Do wykonania jest witryna internetowa, symulująca rzucanie zwykłymi kośćmi do gry. Witryna jest zbudowana zgodnie ze standardami HTML5. Należy wykorzystać zarówno język JS (do generowania wyników rzutów, wpisywania ich na stronie, wyświetlania odpowiednich obrazów kości w zależności od wygenerowanego wyniku oraz dekorowania strony w odpowiednim momencie) oraz PHP (do zapisania gracza oraz wyświetlenia danych ostatnio zarejestrowanego gracza)

# Baza danych

Należy przygotować bazę danych o nazwie „**kosci**”, w której znajdzie się jedna tabela „**gracze**”. Mają w niej być kolumny:

* ID\_gracza – całkowitoliczbowy, automatyczna inkrementacja,
* imie – napisowy, długość maksymalna – 50 znaków, domyślna wartość „Gracz”
* wiek – całkowitoliczbowy, wymagany, baza musi sprawdzać, czy gracz ma 15 lub więcej lat
* data\_rejestracji – typ TIMESTAMP

Do tej tabeli w bazie będą dodawane rekordy poprzez stronę.

# Układ strony

* Stronę „**index.php**” należy przygotować stosując język HTML5, jawnie zadeklarować język polski jako język zawartości strony i zastosować odpowiednie kodowanie znaków
* Do strony należy dołączyć zewnętrzny arkusz stylów CSS „**kosci.css**”
* Tytuł strony, wyświetlany w pasku zakładek, ustawić na „Rzuty kością”
* Strona podzielona na sekcje z zastosowaniem semantycznych znaczników blokowych:
  + W bloku banneru – nagłówek pierwszego stopnia o treści „Symulator rzutów kością”
  + W bloku głównym:
    - Nagłówek drugiego stopnia o treści „Kliknij na kości, by nimi rzucić”.
    - Nagłówek trzeciego stopnia o treści „Zdobywasz punkt, gdy suma oczek wynosi 7”,
    - Blok, do którego przypisana jest funkcja opisana w **skrypcie JS1**. W tym bloku:
      * Akapit z napisem „Wynik:”, a wewnątrz akapitu liniowy znacznik <span> z wartością „2”.
      * Dwa obrazy „kosc1.png” obok siebie.
  + W bloku pobocznym:
    - Nagłówek drugiego stopnia o treści „Zarejestruj gracza”:
    - Formularz, korzystający z niejawnej metody przesyłania danych, wysyłający dane na tę samą stronę (na index.php). Każde pole formularza musi być odpowiedniego typu i posiadać etykietę <label>:
      * Pole na wpisanie imienia gracza
      * Pole na wpisanie wieku gracza. Gracz musi mieć minimum 15 lat (trzeba ograniczyć minimalną dopuszczalną wartość do wpisania w to pole)
      * Przycisk przesyłający formularz
    - Pod formularzem należy umieścić skrypt **PHP1**
    - Nagłówek trzeciego stopnia o treści „Ostatni zarejestrowany gracz:”
    - Lista nieuporządkowana:
      * Punkt o treści „Imię:” i wynik działania skryptu **PHP2**
      * Punkt o treści „Wiek:” i wynik działania skryptu **PHP3**
      * Punkt o treści „Data rejestracji:” i wynik działania skryptu **PHP4**
  + W bloku stopki:
    - Nagłówek czwartego stopnia o treści „Symulator wykonał:” z podpisem autora strony

# Styl strony

* Do wszystkich bloków: krój czcionki Courier New, marginesy wewnętrzne – 0, marginesy zewnętrzne – 0 (resetowanie domyślnych stylów przeglądarki)
* Wspólnie dla bloków bannera i stopki: środkowanie tekstu, kolor tła – złoty
* Dla bloku bannera: tekst wyśrodkowany w pionie i w poziomie (należy użyć flex), szerokość – 75%, wysokość – 100 pikseli, margines wewnętrzny – 1%
* Dla bloku głównego: kolor tła – „goldenrod”, szerokość – 75%, wysokość – 400 pikseli, margines wewnętrzny – 1%
* Dla bloku bocznego: szerokość 20%, wysokość 500 pikseli, margines wewnętrzny – 1%
* Dla pól edycyjnych formularza: szerokość – 50 pikseli, obramowanie – linia przerywana, złota o grubości 2 pikseli
* Dla przycisków formularza: szerokość – 200 pikseli, margines zewnętrzny – 10 pikseli
* Dla elementów listy: margines wewnętrzny – 2 piksele, punktory listy – wyłączone (list-style-type), po najechaniu na element listy: kolor tła – goldenrod, kolor tekstu – biały. Przejście między stanami powinno być płynne (transition)
* Blok ze skryptem JS i obrazami kości: blok wyśrodkowany poprzez marginesy zewnętrzne, szerokość – 300 pikseli, wysokość – 200 pikseli, obrazy w środku wyśrodkowane, rozmieszczone przez flex
* Dla obrazów: margines zewnętrzny – 10 pikseli

# Skrypty

* Skrypt JS1 – wykonywanie rzutu dwiema kośćmi. Przypisany do bloku przechowującego obrazy z kośćmi, wykonujący się przy kliknięciu:
  + Wylosuj dwie wartości z zakresu od 1 do 6 i przypisz je do zmiennych (np. pierwszaKosc, drugaKosc)
  + W zależności od wylosowanych wartości, zmień wyświetlany na stronie obrazek. Przykładowo, jeżeli pierwszaKosc = 3, a drugaKosc = 4, to na stronie powinny pokazywać się obrazy „kosc3.png” i „kosc4.png”
  + W znaczniku <span>, umieszczonym nad obrazami kości, należy wpisać sumę obydwu losowań (zgodnie z przykładem powyżej, w tym bloku powinno być napisane „7”)
  + Jeżeli suma losowań wynosi 7, należy zmienić tło bloku z kośćmi na zielone. W każdym innym przypadku tło powinno pozostawać w domyślnym kolorze.
* Skrypt PHP1 – pobieranie wartości z formularza rejestrującego nowego gracza i umieszczenie jej w bazie danych:
  + Przed wykonaniem reszty skryptu sprawdź, czy pola na imię gracza i jego wiek mają wpisaną wartość (czy nie są puste). Jeżeli tak, przejdź do reszty skryptu
  + Pobierz wartości z pól formularzy na imię i wiek, przypisz je do zmiennych
  + Wykonaj zapytanie wstawiające imię i wiek gracza do tabeli „gracze” w bazie danych „kosci”.
* Skrypt PHP2 – pobierz z bazy i wyświetl imię gracza, który zarejestrował się ostatnio
* Skrypt PHP3 – pobierz z bazy i wyświetl wiek gracza, który zarejestrował się ostatnio
* Skrypt PHP4 – pobierz z bazy i wyświetl datę rejestracji gracza, który zarejestrował się ostatnio

# Przydatne funkcje

## JavaScript

* document.getElementById() – zwraca obiekt o danym id
* Math.floor(Math.random() \* (max – min)) + min – losuje wartość z przedziału <min, max>
* element.src – pozwala zmienić źródło obrazu (jeżeli element jest obrazem)

## PHP

* mysqli\_connect(serwer, użytkownik, hasło użytkownika, baza danych) – łączy z bazą danych
* isset() – czy wartość istnieje
* empty() – czy zmienna jest pusta
* mysqli\_query(połączenie, zapytanie) – wykonuje zapytanie do danej bazy
* mysqli\_fetch\_row(tabela wynikowa) – pobiera pojedynczy wiersz z tabeli wynikowej jako lista numeryczna