# Aplikacje internetowe [PROJEKT]

GR02:

Tomasz Ciuła

Rafał Klepacz

Lucjan Bąkowski

## Aplikacja BookApp

Prosta aplikacja do zarządzania domową biblioteczką.

Na podstawową funkcjonalność składają się konteksty:

* użytkownika, w ramach którego można:
  + zarejestrować użytkownika przy pomocy adresu email,
  + zalogować użytkownika przy pomocy adresu email
* biblioteki, dający możliwość:
  + dodawania/aktualizacji/usuwania:
    - książek,
    - autorów,
    - wydawnictw
  + filtrowania listy:
    - książek po tytule, autorze, nr wydania, roku wydania,
    - autorów po imieniu i nazwisku,
    - wydawnictw po nazwie

Książka zawiera:

* autor (jeden lub kilku) [wymagane],
* wydawnictwo [wymagane],
* tytuł [wymagane],
* rok wydania [opcjonalne],
* nr wydania [opcjonalne],
* komentarz użytkownika [opcjonalne]

Wydawnictwo zawiera:

* nazwa [wymagane]

Autor zawiera:

* imię [wymagane],
* nazwisko [wymagane],

Na aplikację składać się będzie część kliencka (frontend) oraz część serwerowa (backend).  
Część kliencka aplikacji zrealizowana będzie w języku JavaScript lub TypeScript przy pomocy biblioteki React lub Next.js, natomiast część serwerowa będzie zrealizowana w języku Python posługując się biblioteką DRF (Django REST Framework). Do stylowania aplikacji klienckiej posłużą nam biblioteki: tailwinds lub styled-components w metodologii atomic design. Backend będzie działał w postaci REST API, na którego udostępnione endpointy, będzie wysyłała żądania aplikacja kliencka. Za bazę danych w projekcie posłuży nam Sqlite. Frontend hostowany będzie wykorzystując usługę Netlify, natomiast backend wykorzystywać będzie usługę Heroku. Do pracy zespołowej jako system kontroli wersji posłuży nam Git.

Stack technologiczny:

* React lub Next.js (frontend),
* Tailwinds/styled-components (css styling),
* Django REST Framework (backend),
* Sqlite (db),
* Netlify (frontend hosting),
* Heroku (beckend hosting),
* Git (version control system)