



SYLABUS PRZEDMIOTU

Nazwa przedmiotu: Technologie frontendowe

Kod przedmiotu: TFN

Kierunek / Profil: Informatyka / praktyczny

Tryb studiów: stacjonarny

Rok / Semestr: 3 / 5

Charakter: obowiązkowy

Odpowiedzialny: mmiotk@pjwstk.edu.pl

Wersja z dnia: 19.02.2026

1. Godziny zajęć i punkty ECTS

Wykłady	Ćwiczenia	Laboratoria	Z prowadzącym	Praca własna	Łącznie	ECTS
30 h	30 h	—	60 h	65 h	125 h	5

2. Forma zajęć

Forma zajęć	Sposób zaliczenia
Laboratorium	Zaliczenie z oceną
Wykład	Egzamin

3. Cel dydaktyczny

Celem przedmiotu jest zapoznanie się z nowoczesnymi technologiami, technikami oraz narzędziami do tworzenia graficznego interfejsu użytkownika w aplikacjach webowych.

4. Przedmioty wprowadzające

Przedmiot	Wymagane zagadnienia
Warsztaty programistyczne	Technologie internetu
Dobra znajomość HTML, CSS oraz JavaScript	—

5. Treści programowe

1. Historia frontendu. Język znaczników HTML i CSS
2. Użycie preprocesorów CSS jako alternatywa pisania stylów w aplikacjach webowych
3. Animacje oraz inne technologie wspierające proces pisania interfejsów użytkownika za pomocą CSS
4. Pojęcie responsywności oraz dostępności w aplikacjach webowych
5. Architektura aplikacji SPA: Środowisko Node.js
6. Wprowadzenie do framework'a budowania aplikacji webowych w środowisku Node.js
7. Format JSX. Integracja CSS w aplikacji.
8. Pojęcie stanu aplikacji w aplikacjach webowych
9. Formularze oraz ich walidacja w aplikacjach webowych
10. Inne rodzaje hooków i dobre praktyki w tworzeniu aplikacji webowych
11. Integracja danych w aplikacjach webowych
12. Trasowanie aplikacji w aplikacjach webowych
13. Testowanie aplikacji frontendowych
14. Technologia suspense w aplikacjach webowych
15. Metody optymalizacji w aplikacjach webowych
16. Informacje dodatkowe
17. Wymagane oprogramowanie w laboratoriach: Serwer Node.js.
18. Uzasadnienie dla prowadzenia przedmiotu/współpraca z rynkiem pracy
19. A. W jakiego typu firmach będą potrzebne umiejętności nabyte w trakcie zajęć:
20. Wszystkie firmy zajmujące się budowaniem serwisów internetowych.
21. B. W jakich zawodach wiedza i umiejętności są istotne:
22. Wiedza i umiejętności z tworzenia interaktywnych interfejsów użytkownika jest bardzo przydatna we
23. wszystkich zawodach informatycznych.

6. Efekty kształcenia

Wiedza

- Student zna i rozumie pojęcia związane z pojęciem Frontend Development. Student zna i rozumie pojęcie preprocesora CSS oraz tworzenia aplikacji webowej z wykorzystaniem narzędzia Node.js oraz przykładowego narzędzia w tym środowisku.

- Student zna i rozumie zaawansowane pojęcia związane z pojęciem Frontend Development. Student zna i rozumie zaawansowane pojęcie preprocesora CSS oraz tworzenia aplikacji webowej z wykorzystaniem narzędzia Node.js oraz przykładowego narzędzia w tym środowisku.
- Student zna i rozumie elementy programowania funkcyjnego na przykładzie języka JavaScript. Student zna i rozumie techniki oraz metody programistyczne wykorzystywane w języku JavaScript.
- Student zna i rozumie pojęcia z zakresu tworzenia interaktywnych aplikacji internetowych, umożliwiających komunikację typu człowiek-komputer.

Umiejętności

- Student potrafi ocenić przydatność różnych podejść programistycznych na podstawie języka JavaScript i związanych z nimi środowisk na przykładzie narzędzia Node.js
- Student potrafi wyspecyfikować, zaprojektować, zaimplementować, przetestować oraz zdebuggować aplikację webową; potrafi korzystać z bibliotek przy użyciu narzędzia npm, środowisk programistycznych, integrujących i uruchomieniowych.
- Student potrafi wytworzyć warstwową aplikację webową w oparciu o wybrane wzorce architektoniczne , przy pomocy narzędzia Node.js oraz wybranego narzędzia w tym środowisku.
- Student potrafi wytworzyć zaawansowaną warstwową aplikację webową w oparciu o wybrane wzorce architektoniczne , przy pomocy narzędzia Node.js oraz wybranego narzędzia w tym środowisku.

Kompetencje społeczne

- Student jest gotów do jest gotów do samodzielnego uczenia się przez całe życie

7. Kryteria oceny

- wykład z elementami dyskusji z prezentacją multimedialną
- burza mózgów
- rozwiązywanie zadań
- analiza przypadków
- projekt praktyczny
- Kryteria oceny
- 50% Kolokwium
- 50% Projekt praktyczny
- 40% Ocena z laboratorium
- 60% Egzamin

8. Metody dydaktyczne

Wykład, laboratoria, praca własna studenta.

9. Literatura

Podstawowa:

- E. Porcello and A. Banks, Learning react: Modern patterns for developing react apps, O'Reilly Media, Incorporated, 2020.
- A. Prabhu, Beginning CSS Preprocessors. With Sass, Compass and Less, Apress, 2015.
- C. Rippon, Learn react with typescript 3: Beginner's guide to modern react web development with typescript 3, Packt Publishing, 2018.
- M.T. Thomas, React in action, Manning Publications, 2018.
- F. Zammetti, Modern full-stack development: Using typescript, react, node.js, webpack, and docker, Apress, 2020.

Uzupełniająca:

- A. Boduch and R. Derks, React and react native: A complete hands-on guide to modern web and mobile development with react.js, Packt Publishing, 2020.
- E. Brown, Web Development with Node and Express, O'Reilly Media, Incorporated, 2020.
- A. Freeman, Pro react 16, Apress, 2019.
- D. Griffits and D. Griffits, React Cookbook. Recipes for Mastering the React Framework, O'Reily, 2021.
- G. Lim, Beginning react with hooks, Independently Published, 2020.
- C.S. Roldán, React cookbook: Create dynamic web apps with react using redux, webpack, node.js, and graphql, Packt Publishing, 2018.
- R. Wieruch, The road to react: Your journey to master plain yet pragmatic react.js, Independently Published, 2017.