

# Sprawozdanie

## Aplikacja CURRENCY\_CONV

### 1. Wybór języka programowania i opis działania programu.

#### Wybór języka programowania:

Do stworzenia aplikacji wybrałem język JavaScript, z wykorzystaniem biblioteki React. React jest popularną biblioteką służącą do budowy interfejsów użytkownika w aplikacjach internetowych. React pozwala na łatwe zarządzanie stanem aplikacji oraz umożliwia tworzenie dynamicznych komponentów.

#### Opis działania programu:

Program to kalkulator walutowy, który pozwala użytkownikowi na przeliczenie kwoty w PLN na wybraną walutę obcą. Program jest aplikacją internetową, która działa w trybie okienkowym (GUI), a nie w trybie konsolowym. Użytkownik podaje kwotę w PLN, wybiera walutę docelową, a program oblicza równowartość tej kwoty w wybranej walucie. Wynik jest wyświetlany na ekranie.

#### Dane wejściowe:

- Kwota w PLN (wartość liczby, która jest wprowadzana przez użytkownika w polu tekstowym).
- Waluta, na którą użytkownik chce przeliczyć (wybór z listy rozwijanej, np. USD, EUR, GBP).

#### Dane wyjściowe:

- Wynik konwersji, czyli kwota w PLN przeliczona na wybraną walutę, wyświetlana na ekranie użytkownika w formacie: "Kwota PLN jest równa X w wybranej walucie".

#### Tryb interfejsu użytkownika:

Program działa w trybie okienkowym (GUI). Użytkownik wchodzi w interakcję z aplikacją za pomocą formularzy, przycisków i rozwijanych list. Wszelkie obliczenia są wykonywane po kliknięciu przycisku "Convert", a wynik jest natychmiast wyświetlany w interfejsie.

## **Użyte elementy obiektowości, algorytmy i biblioteki:**

### **Elementy obiektowości:**

Program wykorzystuje React (biblioteka JavaScript), która jest oparta na komponentach. Każdy komponent (np. formularz wejściowy, przycisk, wynik) może być traktowany jako obiekt, który przechowuje stan i reaguje na zdarzenia użytkownika.

### **Algorytmy:**

Program opiera się na prostym algorytmie konwersji walut. Po wprowadzeniu kwoty i wybraniu waluty, program sprawdza odpowiedni kurs wymiany z obiektu `ExchangeRates` i przelicza kwotę w PLN na wybraną walutę.

### **Biblioteki:**

Wykorzystano React do budowy aplikacji oraz `useState` z Reacta do zarządzania stanem aplikacji (kwoty wejściowej, waluty, wyniku).

### **Obsługa błędów:**

Program posiada prostą obsługę błędów. Jeżeli użytkownik nie wprowadzi liczby lub wpisze nieprawidłową wartość, wyświetli się komunikat o błędzie. Przykładem jest walidacja, która sprawdza, czy wprowadzone dane są liczbą oraz czy użytkownik wprowadził wartość przed kliknięciem przycisku. Użyto w tym celu instrukcji warunkowych i funkcji `isNaN()`.

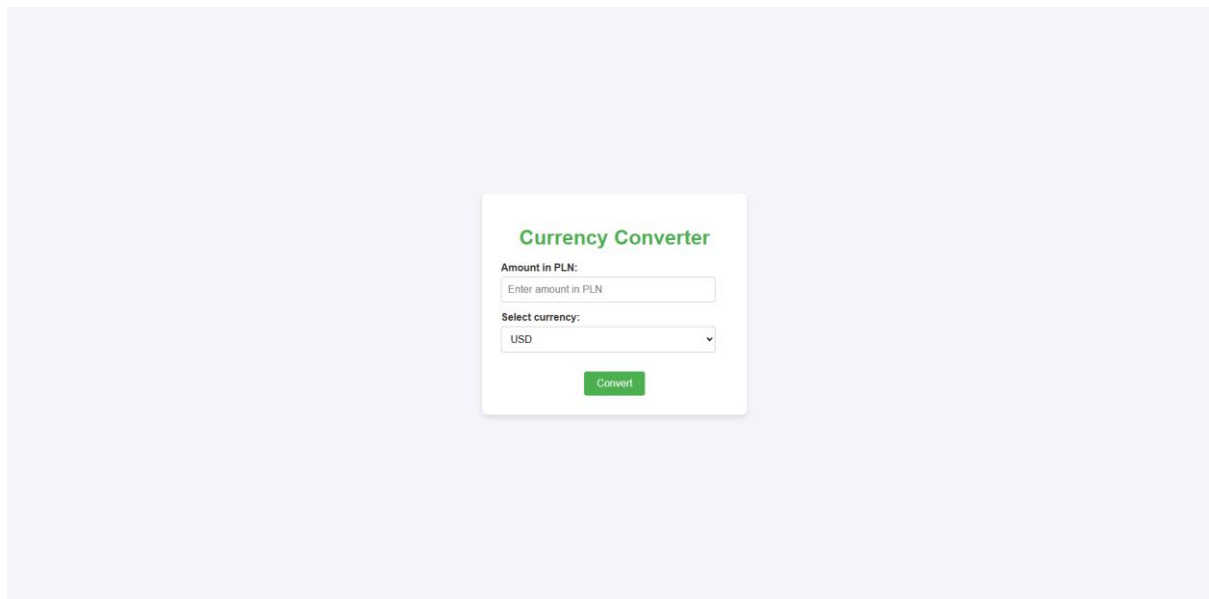
W przypadku błędów użytkownika, takich jak wprowadzenie tekstu zamiast liczby, wyświetlany jest komunikat z prośbą o poprawne wprowadzenie danych.



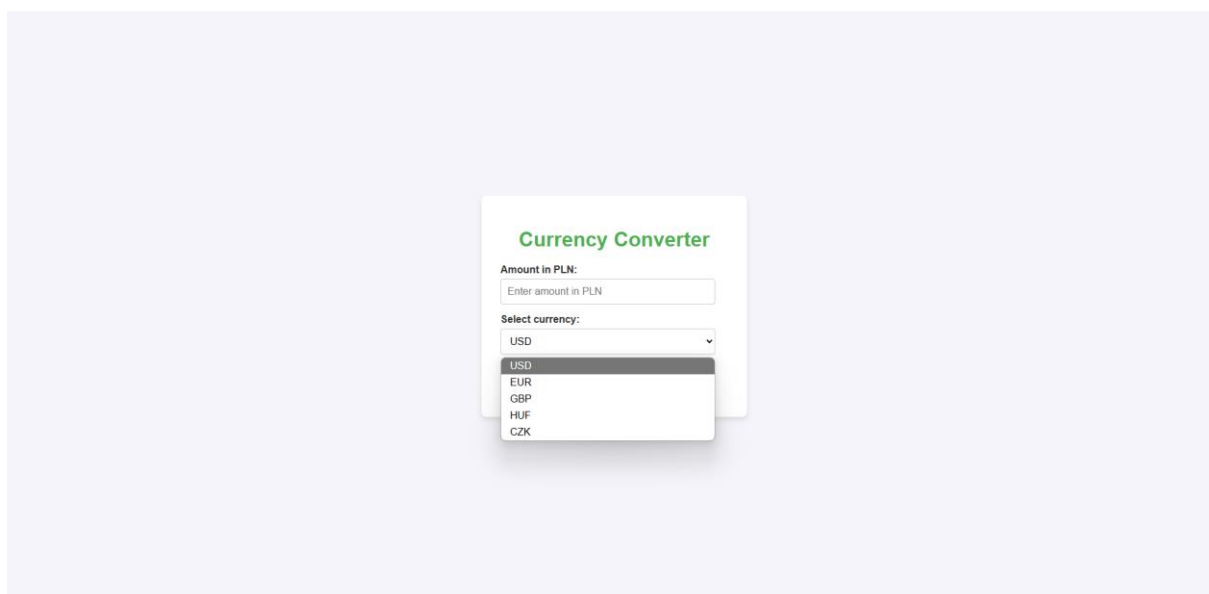
## Zrzuty ekranu:

Widok aplikacji:

Pokazuje formularz z polem do wprowadzenia kwoty, rozwijaną listą walut i przyciskiem konwersji.

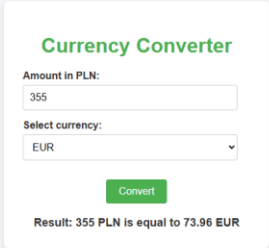


Wybór walut:



Wynik po konwersji:

Po kliknięciu przycisku "Convert" pojawia się wynik przeliczenia.



The screenshot shows a web form titled "Currency Converter". It includes a text input field labeled "Amount in PLN:" containing the number "355". Below it is a dropdown menu labeled "Select currency:" with "EUR" selected. A green button labeled "Convert" is positioned below the dropdown. At the bottom of the form, the result is displayed: "Result: 355 PLN is equal to 73.96 EUR".

### 3. Ocena programu

#### Ocena działania programu:

Program działa zgodnie z założeniami. Użytkownik może wprowadzić kwotę w PLN, wybrać walutę i uzyskać wynik przeliczenia. Obsługuje błędy takie jak niepoprawny format danych wejściowych (np. litery zamiast liczb), wyświetlając komunikat ostrzegawczy.

#### Obsługa błędów:

##### Błędy miękkie (soft errors):

Aplikacja obsługuje błędy związane z wprowadzeniem danych, np. jeżeli użytkownik nie poda liczby w polu kwoty lub poda wartość nie będącą liczbą (tekst), aplikacja wyświetli komunikat o błędzie, prosząc o poprawne wprowadzenie danych. Komunikaty błędów są prostymi alertami JavaScript.

##### Błędy twarde (hard errors):

W przypadku błędów, które wystąpiłyby w wyniku problemów w kodzie (np. zmiana w strukturze ExchangeRates), aplikacja nie będzie w stanie kontynuować działania.

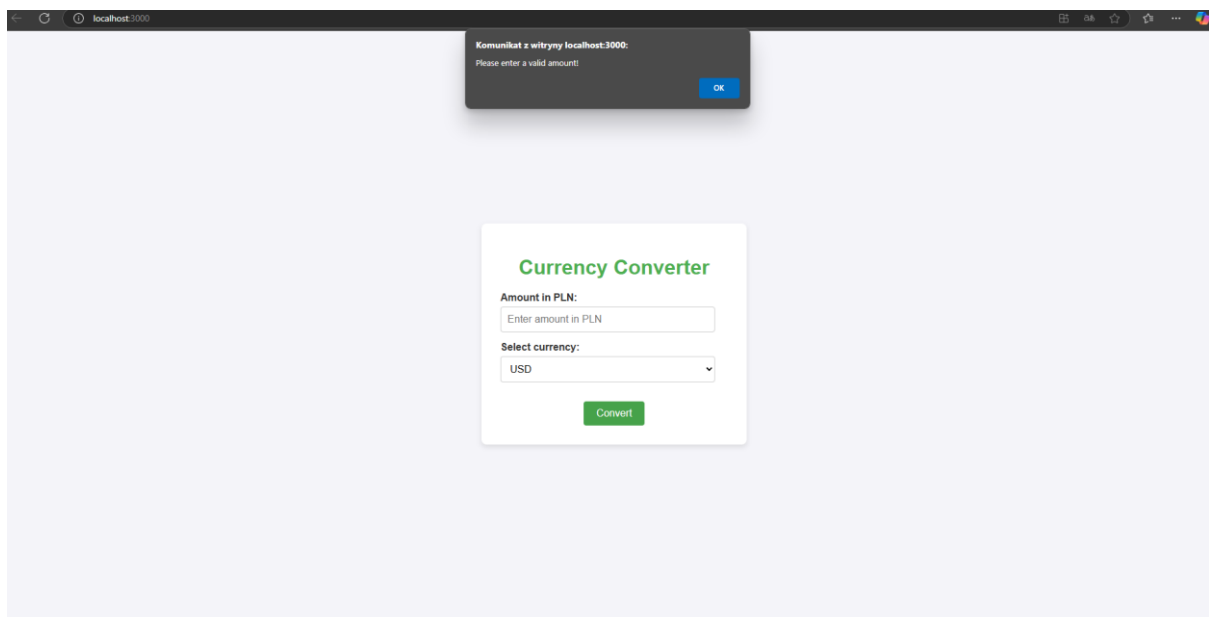
Takie błędy powinny zostać złapane i obsłużone przez odpowiednie mechanizmy, np. użycie try-catch, ale obecnie aplikacja nie ma zaawansowanej obsługi błędów.

### Przykład obsługi błędu:

Wprowadzenie tekstu zamiast liczby w polu kwoty skutkuje wyświetleniem komunikatu: "Please enter a valid amount!".

Program również poprawnie obsługuje wybór nieistniejącej waluty w przypadku zmiany kursów wymiany.

### Zrzut ekranu:



Wszystkie komunikaty o błędach i sukcesie są wyświetlane w oknie alertu lub w elemencie HTML.

### Podsumowanie:

Aplikacja jest prosta, ale funkcjonalna. Wykorzystuje podstawy React, takie jak komponenty, zarządzanie stanem oraz prostą logikę konwersji walut. Obsługuje błędy w sposób podstawowy i jest gotowa do dalszego rozwinięcia, np. o dynamiczne pobieranie kursów wymiany z API.