**UNIWERSYTET WARMIŃSKO MAZURSKI W OLSZTYNIE**

**WYDZIAŁ MATEMATYKI I INFORMATYKI**

**ALEKSANDER CHOJNOWSKI**

**Informatyka**

**Aplikacja webowa do zarządzania budżetem domowym**

**Praca inżynierska**

**wykonana w Katedrze …………………**

**pod kierunkiem dr Paweł Drozda**

**Olsztyn 2023**

**UNIVERSITY OF WARMIA AND MAZURY IN OLSZTYN**

**FACULTY OF MATHEMATICS AND COMPUTER SCIENCE**

**ALEKSANDER CHOJNOWSKI**

**Computer Science**

**Web application for managing household budget**

**Engineering Thesis written in  
\*katedra\***

**Under supervision of dr Paweł Drozda**

**Olsztyn 2023**

**Spis treści**

[Streszczenie 5](#_Toc124340043)

[Abstract 5](#_Toc124340044)

[1. Wstęp 5](#_Toc124340045)

[2. Opis technologii 5](#_Toc124340046)

[2.1 Opis bibliotek JavaScript użytych w projekcie React.JS 7](#_Toc124340047)

[3. Analiza techniczna aplikacji 8](#_Toc124340048)

[3.1 Słownik pojęć 8](#_Toc124340049)

[3.2 Schemat bazy danych 8](#_Toc124340050)

[3.2.1 Opis poszczególnych tabel bazy danych 8](#_Toc124340051)

[3.3 Analiza przypadków użycia w systemie 8](#_Toc124340052)

[3.3.1 Diagram przypadków użycia 8](#_Toc124340053)

[3.3.2 Opis przypadków użycia 8](#_Toc124340054)

[3.3.3 Rozwinięcie opisu przypadku użycia … 8](#_Toc124340055)

[3.3.4 Rozwinięcie opisu przypadku użycia … 8](#_Toc124340056)

[4. Przewodnik po poszczególnych sekcjach i ekranach aplikacji 8](#_Toc124340057)

[4.1 Autoryzacja użytkownika 8](#_Toc124340058)

[5. Testowanie aplikacji 8](#_Toc124340059)

[6. Podsumowanie i wnioski 8](#_Toc124340060)

[Bibliografia 9](#_Toc124340061)

[Spis obrazów 9](#_Toc124340062)

[Spis tabel 9](#_Toc124340063)

[Spis diagramów 9](#_Toc124340064)

# Streszczenie

Celem pracy inżynierskiej jest stworzenie responsywnej aplikacji webowej, umożliwiającej zarządzanie i monitorowanie wydatków i przychodów w budżecie użytkownika, wraz z analizą techniczną zaprojektowanych funkcjonalności. Głównym założeniem było stworzenie przejrzystego, schludnego i przede wszystkim intuicyjnego interfejsu użytkownika, który ma zachęcać do korzystania z systemu i przyciągać nowych odbiorców. W tym celu przeanalizowane zostały funkcjonalności aplikacji pod kątem potrzeb klientów, a następnie powstały szczegółowe makiety, które przed wdrożeniem zostały przetestowane przez niezależnych testerów. Każda funkcja aplikacji została odpowiednio opisana i zaprezentowana za pomocą diagramów przypadków użycia oraz scenariuszy, które w sposób szczegółowy określają sposób działania danego fragmentu systemu. Praca zawiera także opis technologii użytych przy tworzeniu systemu, a także przewodnik po stworzonej aplikacji.

# Abstract

The aim of the engineering thesis is…

# 1. Wstęp

Tematem pracy inżynierskiej jest …

# 2. Opis technologii

**HTML** – (eng. HyperText Markup Language) Język znaczników wykorzystywany modelowania struktury aplikacji webowych oraz stron internetowych. Przy pomocy HTML programista jest w stanie uzyskać schemat w jakim zostaną wyświetlane dane. W skład języka wchodzą znaczniki określające cel ich zastosowania, na przykład <input/> używany jest, gdy oczekuje się od użytkownika wprowadzenia określonych danych.

**CSS** – (eng. Cascading Style Sheets) lista reguł, którymi poprzez wprowadzenie określonych wartości programista manipuluje wyglądem oraz zachowaniem się elementów struktury strony, między poprzez określenie koloru tła, rozmiarem elementu czy marginesem wewnętrznym i zewnętrznym, a także pozycjonowaniem elementu na stronie. Arkusz stylów zawiera także wiele reguł określanych jako pseudo-elementy, które umożliwiają zmianę wyglądu danego fragmentu strony w konkretnych, specyficznych sytuacjach.

**SCSS** – jeden z najpopularniejszych pre-procesorów do CSS. Optymalizuje czas i usprawnia pracę, którą należy wykonać podczas definiowania reguł arkusza styli przez programistę. Napisany kod w SCSS jest redagowany do formatu .css, który przeglądarka jest w stanie odczytać i zastosować do elementów na stronie

**JavaScript** – język programowania, który w aplikacjach webowych oraz stronach internetowych wywoływany jest po stronie użytkownika i wykorzystywany do interakcji z elementami struktury dokumentu. Dzięki JavaScript przeglądarka jest w stanie zarejestrować zdarzenia wywołane przez klienta, na przykład kliknięcie na dany element, i wywołanie określonych działań.

**React.JS** – Biblioteka JavaScript, optymalizująca i ułatwiająca tworzenie zaawansowanych projektów oraz aplikacji webowych. React ułatwia sposób manipulowania elementami na stronie oraz danymi, otrzymywanymi z serwera, a dobrze zaprojektowana struktura plików poprawia czytelność kodu oraz usprawnia dalszy rozwój aplikacji.

**Python** – Popularny język programowania wysokiego poziomu, najczęściej wykorzystywany do modelowania i wizualizacji danych, w uczeniu maszynowym, a także do tworzenia interfejsów aplikacji. Dzięki swojej prostocie i czytelności kodu, jest częstym wyborem do nauki programowania przez początkujących programistów. Posiada także bogatą ilość bibliotek rozbudowujących podstawowe funkcje języka.

**FastAPI** – Biblioteka języka Python wykorzystywana do tworzenia interfejsów programistycznych aplikacji, w skrócie API (eng. Application Programming Language). Poprzez swoją prostotę i przejrzystość umożliwia szybkie i sprawne tworzenie zaplecza aplikacji nawet przez mniej wprawionych programistów.

**Swagger UI** – Narzędzie, które w sposób automatyczny tworzy dokumentację dla utworzonych ścieżek końcowych interfejsu aplikacji, wraz z formularzami umożliwiającymi przetestowanie zaprojektowanych funkcjonalności.

**Git** – Najpopularniejszy system kontroli wersji, umożliwiający dokumentacje postępów pracy nad projektem. Jest także w pewnym sensie zabezpieczeniem kodu, gdyż poprzez odpowiednie komendy programista jest w stanie przywrócić wcześniejszy kod, na przykład wersję z działającym kodem, gdy aktualny odmawia posłuszeństwa.

**GitHub** – Serwis internetowy udostępniający możliwość przechowywania repozytoriów z kodem aplikacji bez konieczności opłaty za taką usługę. GitHub wykorzystuje system kontroli wersji GIT, dzięki któremu w łatwy sposób z poziomu wiersza poleceń na komputerze, można przesłać pliki na serwer serwisu, gdzie będą przechowywane.

**Visual Studio Code** – Popularny i darmowy edytor kodu źródłowego aplikacji. Posiada on przejrzysty i intuicyjny interfejs, a także formatowanie kodu poprzez kolorowanie składni co znacznie usprawnia i ułatwia pracę. Połączenie tych elementów z bogatym zbiorem wtyczek i dodatków tworzy świetne narzędzia do pracy z kodem.

## 2.1 Opis bibliotek JavaScript użytych w projekcie React.JS

**Material UI** – Biblioteka zawierająca pokaźny zbiór ikon i komponentów, poprawiających wygląd aplikacji.

**Date-fns** – Biblioteka, której zadaniem jest zmiana standardowego, zwracanego przez serwer oraz elementy na stronie, formatu daty i godziny na format czytelny dla użytkownika. Na przykład format „2022-01-23T12:34” Date-fns przeformatuje na „wczoraj o 12:34”

**Chart.js** – Biblioteka języka JavaScript umożliwiająca wizualizację danych na stronie internetowej. Otrzymując odpowiednio sformatowane dane wyświetla je w formie wykresu, a dzięki zdefiniowaniu konkretnych opcji jest w stanie animować je i poprawiać ich czytelność.

**Sweetalert2 i Toastify** – Biblioteki, których głównym zadaniem jest tworzenie i wyświetlanie w przejrzysty sposób informacji w formie alertów i popup`ów.

# 3. Analiza techniczna aplikacji

## 3.1 Słownik pojęć

## 3.2 Schemat bazy danych

### 3.2.1 Opis poszczególnych tabel bazy danych

## 3.3 Analiza przypadków użycia w systemie

### 3.3.1 Diagram przypadków użycia

### 3.3.2 Opis przypadków użycia

### 3.3.3 Rozwinięcie opisu przypadku użycia …

### 3.3.4 Rozwinięcie opisu przypadku użycia …

# 4. Przewodnik po poszczególnych sekcjach i ekranach aplikacji

## 4.1 Autoryzacja użytkownika

# 5. Testowanie aplikacji

# 6. Podsumowanie i wnioski

# Bibliografia

# Spis obrazów

# Spis tabel

# Spis diagramów