzez interaktywne narzędzia do obliczeń. Z racji na funkcjonalności oferowane przez tę stronę, może być ona wykorzystywana przez osoby, które zarówno dopiero zaczynają swoją naukę metod numerycznych i przy wykorzystaniu dostępnych zawartości pragną lepiej zrozumieć ich działanie, jak i również osoby już zaznajomione z tymi metodami i potrafiące z nich korzystać, które to jednakże napotkały pewne problemy w trakcie wykonywania obliczeń i chcące przyspieszyć ten proces.

Strona internetowa zostanie zaprojektowana z wykorzystaniem nowoczesnych technologii webowych, takich jak HTML5, CSS, Angular oraz Java, wraz z wybranymi frameworkami i bibliotekami wspomagającymi proces programowania. Implementacja obejmie moduły umożliwiające prezentację teorii, interaktywne wizualizacje oraz automatyczne obliczenia numeryczne. W ramach projektu zostanie zaimplementowanych kilka przykładowych algorytmów, które taka strona może zawierać. Wybrany na potrzeby projektu zestaw ma wyłącznie za zadanie pokazanie możliwości oraz potencjału drzemiącego w tym projekcie, natomiast sam projekt został wykonany w sposób umożliwiający proste dodawanie kolejnych metod i algorytmów tak, aby strona zawierała kompleksowe omówienie i wykorzystanie tych najczęściej używanych w praktyce metod. Użytkownicy będą mieli możliwość krok po kroku prześledzić działanie tych algorytmów oraz eksperymentować z danymi wejściowymi w celu lepszego zrozumienia ich zasad działania.

Niniejsza praca jest odpowiedzią na rosnące zapotrzebowanie na narzędzia wspierające edukację matematyczną i inżynierską w sposób nowoczesny i przyjazny dla użytkownika. Pomimo istnienia licznych opracowań i aplikacji w tym zakresie, wiele z nich jest trudno dostępnych, ma ograniczoną funkcjonalność lub nie oferuje wystarczająco wysokiej jakości treści edukacyjnych. Celem pracy jest zatem nie tylko stworzenie działającego narzędzia, ale także dostarczenie produktu, który łączy funkcjonalność z wartością dydaktyczną, odpowiadając na potrzeby współczesnego użytkownika i jednocześnie przy tym nie pobierając żadnych opłat od użytkowników.

W kolejnych rozdziałach pracy zostaną przedstawione szczegółowe założenia projektu, wybór technologii, proces implementacji oraz analiza wyników. Podjęta zostanie także refleksja nad możliwościami rozwoju stworzonej platformy oraz jej potencjalnym wkładem w edukację matematyczną i naukę metod numerycznych. Tym samym, niniejsza praca wpisuje się w nurt wykorzystania nowoczesnych technologii w edukacji oraz promowania nauki przez praktykę i doświadczenie.

1. **Cel i założenia projektu**
2. **Opis wykorzystanych metod**

Projekt zakłada wykonanie strony internetowej, stanowiącej pomoc w rozwiązywaniu i analizie metod numerycznych. Będzie się składała z 5 sekcji:

1. **Menu główne**

To będzie główna część strony, i zarazem pierwsza którą użytkownik zobaczy zaraz po wejściu na nią. Z tego ekranu użytkownik będzie miał możliwość: zarejestrować swoje konto na stronie, zalogować się podając dane do już istniejącego konta, bądź też wejść na stronę jako gość i przejść od razu do ekranu z metodami. Celem tego ekranu oraz możliwości logowania się na stronę jest danie użytkownikowi swobody w zakresie, z jakiej będzie chciał z niej korzystać: dla zalogowanego użytkownika, podczas wykonywania obliczeń dane podawane w postaci parametrów będą zapisywane w bazie danych. Pozwoli to później na stworzenie historii wyników, która będzie dostępna jako osobna zakładka na stronie, po kliknięciu której użytkownikowi zostaną wyświetlone wcześniejsze operacje co pozwoli mu np. wrócić do poprzednich działań gdyby tego potrzebował bez konieczności ponownego wprowadzania tych samych danych.

1. **Metody**

W tym miejscu będzie dostępny podgląd wszystkich dostępnych metod do wykorzystania. Każda metoda po kliknięciu na nią, rozwinie podmenu zawierające wszystkie przewidziane sposoby na rozwiązanie tej metody, a z kolei każdy ze sposobów, po naciśnięciu wyświetli okienko z danymi potrzebnymi do wprowadzenia aby wykonać obliczenia. To okno będzie się zmieniać w zależności od wybranej metody, tak, aby zawrzeć wszystkie potrzebne dane. Ponadto, Z menu będzie również możliwość przeniesienia się do innych sekcji znajdujących się na stronie przy pomocy menu, widocznego na górze oraz stopce strony. Część z linków będzie dostępna tylko dla zalogowanych użytkowników, natomiast te które nie będą wymagały konkretnego konta użytkownika będą dostępne dla wszystkich.

1. **Teoria**

Na tym ekranie dostępne będzie teoretyczne omówienie metod dostępnych na stronie.

Oprócz krótkiego opisu teoretycznego, będzie zawierała jakąś grafikę, bądź też krótki opis wzoru pomagające lepiej zobrazować sobie daną metodę. Pod każdym opisem będzie znajdował się odnośnik/odnośniki do innych stron zewnętrznych, opisujących dane zagadnienie w bardziej szczegółowy sposób w przypadku gdyby któtka informacja na stronie okazała się dla użytkownika niewystarczająca.

1. **Wyniki**

Ta podstrona będzie wyświetlała rezultaty obliczeń wcześniej zaznaczonej metody. W zależności od tego, z jakiej skorzystamy, strona będzie dostosowywała wyświetlaną zawartość. Wśród wyświetlanej zawartości, zawsze będą znajdowały się wszystkie dane które zostały wprowadzone do programu, a także obliczony wynik algorytmu. Ponadto, jeśli to będzie możliwe, zostanie zaprezentowany wykres obrazujący w lepszy sposób dane oraz wynik tak, aby użytkownik mógł łatwiej zrozumieć działanie danej metody. Będzie też możliwość z tego miejsca wybrać inną metodę rozwiązania tego samego problemu, a następnie porównania rezultatów dwóch metod w celu analizy różnic pomiędzy nimi, dzięki czemu użytkownik będzie mógł w prostszy sposób przeprowadzić analizę różnych sposobów na obliczenie tego samego zadania.

1. **Historia wyników**

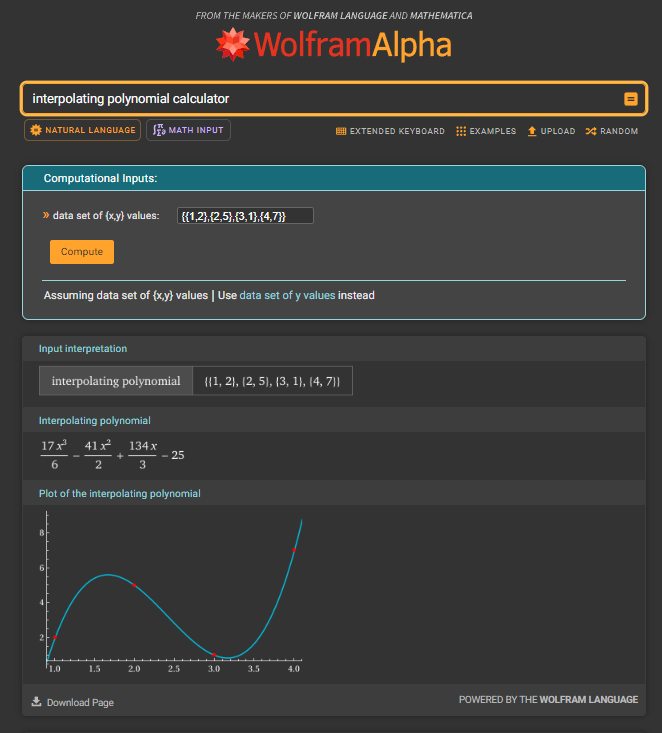
Tam będzie znajdowała się lista ostatnich obliczeń wykonywanych przez użytkownika. Dzięki temu, użytkownik będzie miał możliwość ponownej analizy konkretnych przykładów algorytmów, z których najczęściej korzysta wraz z już podstawionymi wcześniej wartościami. Ta lista będzie ograniczona co do ilości wyświetlanych rekordów, natomiast wszystkie starsze wyszukania będą systematycznie usuwane. Pozwoli nam to uniknąć potencjalnych problemów z wielkością przechowywanych danych od wszystkich osób w dłuższej perspektywie. Po wybraniu danego zapisu, strona przekieruje nas ponownie na ekran wyników, tak, jakbyśmy wybrali tę metodę i dane z menu głównego.

Stworzenie tych komponentów od strony wizualnej wraz z prostą szatą graficzną, napisanie algorytmów koniecznych do wykonywania wszystkich uwzględnionych obliczeń wraz z przekazaniem niezbędnych informacji na temat wyników, weryfikacja poprawnego działania algorytmów poprzez stworzenie testów jednostkowych dla każdej z metod z osobna oraz własnoręczne przejście ścieżki procesu programu, uwzględnienie pozostałych funkcjonalności mający pomóc użytkownikom w nauce i połączenie tych segmentów w spójną całość będzie stanowić cel niniejszej pracy.

1. **Omówienie dostępnych rozwiązań**

W tym podrozdziale przyjrzymy się obecnie istniejącym kilku najpopularniejszym rozwiązaniom, pozwalającym w jakiś sposób nauczyć się bądź też obliczyć wybrane metody numeryczne. Będę tutaj uwzględniał wyłącznie rozwiązania, które są dostępne w wersjach darmowych oraz te, które są szeroko dostępne i nie wymagają żadnych dodatkowych kroków koniecznych do skorzystania z nich jak np. wymagane założenie konta. Pominę tutaj również aplikacje do pobrania na komputer bądź telefon, ponieważ z racji na swoją specyfikację będą one borykać się innymi problemami nieznanymi stronom internetowym(m.in. utrudniony dostęp z racji na sam fakt wymogu pobrania aplikacji czy też uzależnienie wydajności i dostępności kalkulatora od urządzenia posiadacza oraz rodzaju posiadanego oprogramowania). Wszystkie poniższe przykłady będę omawiał na podstawie próby wykonania interpolacji, w celu zachowania przejrzystości i spójności przeglądu.

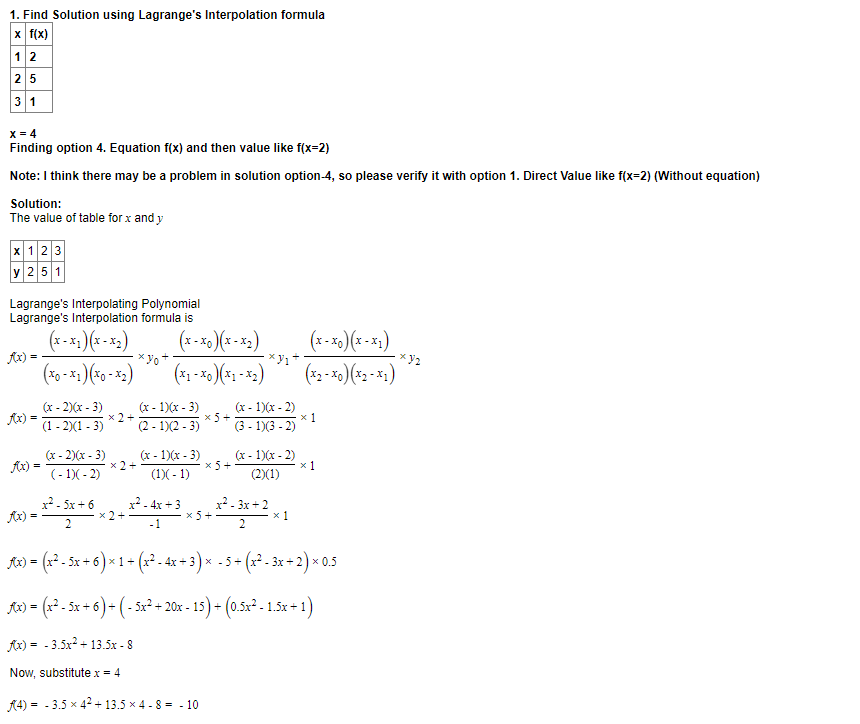
1. **Wolfram Alpha**

Obecnie najpopularniejszy kalkulator naukowy, oferuje szeroki wachlarz możliwości zarówno w kontekście metod numerycznych jak i innych pomocy naukowych. Po wejściu do menu głównego, mamy możliwość wpisania w pasku wyszukiwania interesującego nas zagadnienia.  
  


Rys.1. Zrzut ekranu przedstawiający wynik przykładowej interpolacji wielomianowej wykonanej za pomocą Wolframu Alpha[[1]](#footnote-1)

Jak widać na Rys.1, Wolfram umożliwia nam wykonanie interpolacji wielomianowej. Po podaniu danych w postaci listy punktów, otrzymujemy wyliczony wzór funkcji oraz fragment jej wykresu wraz z zaznaczonymi punktami które zostały podane w danych wejściowych. Już na tym etapie można zauważyć jednak kilka wad zaprezentowanego rozwiązania: lista punktów, które chcemy poddać interpolacji musi zostać podana w postaci dwóch wartości zamkniętych klamrami, co zważywszy na wstawienie małego okienka bez możliwości jego powiększenia, może powodować problemy z czytelnością danych wejściowych, zwłaszcza przy ich większej ilości. Nie mamy również możliwości obliczenia wartości dla konkretnego punktu, a jedynie wzór pozwalający nam na zrobienie tego. Takie rozwiązanie wciąż pozostawia użytkownikowi do wykonania część obliczeń, w postaci podstawienia pożądanej wartości do pokazanego wzoru. Ponadto, brakuje tutaj również opcji porównującej dwie różne metody wykonania takiej interpolacji, co pozwoliłoby użytkownikowi porównać otrzymane rezultaty, aby wyciągnąć wnioski na ich podstawie i stwierdzić, która jest odpowiedniejsza do zadanego przez niego przypadku.

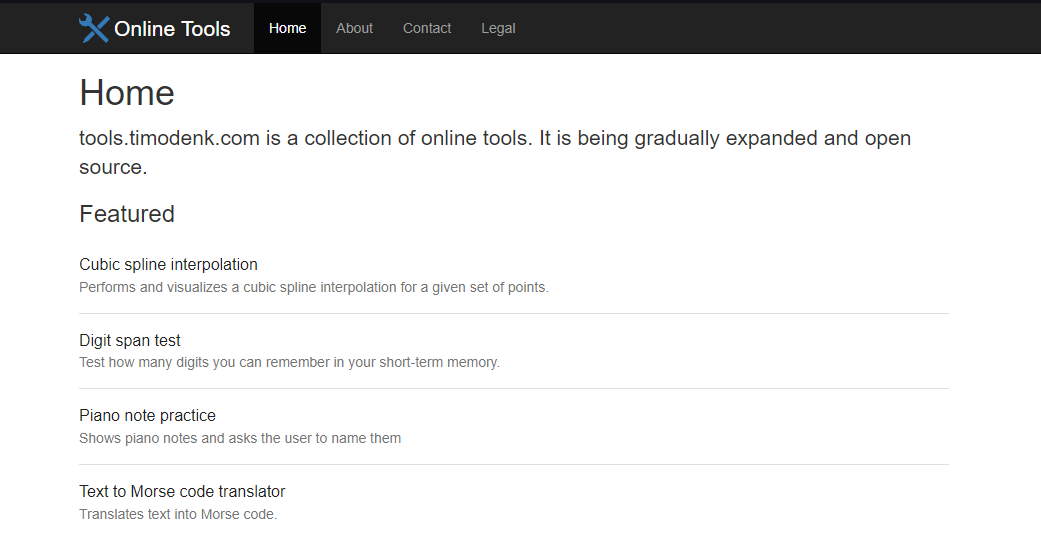
1. **AtoZmath**

Strona specjalizująca się rozwiązywaniu wszelakich problemów obliczeniowych. Tak jak w przypadku Wolframu, tutaj również mamy możliwość wyszukania interesującej nas metody w celu wykonania obliczeń z jej pomocą. Po wejściu na podstronę przeznaczoną obliczaniu interpolacji Lagrange’a mamy możliwość podania ilości punktów, ich umiejscowienia na osi oraz typ rozwiązania.  
  


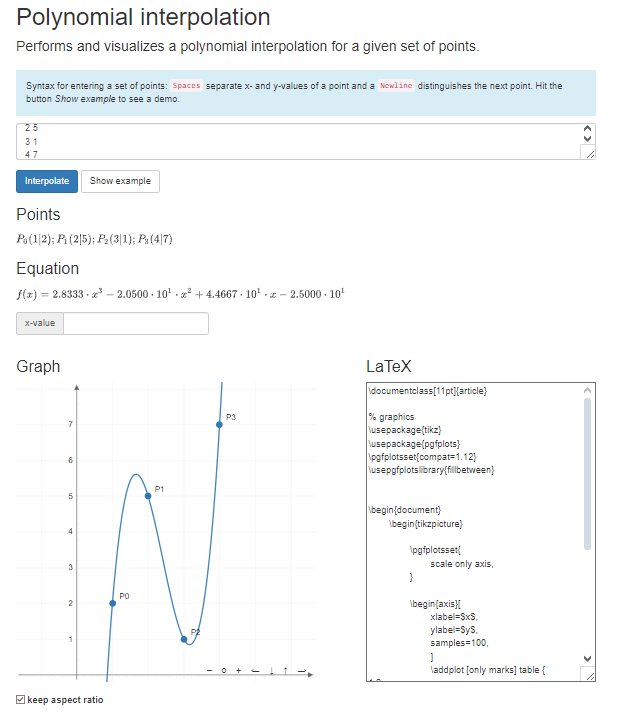
Rys.2. Zrzut ekranu przedstawiający wynik przykładowej interpolacji wielomianowej wykonanej za pomocą AtoZmath[[2]](#footnote-2)

Na Rys.2 widać rezultat takiego przykładowego zapytania o rozwiązanie dla trzech punktów. Pierwsze co rzuca się w oczy to skrupulatne rozpisanie algorytmu wykonującego obliczenia, które to złożyły się na uzyskanie zaprezentowanego rezultatu. Ponadto, pochwalić można również czytelność strony: w przeciwieństwie do rozwiązania dostępnego w Wolframie, tutaj zarówno wpisywanie danych wejściowych jak i odczytywanie wyniku jest znacznie łatwiejsze i przystępniejsze, zwłaszcza dla mniej doświadczonych użytkowników. Jednakże strona ta również ma pewne wady: Wyskakujące zewsząd reklamy utrudniają poruszanie się w jej obrębie, a do uzyskania pełnego rozwiązania należy odczekać kilkanaście sekund dla pojedynczego przypadku, a czas ten dodatkowo zwiększa się z każdym kolejnym obliczeniem w krótkim odstępie czasu, przez co systematyczne korzystanie jest czasochłonne w porównaniu do innych rozwiązań. Dodatkowo, nawet pełne rozwiązanie pozbawione jest wizualnego zobrazowania wyniku oraz opcji porównania z alternatywną metodą obliczeniową.

1. **Online Tools**

Jak sama nazwa wskazuje, jest to witryna obejmujące wiele różnorakich narzędzi online, zarówno z zakresu metod obliczeniowych, jak i tych w żaden sposób nie związanych z matematyką jak np. ćwiczenia z zapamiętywania notacji muzycznej bądź też konwersje dowolnego znaku na jej kod UTF-16. Pomimo iż nie jest to strona oferująca tak szeroki wachlarz możliwości, została tutaj umieszczona jako prezentacja możliwości trochę mniej znanych i rozbudowywanych stron.   


Rys.3. Zrzut ekranu przedstawiający stronę główną witryny Online Tools[[3]](#footnote-3)

Pierwszy problem jak napotykamy po wejściu na samą stronę zaczyna się jeszcze przed próbą rozwiązania jakiegokolwiek problemu, ponieważ strona główna widoczna na Rys.3 nie pozwala na wyszukanie w żaden sposób funkcjonalności, z której chcielibyśmy skorzystać. W przypadku liczby funkcjonalności takiej, jaką oferuje ta strona, brak takiej możliwości już na początku utrudnia użytkownikowi nawigację po niej. Gdy już znajdziemy stronę odpowiadającą za interpolację wielomianową, możemy wpisać punkty w polu tekstowym, przy czym każdy punkt musi być podany jako dwie wartości oddzielone spacją, a każdy punkt zaczynać się od nowej linii.  
  
  


Rys.4. Zrzut ekranu przedstawiający stronę wynikową Online Tools po rozwiązaniu interpolacji wielomianowej

Ekran wynikowy zaprezentowany na Rys.4 składa się ze wzoru funkcji, obliczonej na podstawie podanych punktów, oraz wykresu tej funkcji. Ponadto mamy możliwość swobodnego poruszania się po wygenerowanym wykresie, a także otrzymaliśmy kod w LaTeXie który możemy zapisać w celu ponownego odtworzenia tej funkcji. Niestety, analogicznie do poprzednich rozwiązań, brakuje tutaj opcji porównującej wyniki z innymi metodami, jak i brakuje tutaj możliwości obliczenia wartości funkcji w podanym przez użytkownika miejscu.

Nie widzę potrzeby omawiać osobno pozostałych rozwiązań dostępnych w internecie, ponieważ dostarczają one możliwości zbliżone do tych już omówionych na poszczególnych przykładach. Podsumowując, wśród najczęściej występujących znajdują się: brak możliwości obliczenia interpolacji dla konkretnego miejsca na wykresie, brak możliwości porównania wyników z konkurencyjnymi metodami, brak możliwości zapisu wyników w celu późniejszego odtworzenia lub analizy, oraz wysoka czasochłonność korzystania ze stron wynikająca z blokad twórców albo niskiej czytelności interfejsu. Ten projekt będzie skupiał się na rozwiązaniu wyżej wymienionych problemów.

1. **Wykorzystane technologie**

Technologie wykorzystywane do wykonania tego projektu można podzielić na trzy kategorie:

1. **Front-end**

W tej grupie technologii zawierają się te podstawowe, niemalże niezbędne do utworzenia jakiejkolwiek strony internetowej, tj. HTML oraz CSS, natomiast całość programistyczną stworzyłem za pomocą jednego z najpopularniejszych frameworków dla JavaScripta, czyli Angularze. Wybór właśnie takich narzędzi do pracy został podyktowany szerokim wachlarzem możliwości oferowanym przez ten właśnie framework: pozwala on zarówno na stworzenie pełnych stron od podstaw, co daje twórcy bardzo duży wpływ na jej finalny wygląd oraz komunikację pomiędzy poszczególnymi segmentami, natomiast z drugiej strony pozwala zachować przy tym przejrzystość w kreowanym projekcie i zapewnia wsparcie dla nowoczesnych rozwiązań, pozwalających w o wiele prostszy sposób wykonać pewne zabiegi na stronie, które to byłyby znacznie trudniejsze do wykonania posługując się zwykłym JavaScriptem.

1. **Back-end**

W tym miejscu zdecydowałem się skorzystać z Javy. W porównaniu z jej głównym konkurentem w tej kategorii, tzn. C#, oferuje ona o wiele większą stabilność w wydajności dla różnych systemów operacyjnych na których będzie uruchamiana, co z kolei powinno wpłynąć na zwiększoną liczbę potencjalnych użytkowników takiej strony, a także zapewnia łatwiejsze zarządzanie projektami tworzonymi na szeroką skalę, zawierających w sobie wiele współpracujących ze sobą podzespołów. Ewentualne wady takiego rozwiązania, wśród których głównym z nich z pewnością może być mniejsza wydajność Javy od C#, mogą zostać zniwelowane poprzez zarówno optymalne tworzenie algorytmów obliczeniowych, jak i w przypadku nadal występujących problemów z wydajnością, zastosowaniem wielowątkowości.[[4]](#footnote-4) [[5]](#footnote-5)

1. **Bazy danych**

Dla zapisania danych rejestrujących się użytkowników oraz danych wejściowych obliczeń przez nich wykonywanych, wykorzystuję w projekcie dwie osobne bazy danych, kolejno: H2 oraz MongoDB. Pierwsza z nich, choć nie jest obecnie powszechnie wykorzystywana jako pełnoprawna baza danych i pełni głównie funkcje testowe, uważam że dla tego przypadku jest odpowiednim wyborem ze względu na:

- Niski wkład pracy potrzebny w utworzenie oraz zarządzanie taką bazą danych

- Niewielki rozmiar tabeli oraz przewidywana liczba rekordów, dzięki czemu kwestie wydajności nie są dla tego zastosowania zbyt istotne

- Projekt nie zostaje na chwilę obecną wdrażany do powszechnego użytku, dzięki czemu nie ma konieczności przykładania tak dużej troski o względy bezpieczeństwa, jak dla przypadku w którym baza ta miałaby funkcjonować na środowisku produkcyjnym.   
W przypadku gdyby któraś z tych zależności uległa zmianie w trakcie rozbudowywania projektu, da się również w dość prosty sposób zamienić tę bazę na inną, stworzoną w bardziej sprawdzonym środowisku jak np. MySQL.

Drugie zastosowanie, czyli zapis danych wejściowych niestety już nie jest możliwy do obsługi tą bazą danych w tak prosty sposób, przez co musiałem zastosować osobny byt w postaci bazy MongoDB. Jej wybór z kolei został podyktowany długą historią powszechnego użytku, co zapewnia wsparcie dla wielkoskalowych projektów, oraz ze względu na swoją specyfikę pozwala na łatwe zapisywanie znacznie różniących się od siebie rekordów w jednej tabeli, co będzie dla tego projektu kluczowe ze względu na mnogość potencjalnych metod wybranych przez użytkowników. W tym względzie jest to wybór nieporównywalnie lepszy od relacyjnych baz danych takich jak MySQL. Jak możemy przeczytać na stronie samych twórców: *„The schemaless design of MongoDB documents makes it extremely easy to build and enhance applications over time, without needing to run complex and expensive schema migration processes as you would with a relational database.”[[6]](#footnote-6)* [pl. Bezschematowa konstrukcja dokumentów w MongoDB sprawia, że niezwykle łatwo jest budować i rozwijać aplikacje w miarę upływu czasu, bez konieczności przeprowadzania skomplikowanych i kosztownych procesów migracji schematów, jak miałoby to miejsce w przypadku relacyjnej bazy danych.], co idealnie powinno się sprawdzić dla przypadku, w którym do bazy danych z danymi obliczeń w miarę upływu czasu mogłyby dochodzić kolejne metody obliczeniowe, co oznaczałoby jednocześnie kolejne, różne od wszystkich poprzednich, struktury rekordów do zapisania w tabeli.

1. **Implementacja**
2. **Napisanie i implementacja algorytmów**

Na potrzebny projektu stworzyłem algorytmy pozwalające obliczyć trzy różne zagadnienia: interpolację, całkowanie oraz układ równań. Dla dwóch pierwszych dodałem kilka różnych sposobów na ich obliczenie, aby dać użytkownikom możliwość porównania skuteczności każdej z metod. Dla interpolacji były to metody: wielomianowa, trygonometryczna oraz funkcjami sklejanymi. Dla całkowania były to metody: punktu środkowego, trapezów oraz Simpsona. Łącznie daje to siedem różnych algorytmów, zaimplementowanych w celu demonstracji możliwości, jakie daje strona.



Fragment kodu 1. Implementacja modelu punktu. Źródło: opracowanie własne

Fragment Kodu 2. Implementacja modelu odpowiadającego danym wejściowym dla interpolacji.

Źródło: opracowanie własne

****Fragment Kodu 3. Implementacja funkcji służącej do obliczania wartości interpolacji trygonometrycznej.

Źródło: opracowanie własne

W załączonych Fragmentach Kodu 1, 2, 3 zamieściłem przykładowy algorytm zastosowany w projekcie. W tym przypadku, jako że próbujemy obliczyć wynik interpolacji trygonometrycznej, oprócz zmiennej *searchedX* odpowiadającej za wartość *x* dla której próbujemy obliczyć wartość, potrzebujemy także zmienić skalę w której znajdują się nasze punkty w taki sposób, aby wszystkie podane przez użytkownika wartości zawierały się w przedziale [0, π)[[7]](#footnote-7). Aby tego dokonać, dla każdego podanego punktu potrzebujemy rozwiązać równanie podane we Wzorze (1)

Wzór 1. Równanie przekształcające dowolną podaną wartość *x* na jej odpowiednik w zbiorze [0;2π)

Gdzie:

*normalizedX* – wartość po dokonaniu przekształcenia

*minX* – dolna granica zbioru określonego na podstawie podanych w danych wejściowych punków

*rangeX* – różnica pomiędzy górną i dolną granicą zbioru wejściowego

*x* – wartość wejściowa, podana przez użytkownika

Pozostała część kodu, skupia się na obliczeniu *szeregu Fouriera*, dzięki której mamy możliwość przedstawić funkcję okresową w postaci sumy funkcji trygonometrycznych. W tym przypadku, funkcja ta będzie przyjmowała postać zawartą we Wzorze (2)

Wzór 2. *Szereg Fouriera*, pozwalający na obliczenie wartości interpolacji dla dowolnego podanego *x*

Gdzie:

*F(x)*- wartość funkcji w punkcie *x*, gdzie w tym przypadku będzie on przyjmował wartość podaną przez użytkownika jako wartość do obliczenia

, , – współczynniki wyliczane na podstawie znanych dotychczas punktów

*N* – ilość podanych punktów

Wartość zwracana przez tę funkcję, tj. pojedyńcza zmienna typu *double* przedstawiająca wartość funkcji w konkretnym punkcie, jest następnie przekazywana do front-endu który wyświetla ją na ekranie z pozostałymi danymi. W analogiczny sposób zostały napisane pozostałe algorytmy obliczające inne metody bądź wykorzystujące inne sposoby realizacji obliczeń, tak samo jak ich obsługa w pozostałych częściach kodu.

1. **Komunikacja front-back**

Ten segment będzie zawierał opis procesu, w którym to dane otrzymane w wyniku wywołania kalkulatora są przekazywane na ekran widokowy.



Fragment Kodu 4. Część kodu serwisu odpowiadającego za komunikację pomiędzy kontrolerem a kalkulatorem. Źródło: opracowanie własne



Fragment Kodu 5. Część kodu kontrolera a Javie odpowiedzialna za obsługę danych dot. Interpolacji trygonometrycznej, i przekazanie rezultatu do widoku Żródło: opracowanie własne

We Fragmentach Kodu 4 oraz 5 widzimy, jak dane są odbierane i przekazywane z powrotem do kontrolera obługującego dany typ obliczeń. W tym przypadku, otrzymujemy dane w postaci z Fragmentu Kodu 2 na adres kończący się ciągiem znaków *„/trigonometric\_interpolation”*, wywołujemy kalkulator obliczający wybrany przez nas problem, a następnie w odpowiedzi zwracamy obiekt *ResponseEntity* z wartością wynikową.



Fragment Kodu 6. Klasa odpowiadająca za zapewnienie możliwości komunikacji do części napisanej w Angularze. Źródło: opracowanie własne

Kluczowe dla komunikacji jest również zapewnienie łączności pomiędzy programami napisanymi w Javie oraz Angularze. Sama budowa projektu wymusza na nas zastosowanie różnych portów dla obu tych części, tak więc w przypadku gdy chcemy przekazać dane z Angulara do Javy bądź też odwrotnie, potrzebujemy w obu przypadkach skonfigurować projekt w sposób umożliwiający taką komunikację. Dla Javy, taką konfigurację zapewnia klasa *CorsConfig* zamieszczona we Fragmencie Kodu 6., w której to nadpisana jest metoda *addCorsMappings*. Możemy prześledzić zmiany przez nią dokonane krok po kroku:  
- Przede wszystkim, potrzebujemy uzyskać dostęp do obiektu *CorsRegistry*, w którym przechowywane są m.in. ustawienia dotyczące zasad jakimi kieruje się program komunikując się ze wszystkimi innymi punktami końcowymi (ang. *endpoints*).

- Pierwsza linia (*addMapping(„/\*\*”)*, dodaje wymienione ustawienia na cały projekt, tj. dla wszystkich występujących w nim punktów końcowych  
- *.allowedOrigins(…)* pozwala na zdefiniowanie zewnętrznego adresu, z którym będziemy mogli się komunikować. Dla nas, adres pozostaje taki sam, gdyż cały projekt jest umieszczony na serwerze lokalnym(ang. *localhost)*, natomiast potrzebujemy dostępu konkretnie do portu 4200, na którym uruchamia się część napisana w Angularze

- *.allowedMethods(…)* pozwala zdefiniować nam metody, którymi będziemy mogli się komunikować z podanymi punktami końcowymi. Wymienione zostały tutaj cztery najczęściej wykorzystywane, choć w praktyce zdecydowana większość wykorzystywana w projekcie jest metodą *POST*, i to ona odgrywa dla nas najważniejszą rolę. Z jej pomocą możemy przesyłać dane pomiędzy punktami, jak np. dane dla algorytmów, a także tworzyć i modyfikować istniejące już zasoby np. modyfikować dane w bazach danych.

- *.allowedHeaders* pozwala na dołączanie dowolnych nagłówków wraz z resztą informacji

- *.allowedCredentials(true)* pozwala na stosowanie autentykacji w kodzie, co może się okazać przydatne gdybyśmy potrzebowali sprawdzić w jakimś miejscu kodu czy użytkownik jest zalogowany na swoim koncie, bądź też korzysta ze strony jako gość.



Fragment Kodu 7. Zawartość pliku *proxy.conf.json* zawierający konfigurację łączności dla Angulara

Źródło: opracowanie własne



Fragment Kodu 8. Wysyłanie danych we front-endzie na opisany w Fragmencie Kodu 5. Kontroler

Źródło: opracowanie własne

We Fragmentach Kodu 7 i 8 została pokazana perspektywa ze strony front-endu. Podobnie jak w Javie, tutaj również potrzebne jest zdefiniowanie ip oraz portu punktów docelowych, z którymi będziemy chcieli w przyszłości się łączyć w celu wymiany danych. W tym przypadku taką funkcję pełni plik *proxy.conf.json*, który zawiera podstawowe informacje na temat przyszłych połączeń. Możemy z niego odczytać:  
- obszar działania wybranych ustawień, przy pomocy obiektu *„/api/\*”*, co oznacza wybranie ustawień dla całego projektu

- ip i docelowy port z którym umożliwiamy komunikację, czyli *localhost:8081*

- flagę odpowiadającę za włączenie zabezpieczeń występujących w kodzie

- poziom na którym będą wyświetlane wszystkie logi programu.  
  
Dzięki zastosowaniu tych ustawień, mamy możliwość w dowolnym miejscu w kodzie zastosować funkcję *this.http.post<>(…)* która wykonuje połączenie. W nawiasach klamrowych podajemy typ zmiennej oczekiwanej w rezultacie wykonania połączenia, natomiast nawiasy okrągłe zawierają w sobie zmienne dotyczące dokładnego punktu końcowego, z którym chcemy się połączyć, oraz dane wejściowe do przekazania do algorytmu. W przypadku sukcesu, funkcja zapisuje zmienne otrzymane w wyniku działania algorytmu i przekierowuje nas na ekran wynikowy, natomiast w momencie wystąpienia jakiegoś błędu, jego treść jest wypisywana na ekranie.

1. **Stworzenie GUI**

*GUI* (ang. *Graphical User Interface*) jest formą prezentacji danych i zestawem narzędzi graficznych umożliwiających użytkownikowi na wchodzenie w interakcję z danym programem. W wykonanym przeze mnie projekcie przybiera on formę wszystkich ekranów dostępnych na stronie, wraz z dobranymi do niej stylami mającym za zadanie ułatwienie poruszanie się po jej poszczególnych ekranach. Ideą, która mi przyświecała przy wykonywaniu elementów graficznych dla tego projektu była prostota interfejsu graficznego, pozwalająca na proste lokalizowanie kluczowych dla użytkownika miejsc na stronie, a także czytelność znajdującej się tam zawartości. Każda strona znajdująca się w projekcie składa się z trzech części:

*-* paska górnego, na którym znajduje się nazwa projektu (w przypadku wydania tej strony do użytku publicznego byłoby to miejsce na umieszczenie jej nazwy i/lub logo) oraz pasek nawigacji, który umożliwia przełączanie się pomiędzy poszczególnymi kartami. Na Rys. 5 zaprezentowany został wraz z ekranem wyszczególniającym dostepne metody do użytku. W momencie jego tworzenia użytkownik nie był zalogowany na stronie i korzystał z niej jako gość, więc część podstron została dla niego zablokowana.

*-* sekcji głównej, której zawartość będzie się zmieniała w zależności od podstrony na którą wejdzie użytkownik. Niezależnie od jego położenia, zapisane style zapewniają ustawienie minimalnej wysokości tak, aby strona obejmowała cały ekran, oraz domyślne tło.

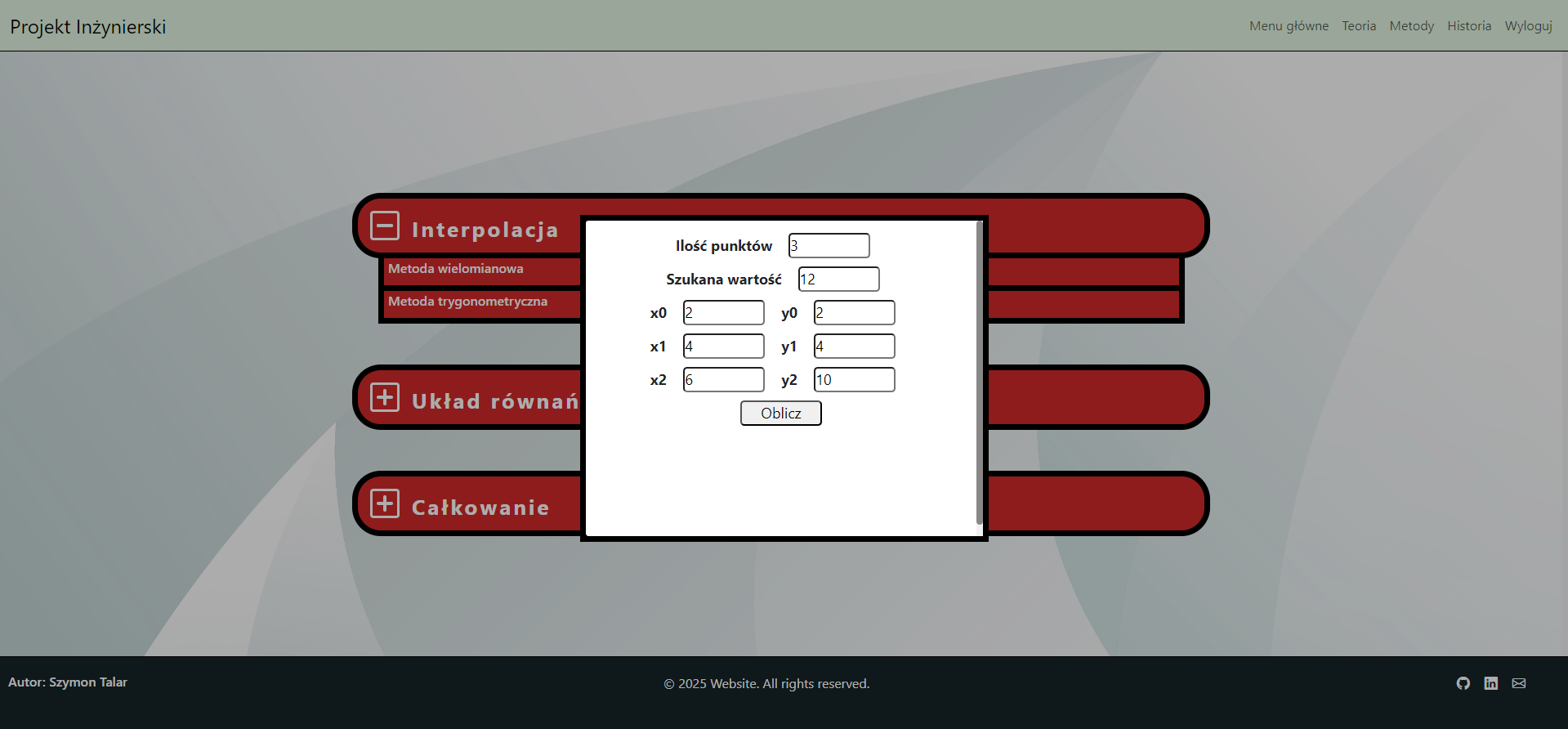
*-* stopki, na której zdecydowałem się umieścić możliwości kontaktu oraz krótki napis o zastrzeżeniu praw autorskich. W przypadku umieszczenia strony do użytku publicznego, ta sekcja mogłaby również zawierać informacje nt. polityki prywatności czy też zmiany szaty graficznej strony, jak możliwość przełączania się pomiędzy trybem jasnym i ciemnym.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, design

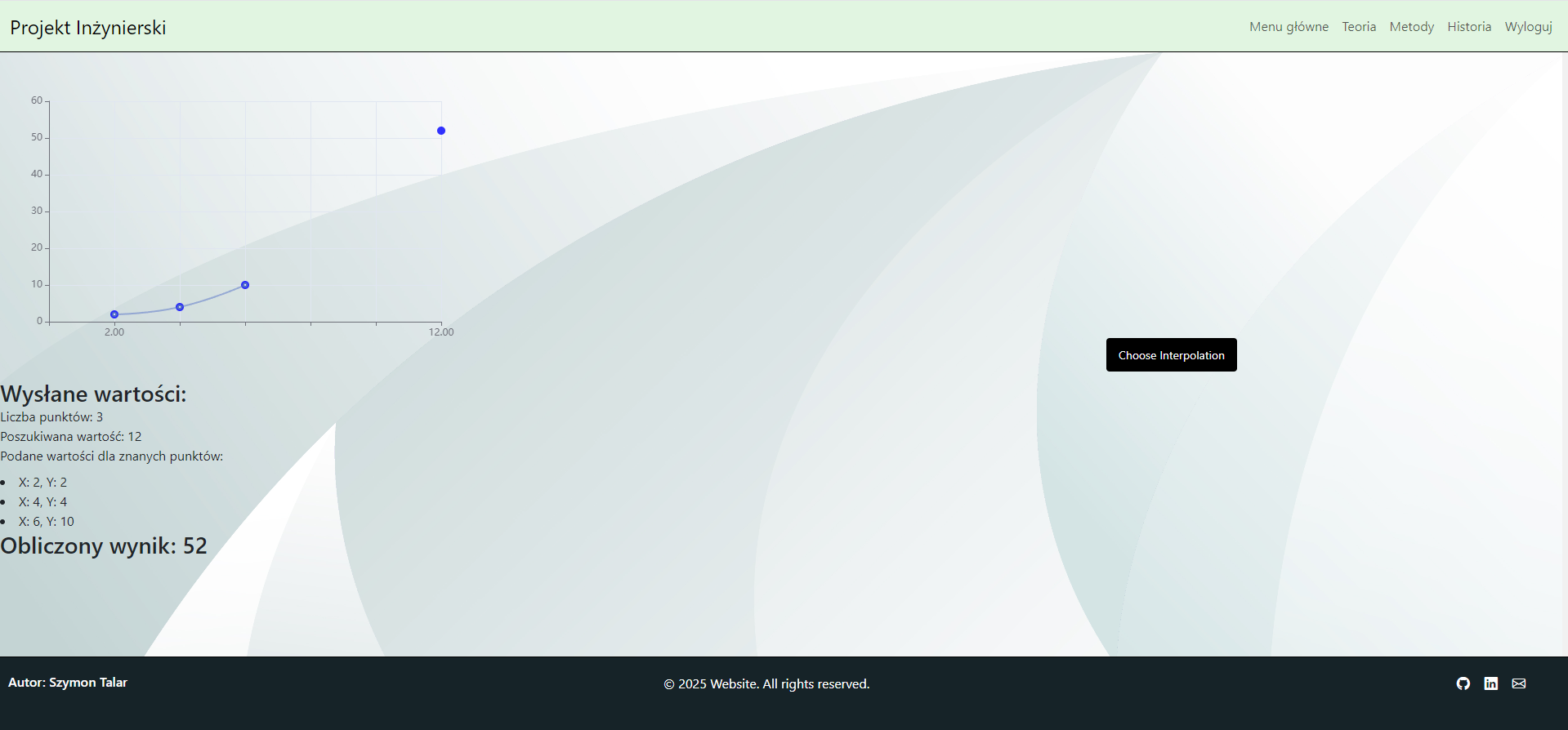
Opis wygenerowany automatycznie

Rys. 5. Ekran przedstawiający listę metod możliwych do wybrania dla dalszego wykonania obliczeń. Źródło: opracowanie własne

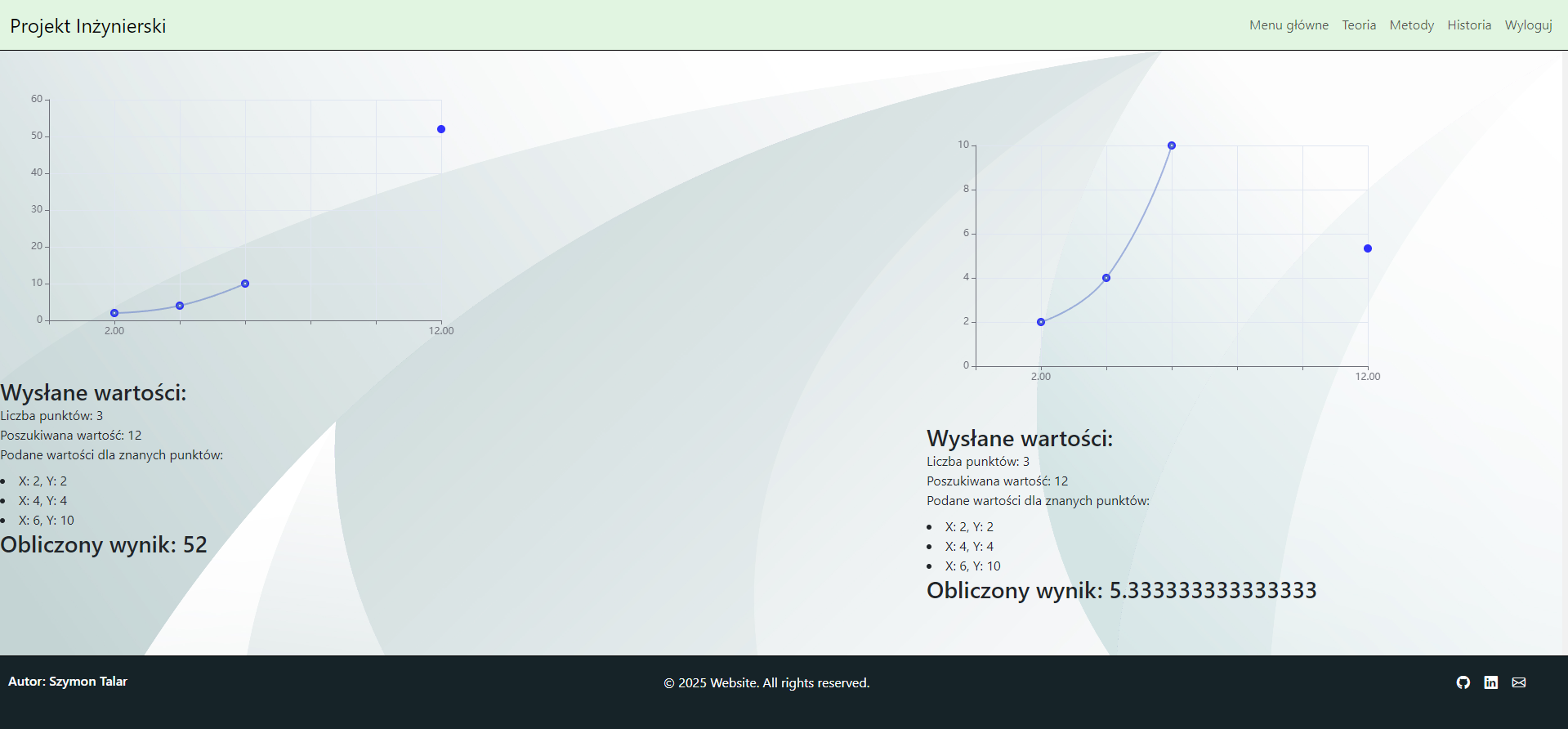
Po wybraniu dowolnej metody, na środku ekranu pojawi się dodatkowe okienko pozwalające wpisać dane potrzebne do wywołania pożądanego algorytmu.



Rys 6. Ekran z dodatkowym oknem pomocniczym dla danych wejściowych. Źródło: opracowanie własne

Jak widać, ekran pomocniczy zawiera okienka umożliwiające wpisanie niezbędnych zmiennych: liczby znanych punktów, wartości *x* oraz *y* tych punktów, a także wartość potrzebną do znalezienia. Po uzupełnieniu takiego formularza i wciśnięciu przycisku ‘Oblicz’, dane te są przekazywane do odpowiedniego kontrolera, a ponieważ została wybrana interpolacja wielomianowa, będzie to analogiczny kontroler do tego oznaczonego we Fragmencie Kodu 5. Po wykonaniu obliczeń, program przenosi nas na stronę zawierającą rezultat.  
  


Rys 7. Ekran wynikowy dla danych wysłanych przy pomocy ekranu z Rys 6. Źródło: opracowanie własne

Ekran wynikowy, oprócz obliczonego wyniku, zawsze będzie wyświetlał wszystkie dane podane przez użytkownika, a także w tym przypadku wykres przezentujący obliczone rozwiązanie. Oprócz tego, mamy również możliwość wykonania analogicznej operacji korzystając z innej metody obliczeniowej dzięki przyciskowi po drugiej stronie ekranu. Rys 8. Prezentuje stan strony po skorzystaniu z niego i wykonaniu dwukrotnych obliczeń dla tych samych danych.  
  


Rys 8. Porównanie ze sobą wyników dwóch metod interpolacyjnych: wielomianowej i trygonometrycznej

Wyniki obu tych operacji zostają zapisane do bazy danych do późniejszego wglądu użytkownika.

1. **Dodanie sekcji pomocniczych**

Oprócz podstawowych funkcjonalności jakie powinna oferować strona tego typu, czyli listy metod możliwych do obliczenia za pomocą kalkulatora i wyświetlenie wyniku, projekt zakładał również utworzenie dwóch sekcji pomocniczych: opisu teoretycznego oraz historii wyników.

1. **Opis Teoretyczny**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Rys 9. Ekran przedstawiający podstronę projektu z krótkim opises metod numerycznych zawartych w projekcie

Źródło: opracowanie własne

Ta podstrona, jak sama nazwa wskazuje, ma za zadanie pomóc osobom korzystającym ze strony w zapoznaniu się z danymi metodami od strony teoretycznej. W tym celu, zamieściłem na niej krótki opis każdej z nich, wraz z obrazem prezentującym przybliżone działanie danej metody, a także dodawałem istotne wzory wykorzystywane w obliczeniach o ile to było możliwe. Pod opisem każdej metody umieszczone zostały linki do stron zewnętrznych, na których użytkownik może otrzymać więcej informacji w temacie konkretnej metody.

1. **Historia wyników**

Obraz zawierający tekst, zrzut ekranu, design

Opis wygenerowany automatycznie

Rys 10. Podstrona zawierająca historię wyników użytkownika. Źródło: opracowanie własne

W tym miejscu przedstawiane będą wszystkie wykonane przez użytkownika obliczenia na stronie. Dla każdej operacji, wyświetlana będzie wybrana metoda numeryczna, a także informacje na temat wpisanych wcześniej danych wejściowych. Ponadto, obok danych znajdować się będzie również przycisk umożliwiające ponowne przeliczenie danych, co w przypadku często powtarzanego zestawu danych pozwoli użytkownikowi na zaoszczędzenie czasu przy wprowadzaniu danych.

1. **Testy aplikacji**

W celu przetestowania sprawnego działania wszystkich komponentów, zdecydowałem się skorzystać z dwóch technik:

1. **Implementacja testów jednostkowych**

Ten sposób sprawdzenia działania kodu był niezwykle przydatny dla pisania algorymów metod, które wykorzystywałem. Dzięki nim całość sprawdzenia działania danej metody sprowadzał się zaledwie do utworzenia kilku przypadków testowych, dla których wyniki były weryfikowane poza stroną korzystając z innych źródeł internetowych bądź też korzystając z samodzielnie wykonanych obliczeń, a następnie wprowadzenie utworzonego scenariusza jako obiekt z danymi dla algorytmu. Po tym następowało wywołanie funkcji, która jeśli zwróciła przewidywane rezultaty dla wszystkich przypadków – była uznawana za sprawną. Warto jednak w tym miejscu zaznaczyć, że przejście założonych testów nie daje gwarancji pełni sprawności algorytmów. Wynika to z faktu, iż testy jednostkowe które zostały zaimplementowane zawsze ograniczają się wyłącznie do pewnej puli przypadków, która niezależnie od ich doboru zawsze może pozostawiać ryzyko pominięcia pewnych specyficznych okoliczności dla których algorytm popełni błąd. W tym projekcie, ograniczyłem się do funkcji trzeciego rzędu jako tych najwyższego rzędu w testach jednostkowych, przez co błędy występujące wyłącznie dla wyższych rzędów mogły zostać niezauważone. Ponadto, nawet dla funkcji które uwzględniłem w swoich testach, z powodu błędów wynikających z braku odpowiedniej precyzji zmiennych oraz błędów wynikających z samej natury metod, których celem często jest wyłącznie przybliżenie poprawnego wyniku, a nie podanie jego dokładnej wartości, zmuszony byłem użyć porównania obejmującego błąd pomiarowy. We Fragmencie Kodu 9 ten błąd jest widoczny w wywołaniu funkcji *assertEquals* jako trzeci argument. Dla niektórych metod, takich jak interpolacja trygonometryczna, dodatkowym problemem był brak zewnętrznych źródeł pomocniczych podczas testowania algorytmu. Wymuszało to ręczne testowanie stosunkowo prostych przypadków, w celu wykrycia potencjalnych błędów algorytmu.   
  
****

Fragment Kodu 9. Przykładowy test jednostkowy sprawdzający działanie metody trapezów dla całkowania w przypadku funkcji liniowych. Źródło: opracowanie własne

1. **Samodzielne przejście ścieżek procesów**

Z racji na obecnie niewielką skalę projektu, zamiast automatyzować bardziej zaawansowane testy jak testy integracyjne lub testy E2E (ang. *end-to-end*), tę część testów postanowiłem wykonać manualnie. Wśród wykonywanych w ten sposób testów funkcjonalności mogę wymienić takie testy jak:

- testy obejmujące zakres działania bazy danych H2 potrzebnej do zapisywania użytkowników. Funkcjonalność tego segmentu była niezbędna do kontynuowania dalszych prac ponieważ jest to pierwszy ekran znajdujący się na stronie, a także część możliwości strony jest uzależniona od prawidłowego logowania się na stronę, tak więc w przypadku braku możliwości logowania strona utraciłaby sporą część dostępnych funkcjonalności

- testy prawidłowego wyświetlania się wyników algorytmów, które obejmowały przejście dla wszystkich algorytmów do ekranu wynikowego, takiego jak na Rys. 7. Z racji na zastosowanie osobnych punktów końcowych dla każdej metody, test ten również musiał być powtarzany osobny dla każdego istniejącego przypadku. Wraz z tym testem, sprawdzana była również możliwość wyświetlenia alternatywnej metody rozwiązania problemu.

- testy bazy MongoDB, zarówno przy pomocy zewnętrznego GUI(MongoDb Compass), jak i poprzez sprawdzanie rekordów znajdujących się w historii wyników po wykonaniu kilku różnych algorytmów.

Pomyślne przejście wyżej wymienionych testów definiowało pomyślne ukończenie projektu.

1. **Podsumowanie**

Strona, która została zaimplementowana na rzecz projektu, przeszła niezbędne testy w celu potwierdzenia sprawności funkcjonalności wymienionych we wcześniejszych rozdziałach. Pomimo braku możliwości pełnego przetestowania wszystich wdrożonych metod i możliwości istnienia błędów obliczeniowych, obecny stan umożliwia już na korzystanie ze strony i wykonywanie obliczeń, które przynajmniej dla prostych scenariuszy wyświetlą na ekranie wynik bliski temu rzeczywistemu. W połączeniu z wprowadzenymi dodatkowymi funkcjonalnościami, tworzy to projekt który potencjalnie daje możliwości obecnie niedostępne na innych stronach w ich darmowych wersjach, i które jednocześnie mogą przynieść dodatkowe korzyści w kontekście powszechnego użytkowania przez osoby, które chcą się nauczyć więcej o wybranych metodach numerycznych. Pomimo to, strona nadal pozostawia wiele możliwości do dalszego rozwoju. Pomijając ewentualną naprawę istniejących błędów i dalsze usprawnienie algorytmów w celu zwiększenia precyzji obliczeń, wśród potencjalnych nowych treści jakie dałoby się wprowadzić i które powinny jeszcze bardziej zwiększyć jej wartość edukacyjną wymieniłbym:

- dodanie kolejnych algorytmów, urozmajcając tym samym dostępną obecnie ofertę

- zwiększenie dostępności dla użytkowników mających określone trudności w korzystaniu ze standardowych stron, poprzez dodanie takich elemntów jak wdrożenie WCAG(ang. *Web Content Accessibility Guidlines*), dodanie atrybutów ARIA(ang. *Accessible Rich Internet Applications*), możliwość głosowego komunikowania się ze stroną, wprowadzenie możliwości dostosowania wielkości czcionki oraz przycisków na stronie, czy też uwydajnienie obliczeń wykonywanych a Javie np. poprzez zastosowanie wielowątkowości dla użytkowników dysponujących mniej zaawansowanymi sprzętami elektronicznymi.

- możliwość większej interakcji ze stroną, np. poprzez możliwość natychmiastowej zmiany danych w celu usprawnienia wykonywania wielu podonych obliczeń, a także wbudowanie interaktywnego wykresu, na którym użytkownik mógłby samemu dodawać/usuwać punkty wprowadzone na wejściu do programu.

- dodanie rozbudowanego opisu wyniku, zarówno przy pomocy kroków przejściowych podczas wykonywania obliczeń, jak i opisu słownego wygenerowanego przez sztuczną inteligencję, np. poprzez wykonywanie zapytań do udostępnionego przez OpenAI API.

1. **Źródła**
2. Wolfram Alpha, *Interpolating polynomial calculator,* <https://www.wolframalpha.com> [dostęp: 06.01.2025r.]
3. AtoZmath, *Numerical interpolation using Lagrange's Interpolation formula*, <https://atozmath.com> [dostęp: 06.01.2025r.]
4. Online Tools, https://tools.timodenk.com [dostęp: 06.01.2025r.]
5. Shivani Makwana, *C# vs Java: Key Differences Explained*, <https://www.ifourtechnolab.com/> [dostęp: 07.01.2025r.]
6. Anastasia Grishina, Roman Ogolikhin, *Java vs. C#: What is the Right Choice for Your Project?*, <https://softteco.com/> [dostęp: 07.01.2025r.]
7. *MongoDB vs. MySQL Differences*, <https://www.mongodb.com/> [dostęp: 07.01.2025r.]
8. Anthony P. Austin, *On trigonometric interpolation in an even number of points*, Kent State University 2023

1. Wolfram Alpha, *Interpolating polynomial calculator,* <https://www.wolframalpha.com> [dostęp: 06.01.2025r.] [↑](#footnote-ref-1)
2. AtoZmath, *Numerical interpolation using Lagrange's Interpolation formula*, <https://atozmath.com> [dostęp: 06.01.2025r.] [↑](#footnote-ref-2)
3. Online Tools, https://tools.timodenk.com [dostęp: 06.01.2025r.] [↑](#footnote-ref-3)
4. Shivani Makwana, *C# vs Java: Key Differences Explained*, <https://www.ifourtechnolab.com/> [dostęp: 07.01.2025r.] [↑](#footnote-ref-4)
5. Anastasia Grishina, Roman Ogolikhin, Java vs. C#: What is the Right Choice for Your Project?, <https://softteco.com/> [dostęp: 07.01.2025r.]

   [↑](#footnote-ref-5)
6. *MongoDB vs. MySQL Differences*, <https://www.mongodb.com/> [dostęp: 07.01.2025r.] [↑](#footnote-ref-6)
7. Anthony P. Austin, *On trigonometric interpolation in an even number of points*, Kent State University 2023 [↑](#footnote-ref-7)