Ogólne założenia projektów:

1. Projekt nie ma progu zaliczenia, chociaż stanowi 50% oceny. Jeśli ktoś uważa, że zaliczy przedmiot poprzez laboratoria i kolokwium, nie musi realizować projektu.
2. Projekt realizuje każdy osobno.
3. Temat projektu wraz z opisem powinien znaleźć się w Excelu z repozytoriami w osobnym arkuszu (do czasu oddania projektu).
4. Projekt oddajemy na dowolnych zajęciach, oprócz przedostatnich (kolokwium). Ostateczny termin oddania projektów to ostatnie zajęcia.
5. W dniu oddawania projekt powinien działać i być zdatny do sprawdzenia.
6. Każdy będzie miał 10 minut na prezentację (po 5 minut na prezentację wizualną i funkcjonalności oraz prezentację kodu + ewentualne pytania do kodu i delikatne zmiany w kodzie).
7. Wymagania projektu podane niżej. Można wybrać to, co się realizuje z wymagań, kosztem nie uzyskania punktów za pozostałe wymagania.
8. Wymagania z oznaczeniem *(dodatkowo)* nie są obowiązkowe i nie stanowią podstawowej części punktacji (są ponad próg). Można zamienić wymaganie podstawowe na wymaganie dodatkowe, jednak jeśli podstawowe wymaganie jest na 2 pkt., a dodatkowe na 1 pkt., to otrzymać można 1 pkt, w odwrotnej korelacji podstawowe 1 pkt., dodatkowe 2pkt., również otrzyma się 1 pkt.
9. Za dodatkowe zadania można uzyskać ograniczoną ilość punktów, nawet jeśli zrealizuje się więcej dodatkowych zadań. Chyba, że realizacja zadania dodatkowego służy “wymianie” za podstawowe.
10. W razie pytań, czy konsultacji zapraszam do kontaktu mailowego.
11. Jeśli projekt rozwiązuje rzeczywisty problem i jest mocno zaawansowany, to jest możliwość uzyskania dodatkowych punktów - mocno uznaniowe punkty.

Wymagania ogólne (funkcjonalne, wizualne i techniczne):

1. użycie formularzy i ich funkcjonalności - odbieranie i przetwarzanie danych
2. zapisywanie do pliku danych - np. zrzut bazy danych, zapis danych z formularza, zapis przesłanych plików itp.
3. zapisywanie do bazy danych (mysql)
4. odczytywanie z bazy danych
5. aktualizowanie i usuwanie z bazy danych
6. prosty system logowania z weryfikacją adresu e-mail
7. użycie sesji w projekcie, nie sztuczne, tylko takie by pozwalało realnie zobrazować ich funkcjonalność
8. użycie ciasteczek - utworzenie oraz realne skorzystanie z nich
9. użycie pętli, instrukcje warunkowe, tablice, funkcji
10. użycie programowania obiektowego - logiczny podział na klasy
11. *(dodatkowo)* użycie czegoś zaawansowanego z PHP, co nie było pokazywane na zajęciach - biblioteka, narzędzie, konstrukcja, temat (wymagane omówienie)
12. *(dodatkowo)* użycie czegoś, co nie było pokazywane na zajęciach (dodatkowy język, skrypty) - biblioteka, narzędzie, konstrukcja, temat (wymagane krótkie omówienie)
13. hostowanie strony na uczelnianym serwerze
14. *(dodatkowo)* hostowanie na zewnętrznym serwerze

Wymagania niefunkcjonalne:

1. Przejrzysty kod - np. rozbicie na pliki, klasy, metody, stosowanie pętli zamiast printowania na sztywno
2. Brak widocznych błędów komunikowanych przez PHP
3. Projekt się “kompiluje” - brak problemów przy oddawaniu projektu
4. Repozytorium - commity, porządek etc.

Każdy projekt ma również założenia co do funkcjonalności projektu, które zdefiniowane są w opisie danego projektu. Funkcjonalności można zmieniać i modyfikować po uprzednim kontakcie z prowadzącym.

Rozkład punktacji znajduje się pod [LINKIEM](https://docs.google.com/spreadsheets/d/1N5988C5e0JBmUUr5RKxy8o8JICcSvjNkHXRWNi2AkCY/edit?usp=sharing).