# Django

## Projekt końcowy

### Podstawowe założenia

* Projekt realizujemy w grupach (2-3 osoby)
* Programujemy zarówno na zajęciach jak i w domu
* Kod przechowujemy na github. Regularnie commitujemy zmiany. Dodajemy adamkar26 jako recenzenta do repozytorium
* Termin oddania projektu: 8 stycznia
* Każdy projekt prezentujemy na forum
* Ocena z projektu:
  + Ocena za jakość projektu: waga 5 (sprawdzian)
  + Ocena za czystość kodu: waga 3 (kartkówka)
  + Ocena za prezentację projektu: waga 1 (aktywność)

### Kamienie milowe

* Wybór tematu, opis projektu:
  + Wymagania
  + Zakres projektu (co chcemy zrobić?)
* Opis bazodanowy (modeli)
* Programowanie widoków, formularzy, templetów itd..

### Wymagane elementy

* Na ocenę dopuszczający:
  + Jeden model
  + Trzy widoki
  + Jeden formularz
  + Bootstrap
  + Estetyka
* Na ocenę dostateczny:
  + Logowanie/wylogowywanie użytkownika
  + Cztery widoki
  + Dwa formularze
* Na ocenę dobry:
  + Rejestracja użytkownika (lub inne rzeczy związane z użytkownikami)
  + Użycie przesyłania plików (np. dodawanie zdjęcia użytkownika)
  + Użycie JavaScript
* Na ocenę bardzo dobry:
  + Użycie JQuery
  + Kompleksowe rozwiązanie problemu
* Na ocenę celujący:
  + Wdrożenie aplikacji

### Proponowane tematy

* Help desk informatyczny:
  + Strona do zgłaszania usterek z podziałem na kategorie (w wersji podstawowej nie wymaga logowania). Przykładowe kategorie: usterka komputera, drukarki, serwisu itp.
  + Informatyk po zalogowaniu widzi wszystkie zgłoszenia posortowane według czasu
  + Każde zgłoszenie ma status: oczekujące, w trakcie, rozwiązane
  + Wersje zaawansowane: logowanie użytkowników (innych niż informatycy), więcej niż jeden informatyk i przypisywanie do nich zadań, priorytety zadań
* Zgłoszenia do konserwatora:
  + Strona do zgłaszania zadań konserwatorowi np. powiesić tablicę itp
  + Użytkownik nadaje zgłoszeniu wagę
  + Użytkownik może przesłać zdjęcie jako załącznik
  + Konserwator po zalogowaniu widzi wszystkie zgłoszenia posortowane według czasu
  + Każde zgłoszenie ma status: oczekujące, w trakcie, rozwiązane
  + Wersje zaawansowane: logowanie użytkowników, więcej niż jeden konserwator, system punktów dla użytkowników (do omówienia)
* System zapisów na poprawy:
  + Zalogowany nauczyciel wprowadza swój plan zajęć (od poniedziałku do piątku) i liczbę dostępnych miejsc w sali
  + Uczeń rejestruje się na poprawę – wybiera datę i godzinę, podaje materiał który chce poprawiać. Uwaga poprawa może trwać więcej niż jedną godzinę
  + System pozwala na rejestrację tylko jeśli są dostępne miejsca
  + Nauczyciel na bieżąco widzi zapisy
  + Wersje zaawansowane: logowanie uczniów (może OAuth2?), każda poprawa musi być zaakceptowana przez nauczyciela, uczeń widzi czy jego poprawa jest zaakceptowana, możliwość importu planu z pliku CSV i/lub ze strony tebwroclaw.edupage.org
* System kolejkowy:
  + Strona do generowania numerków w kolejce. Może istnieć więcej niż jedna kolejka
  + Strona wyświetlająca aktualnie obsługiwany numer (dla każdej kolejki osobna strona)
  + Strona wyświetlająca jakie numerki obsługiwane są przy danym stanowisku
  + Panel administratora (logowanie): dodawanie kolejek, ustalanie liczby stanowisk dla danej kolejki, podgląd statystyk
  + Wersje zaawansowane: optymalizacja kolejek np. jedno stanowisko nie musi obsługiwać tylko jednej kolejki, import spisu stanowisk z pliku