

Projekt zaliczeniowy

Poniżej znajdują się zasady zaliczenia projektu semestralnego z kursu Programowanie Frontend - założenia aplikacji, wymagania techniczne, kryteria, które należy spełnić, aby uzyskać konkretną ocenę oraz trochę przydatnych linków. Waszym zadaniem będzie stworzenie aplikacji służącej do wyświetlania **prognozy pogody** przy pomocy biblioteki React.

Założenia funkcjonalne aplikacji

Aplikacja powinna posiadać wymione niżej funkcjonalności. Zostały one podzielone na konieczne, czyli **niezbędne do uzyskania oceny 3.0**, oraz dodatkowe, które pozwolą Wam na zdobycie wyższej oceny.

Wymagania konieczne

Konieczne wymagania funkcjonalne to:

- wyświetlanie listy co najmniej 5 miejscowości na ekranie głównym,
- możliwość podglądu szczegółów prognozy pogody dla podanej miejscowości, a w tym:
 - bieżąca temperatura (w stopniach Celsjusza),
 - bieżące warunki pogodowe (w formie odpowiedniej ikony),
 - prognozowana temperatura i warunki pogodowe na najbliższe 5 dni,
 - prawdopodobieństwo wystąpienia opadów (wyrażona w procentach), ich rodzaj oraz ilość (wyrażona w milimetrach na metr kwadratowy),
 - prędkość i kierunek wiatru,
 - stopień zachmurzenia.

Można zainspirować się aplikacjami takimi jak [Windy](#) czy [ICM Meteo](#).

Wymagania dodatkowe

Do dodatkowych wymagań należą:

- możliwość globalnej zmiany jednostek temperatury (Celsjusz/Fahrenheit/Kelvin),
- dodanie formularza umożliwiającego wyszukiwanie miast,
- oznaczanie ulubionych miast i wyświetlanie ich na osobnej podstronie,
- integracja z API udostępniającym prawdziwe warunki pogodowe, np. darmowym i publicznie dostępnym [OpenWeatherMap API](#).

Wymagania na poszczególne oceny

Poniżej znajdziecie jakie wymagania należy spełnić, aby uzyskać konkretną ocenę - zarówno te funkcjonalne, wymienione wyżej, jak i techniczne.

3.0

- wszystkie wymagania funkcjonalne opisane w sekcji [Wymagania konieczne](#),
- użycie w aplikacji hook'ów takich jak `useState`, `useCallback`, `useMemo`, `useEffect`,
- przełączanie pomiędzy szczegółowym widokiem dla każdej miejscowości za pomocą natywnych hooków (`useState`, `useReducer`),
- tworzenie reużywalnych komponentów do powtarzalnych części kodu,
- zastosowanie stylów CSS do zdefiniowania wyglądu komponentów,

Aby uzyskać ocenę 3.0, nie trzeba korzystać z prawdziwych danych pogodowych, ale można jest *na sztywno* wpisać do aplikacji. Ważne jednak, aby każda z miejscowości prezentowała **inne warunki pogodowe**.

3.5

- wszystkie wymagania na ocenę 3.0,
- nawigacja pomiędzy poszczególnymi podstronami aplikacji przy użyciu biblioteki [React Router](#),
- użycie biblioteki [Redux](#) do globalnej zmiany jednostek temperatury (dla wszystkich podstron miejscowości),

4.0

- wszystkie wymagania na ocenę 3.5,
- dodanie formularza umożliwiającego wyszukiwanie miejscowości po jej nazwie na ekranie z listą miejscowości,

4.5

- wszystkie wymagania na ocenę 4.0,
- dodanie możliwości wyboru (np. przy pomocy ikony *gwiazdki* wyświetlanej przy każdej z miejscowości) i zapisywanie tych informacji w globalnym stanie aplikacji (Redux),
- przechowywanie stanu aplikacji (wybrane jednostki temperatury oraz ulubione miejscowości) w [local storage](#) przeglądarki; po odświeżeniu strony jej ustawienia powinny zostać zachowane

5.0

- wszystkie wymagania na ocenę 4.5,
- integracja przy wykorzystaniu np. biblioteki [Axios](#), z OpenWeatherMap API i wyświetlanie rzeczywistych danych pogodowych dla miejscowości na liście,

Jak zacząć?

Aplikacja stworzona w ramach kursu musi zostać wykonana przy pomocy biblioteki **React**. Istnieje dowolność co do konfiguracji projektu i zastosowanych bibliotek. Można również skorzystać, do czego zachęcam, z [przygotowanego szablonu projektu](#) dostępnego na moim GitHubie oraz na platformie Moodle w formie pliku w formacie ZIP. Projekt zawiera proponowaną strukturę oraz kilka zainstalowanych przydatnych bibliotek (szczegóły znajdują się w pliku [README](#)).

Można również stworzyć własny, prosty projekt przy użyciu narzędzia [Create React App](#).

Przydatne link

W projekcie można wykorzystać różne biblioteki dedykowane dla ekosystemu React'a. Poniżej znajduje się kilka z nich (*dla chętnych*):

- [TailwindCSS](#),
- [React Redux](#),
- [Vite.js](#),
- [Axios](#),
- [Create React App](#),
- [Szablony projektów opartych na React + Vite.js](#),
- [Dokumentacja biblioteki React](#),
- [Dokumentacja MDN](#).