

Dokumentacja projektowa z przedmiotu PAINT w realizacji 2024Z

Temat: StudHub - portal wspomagający studenta

Skład zespołu:

- Jeż Adam - kierownik, bazodanowiec, backend
- Maciej Kozłowski - backend, frontend, bazodanowiec
- Jakub Stacherski - backend, frontend, integracja

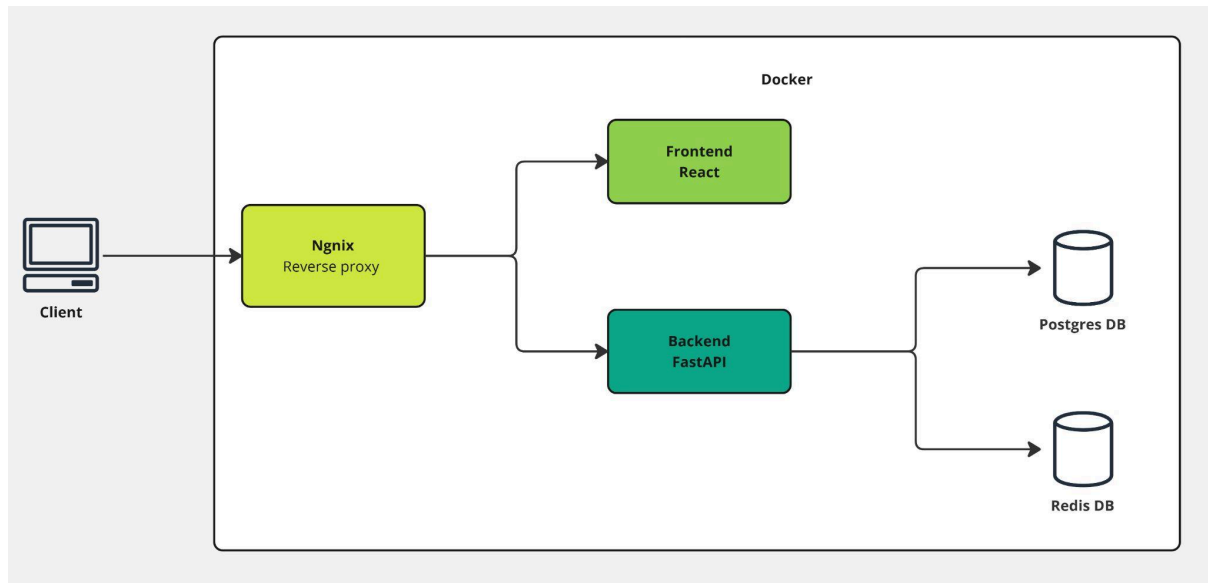
Opis projektu:

Portal wspomagający studenta w zaliczeniu semestru. Umożliwia użytkownikowi wyświetlanie obecnie realizowanych przedmiotów, dzięki integracji z systemem USOS PW. Pokazuje, na których przedmiotach powinien się obecnie skupić. Wymiana informacji między studentami jest umożliwiona za pomocą czatów grupowych.

Opis szczegółowych założeń funkcjonalnych:

- Definiowanie przez użytkownika warunków zaliczeń i wydarzeń dla poszczególnych przedmiotów
- Wskazywanie stanu zaliczenia przedmiotu: wszystkie warunki zostały spełnione / przedmiot jest w trakcie zaliczenia / przedmiot jest niezaliczony (nie ma możliwości zaliczenia)
- Kalendarz umożliwiający przeglądanie dodanych wydarzeń (np. sprawdziany, laboratoria)
- Uwierzytelnianie i pobieranie informacji z systemu USOS PW (integracja)
- Czaty grupowe dla użytkowników (oddzielny czat dla każdego przedmiotu; studenci są dodawani do czatów na podstawie ich listy podpięć - jeżeli czat dla danego przedmiotu nie istnieje, zostaje stworzony)

Opis szczegółowych założeń architektonicznych aplikacji i opis techniczny (“wewnętrzny”) produktu:



Aplikacja składa się z następujących modułów:

- Frontend (warstwa prezentacji) - *React*
- Backend (warstwa logiki biznesowej) - *FastAPI*, *SQLModel*
- Reverse proxy - *Nginx*
- Database (przechowywanie danych użytkownika i jego przedmiotów) - *Postgres*
- Messaging service (przechowywanie wiadomości z czatów przedmiotowych) - *Redis*

Aplikacja składa się z kilku modułów, które wspólnie tworzą kompletny ekosystem umożliwiający sprawne działanie systemu. Warstwa frontendowa, odpowiedzialna za interfejs użytkownika, została zaimplementowana przy użyciu biblioteki *React*, co zapewnia dynamiczną i responsywną obsługę aplikacji. Backend, pełniący rolę logiki biznesowej, opiera się na frameworku *FastAPI* oraz *SQLModel*, co umożliwia szybkie przetwarzanie żądań oraz efektywne zarządzanie danymi. Komunikacja z użytkownikami jest wspierana przez reverse proxy *Nginx*, który odpowiada za bezpieczeństwo aplikacji. Dane użytkowników oraz ich przedmiotów przechowywane są w bazie danych *PostgreSQL*, natomiast wiadomości z czatów przedmiotowych obsługiwane są za pomocą usługi *Redis*.

Wszystkie moduły aplikacji zostały skonteneryzowane przy użyciu narzędzia *Docker*, co pozwala na łatwe zarządzanie środowiskiem, skalowalność oraz izolację poszczególnych komponentów. Choć konteneryzacja rozwiązuje wiele problemów związanych z wdrażaniem i utrzymaniem aplikacji, nie jest rozwiązaniem idealnym. Potencjalne wyzwania to m.in. dodatkowa warstwa złożoności, zarządzanie zasobami systemowymi oraz konieczność monitorowania i optymalizacji wydajności kontenerów.

Komunikacja z *USOS API* inicjowana jest przez backend, który przekierowuje użytkownika na stronę logowania *USOSa*, a po zalogowaniu, użytkownik wraca na *StudHub* z tokenem zapisanym w ciasteczkach.

Realizacja projektu:

Proces powstawania aplikacji:

1. Zdefiniowanie celu projektu:

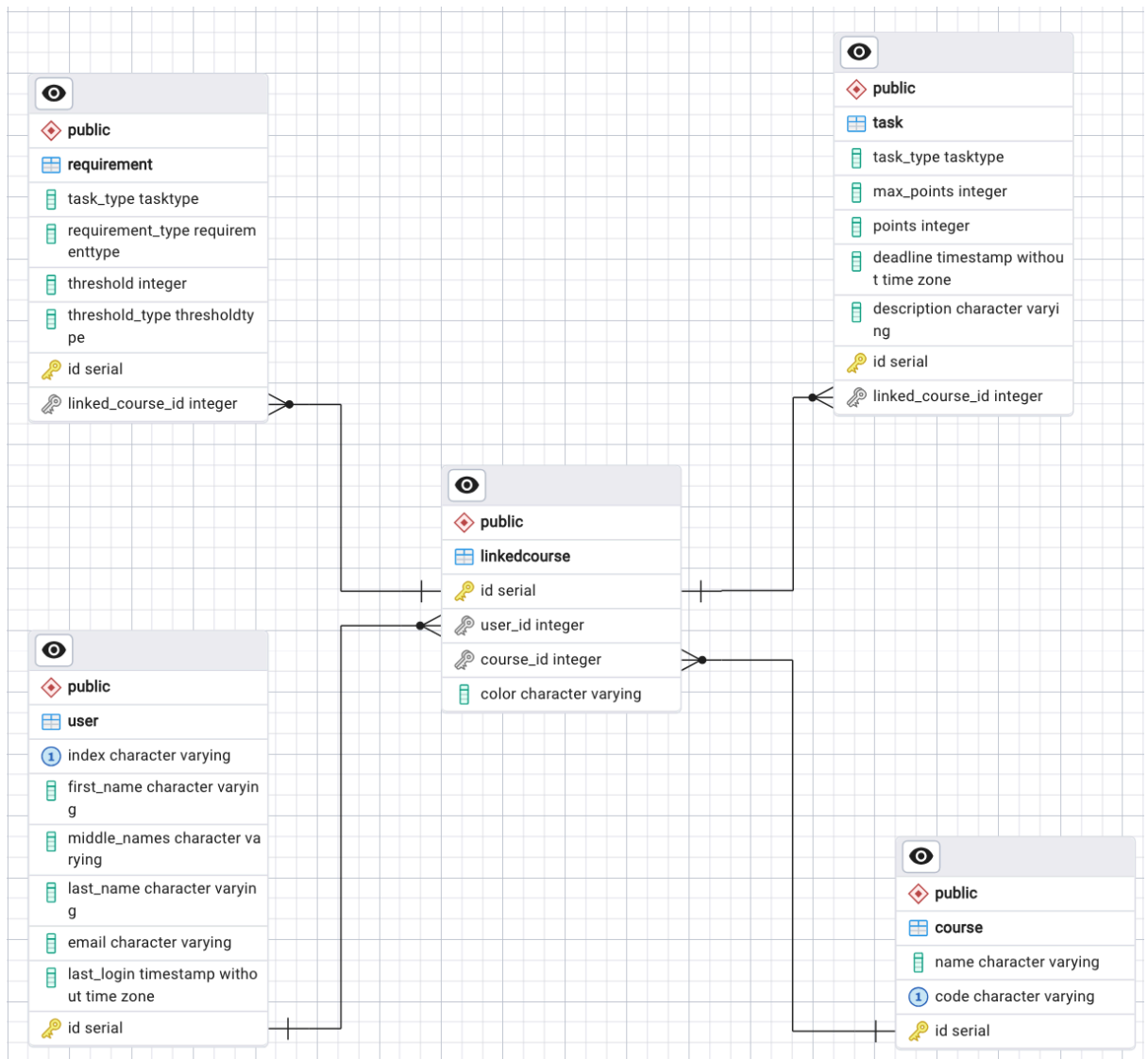
Projekt miał być wsparciem do monitorowania postępu stopnia zaawansowania studenta w danym semestrze na studiach.

2. Analiza wymagań projektowych:

Zdefiniowanie głównych wymagań funkcjonalnych z perspektywy studenta, który miał założone konto na portalu USOS PW.

3. Wybór technologii i rozwiązań:

Stworzenie diagramu architektury przedstawiający zależności i rolę poszczególnych komponentów. Analiza dostępnych narzędzi spełniających dane role. Wybranie odpowiednich rozwiązań technologicznych. Do roli serwera webowego wybraliśmy *uvicorn*, który jest zintegrowany z frameworkiem *FastAPI*, w którym też zaimplementowaliśmy logikę biznesową i API. Zaprojektowaliśmy diagramy bazy danych:



Do przechowywania struktury relacyjnej danych wybraliśmy bazę danych Postgres. Wiadomości z czatów przedmiotowych zostały przechowywane w bazie nierelacyjnej Redis do szybkiego przetwarzania tych danych w czasie rzeczywistym (*Redis Streams*), z uwzględnieniem ich kolejności.

4. Implementacja wymagań funkcjonalnych za pomocą narzędzi.
5. Testowanie:
Równoległe tworzenie testów jednostkowych wraz z rozwijaniem aplikacji. Dalej testy integracyjne łączące różne komponenty (w tym z systemem uwierzytelniania dostępnym przez USOS PW). Na koniec testy akceptacyjne w docelowym środowisku, aby sprawdzić poprawność zdefiniowanych wymagań funkcjonalnych (interakcje użytkownika z aplikacją, wykrywanie krytycznych błędów, sprawdzanie poprawności działania poszczególnych rozwiązań).

Instalacja

Wymagania do instalacji:

- System zgodny ze standardem POSIX
- Zainstalowane narzędzie Docker oraz Git
- Wygenerowany token do API USOS PW <https://apps.usos.pw.edu.pl/developers/> i zapisane id i klucz aplikacji

Instrukcja - step-by-step:

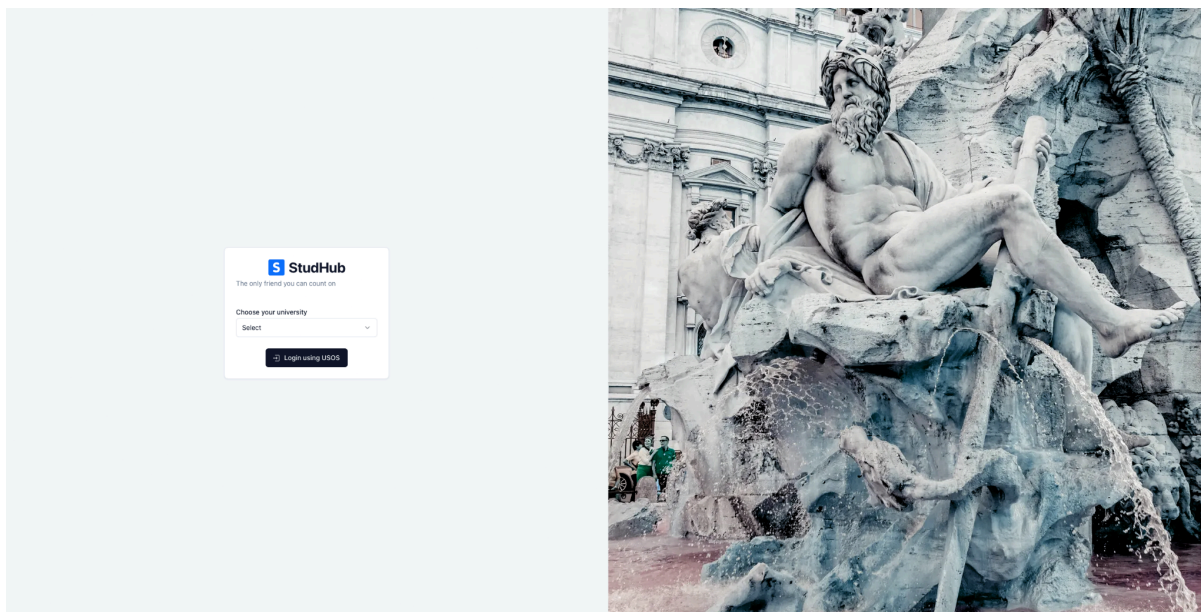
1. Pobranie kodu z repozytorium <https://github.com/AdamJ77/studhub>
2. Ustawienie w pliku `.env.dev` wartości zmiennych `USOS_CLIENT_ID` oraz `USOS_CLIENT_KEY` zgodnie z wartościami zapisanymi podczas generowania tokenu do API USOS PW
3. Start aplikacji:
 - uruchomić skrypt budujący aplikację:
`./docker-compose.sh build dev`
 - uruchomić skrypt uruchamiający aplikację:
`./docker-compose.sh run dev`

Opis użytkowania:

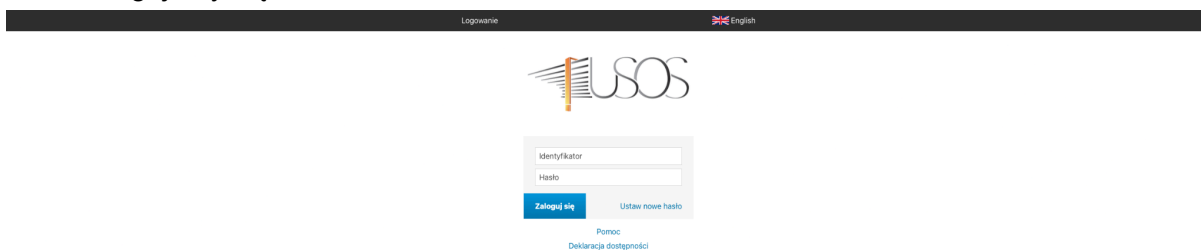
Klient: Student Politechniki Warszawskiej posiadający konto w systemie USOS

Logowanie:

1. Wchodzimy na localhost
2. Klikamy przycisk "Login using USOS"
 - a. Następuje przekierowanie na stronę systemu USOS



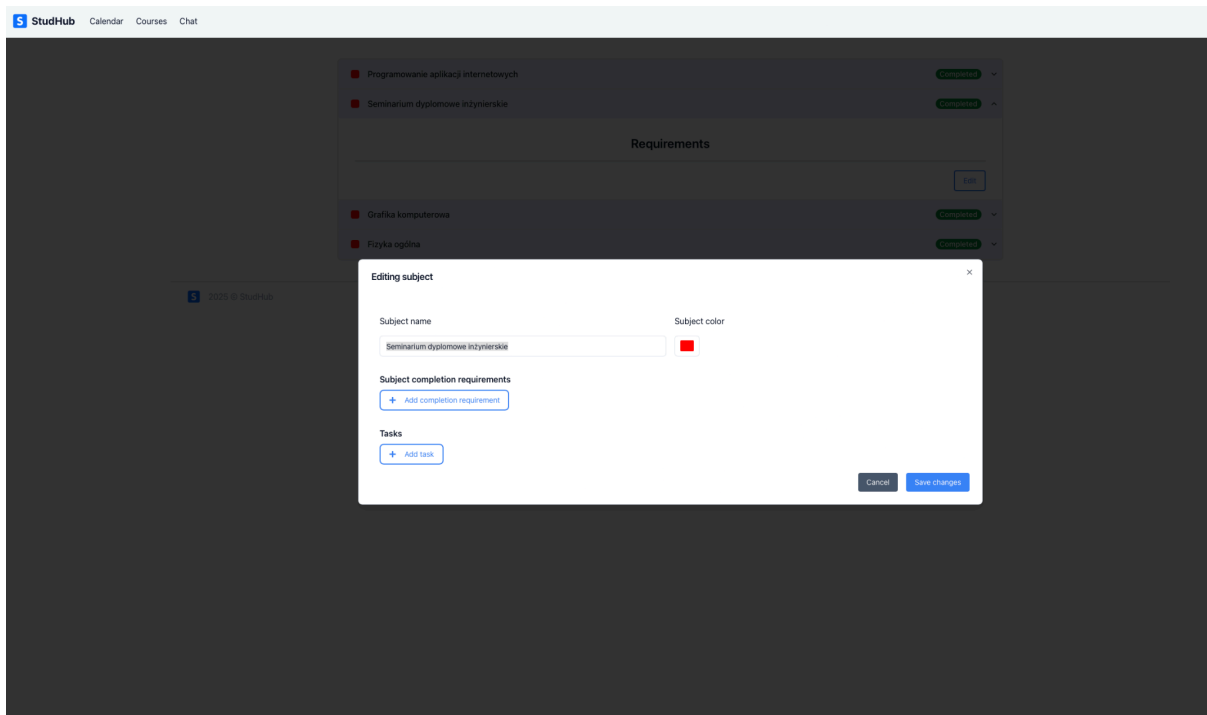
3. Logujemy się na swoim koncie USOS



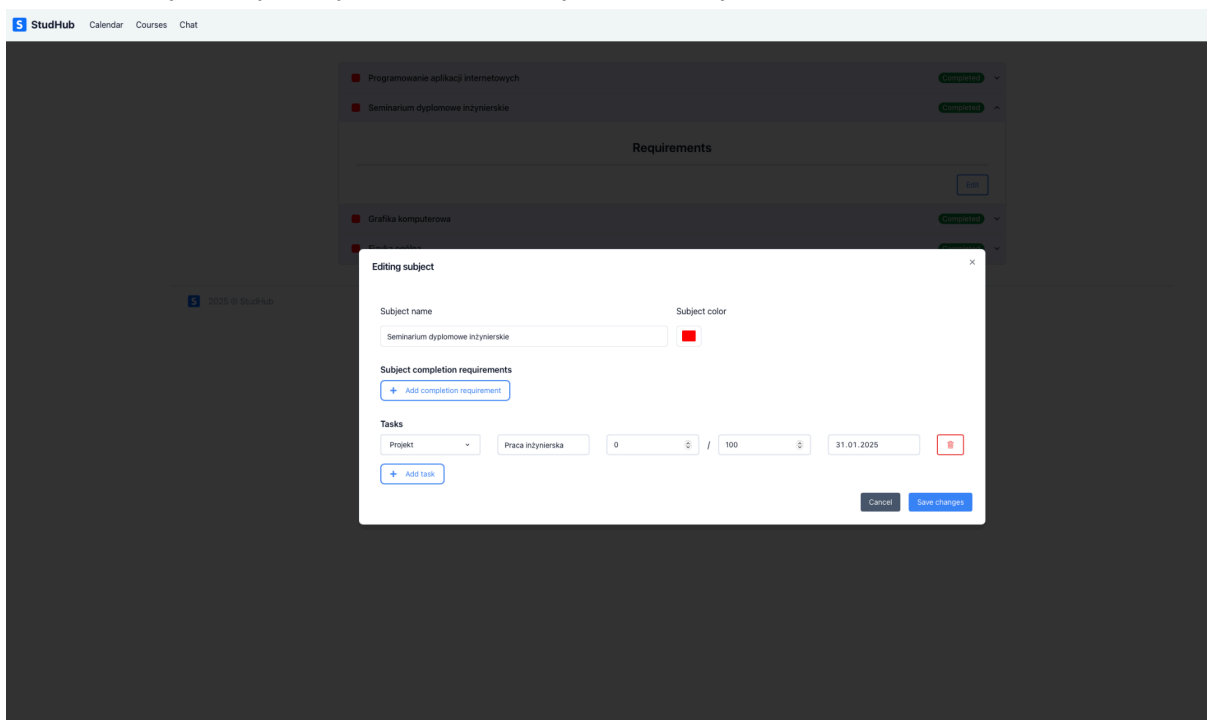
4. Wyrażamy zgodę na przesyłanie danych do aplikacji (wymagane tylko przy pierwszym logowaniu)
 - a. Zostajemy przekierowani na stronę portalu StudHub

Dodawanie zaliczeń do przedmiotu:

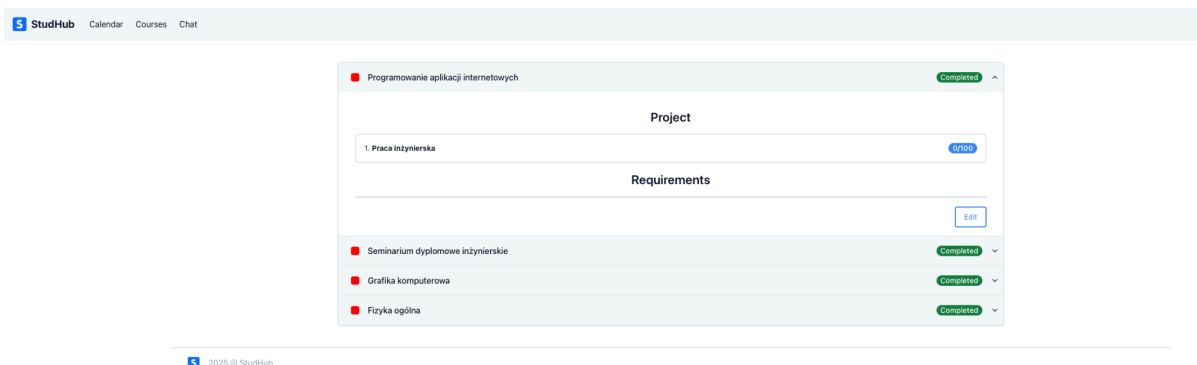
1. Wchodzimy w zakładkę *Courses*
 - a. Wyświetlona zostaje lista przedmiotów, na które jesteśmy zapisani
2. Klikamy na interesujący nas przedmiot
 - a. Rozwijany jest panel ze szczegółowymi informacjami
3. Klikamy przycisk *“Edit”*
 - a. Pokazuje się okno edycji przedmiotu



4. Klikamy przycisk “Add task”
 - a. Pojawia się nowy wiersz z nowo dodanym zaliczeniem
5. Wybieramy typ zaliczenia, nazwę zaliczenia, opcjonalnie ilość zdobytych punktów, maksymalną liczbę punktów możliwych do zdobycia oraz termin zaliczenia

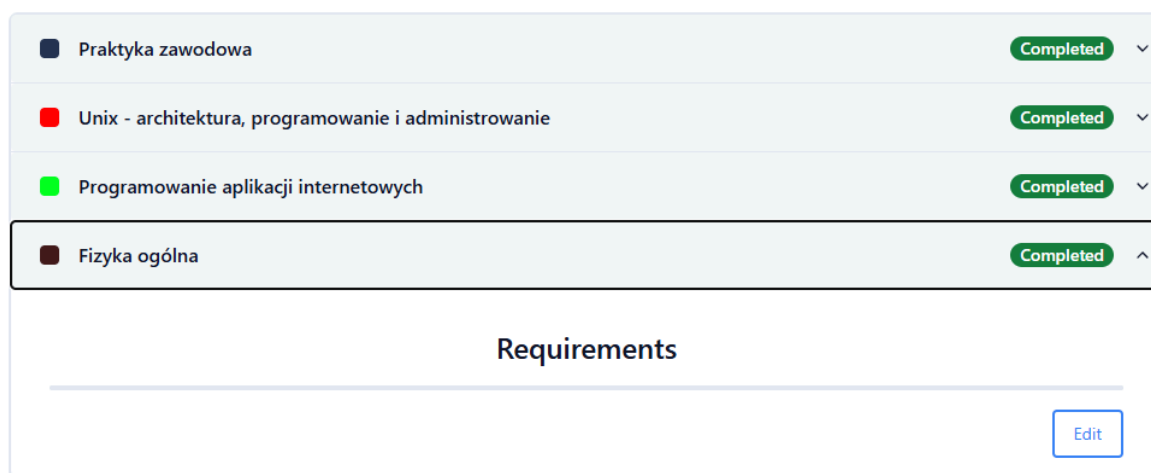


6. Klikamy przycisk “Save changes”



Dodawanie wymagań do zaliczenia przedmiotu:

1. Wchodzimy w zakładkę *Courses* i klikamy interesujący nas przedmiot



2. Klikamy przycisk *“Edit”*

Editing subject

Subject name

Fizyka ogólna

Subject color

Subject completion requirements

+ Add completion requirement

Tasks

+ Add task

Cancel

Save changes

- Klikamy przycisk *"Add completion requirement"*
- Wybieramy rodzaj zaliczenia związany z danym wymaganiem, wartość progu, który potrzebny jest do zaliczenia wymagania, rodzaj progu (punktowy / procentowy), rodzaj wymagania (z sumy wszystkich zaliczeń / każdego zaliczenia z osobna)

StudHub

Calendar

Courses

Chat

Programowanie aplikacji internetowych

Project

1. Praca inżynierska

Requirements

The sum of all tasks of the type: Projekt must be at least: 50 points

Editing subject

Subject name

Programowanie aplikacji internetowych

Subject color

Subject completion requirements

Projekt

50

Points

Total

+ Add completion requirement

Tasks

Projekt

Praca inżynierska

0

/

100

31.01.2025

+ Add task

Cancel

Save changes

- Klikamy przycisk *"Save changes"*. W przypadku, gdy nie mamy obecnie spełnionych wymagań do zaliczenia przedmiotu, dany przedmiot jest oznaczony na czerwono jako *"Failed"*.

StudHub

Calendar

Courses

Chat

Programowanie aplikacji internetowych

Project

1. Praca inżynierska

Requirements

The sum of all tasks of the type: Projekt must be at least: 50 points

0%

Edit

Seminarium dyplomowe inżynierskie

Completed

Grafika komputerowa

Completed

Fizyka ogólna

Completed

6. Przy rozwinięciu przedmiotu z określonymi wymaganiami zaliczenia, mamy wartość wyświetlony procent spełnienia wszystkich wymagań

StudHub Calendar Courses Chat

Programowanie aplikacji internetowych Completed

Project

1. Praca inżynierska 50/100

Requirements

The sum of all tasks of the type: Projekt must be at least: 50 points

100%

Edit

Seminarium dyplomowe inżynierskie Completed

Grafika komputerowa Completed

Fizyka ogólna Completed

2025 © StudHub

Zmiana koloru przedmiotu:

1. Wchodzimy w zakładkę *Courses* i klikamy interesujący nas przedmiot

Praktyka zawodowa Completed

Unix - architektura, programowanie i administrowanie Completed

Programowanie aplikacji internetowych Completed

Fizyka ogólna Completed

Requirements

Edit

2. Klikamy przycisk *"Edit"*

Editing subject

Subject name

Fizyka ogólna

Subject color

Subject completion requirements

+ Add completion requirement

Tasks

+ Add task

Cancel

Save changes

- Klikamy pole z kolorem przedmiotu
- Wybieramy interesujący nas kolor

Programowanie aplikacji internetowych

Completed

Editing subject

Subject name

Fizyka ogólna

Subject color

Subject completion requirements

Laboratorium

0

+ Add completion requirement

Tasks

+ Add task

Color picker

158

251

65

R

G

B

Total

Cancel

Save changes

- Klikamy przycisk "Save changes"

Praktyka zawodowa

Completed

Unix - architektura, programowanie i administrowanie

Completed

Programowanie aplikacji internetowych

Completed

Fizyka ogólna

Completed

Requirements

The sum of all tasks of the type: **Laboratorium** must be at least: 0 points

NaN%

Edit

Przeglądanie dodanych zaliczeń w kalendarzu:

1. Wchodzimy w zakładkę *Calendar*

Previous Month

styczeń 2025

Next Month

Mon	Tue	Wed	Thu	Fri	Sat	Sun
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27 Fizyka ogólna	28	29 Programowanie aplikacji	30	31	1	2

2. Za pomocą przycisków “*Previous Month*” i “*Next Month*” wybieramy interesujący nas miesiąc
3. Wybieramy interesujący nas dzień

Previous Month

styczeń 2025

Next Month

Mon	Tue	Wed	Thu	Fri	Sat	Sun
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27 Fizyka ogólna	28	29 Programowanie aplikacji	30	31	1	2

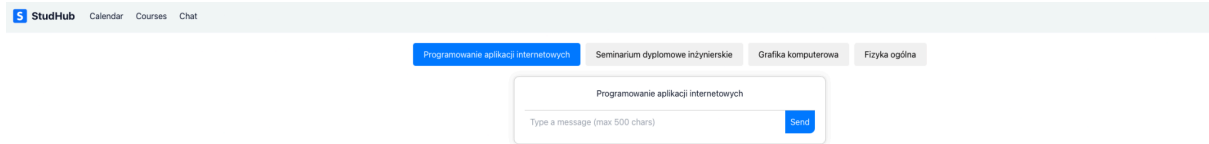
Selected Date: Mon Jan 27 2025

Events on Selected Date:

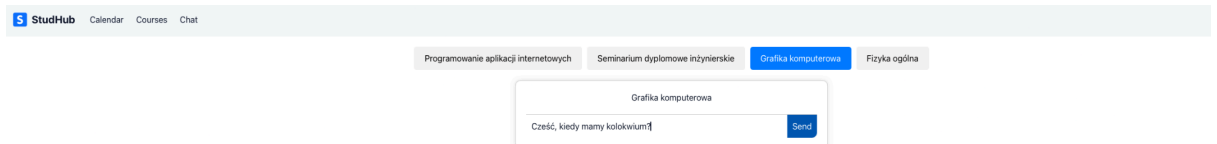
Fizyka ogólna - Laboratorium

Komunikacja z innymi uczestnikami przedmiotu:

1. Wchodzimy w zakładkę *Chat*
2. Wyświetla nam się lista osobnych czatów grupowych dla każdego podpętego przedmiotu



3. Wybieramy czat dla danego przedmiotu klikając w jego nazwę
4. W wolnym polu oznaczonym "*Type a message*" wpisujemy wiadomość i klikamy przycisk *Send*



5. Nasza wiadomość (oznaczona kolorem niebieskim) wyświetla się na czacie

S StudHub

Calendar

Courses

Chat

Welcome, Adam Zbigniew Jez!

Programowanie aplikacji internetowych

Seminarium dyplomowe inżynierskie

Grafika komputerowa

Fizyka ogólna

Grafika komputerowa

Adam Zbigniew Jez: Cześć, kiedy mamy kolokwium?

Type a message (max 500 chars)

Send