

# Języki programowania i GUI

## Lista 4 - 2021

1. (2pkt) (a) Utwórz nowy folder, uruchom terminal wewnątrz i komendą `npm init` stwórz metadane Twojej aplikacji webowej. Wpisz nazwę aplikacji, swoje imię i nazwisko jako autora. Plikiem startowym niech będzie `index.js`. Napisz używając tylko modułu `http` serwis o trzech podstronach, które się nawzajem do siebie odwołują. W przypadku wpisania innego adresu `url` powinna się pokazywać informacja o błędzie 404.  
(b) Dodaj do serwisu dwa pliki (obrazek i dokument pdf), a odwołania do nich w znacznikach `img` oraz `A` zawrzyj na stronie głównej. Zadbaj o to, by po wpisaniu ich nazw jako adresu `url`, Twój serwer wysyłał do klienta, odpowiedni nagłówek `content-type`, oraz odczytaną za pomocą `fs.readFileSync()` ich zawartość.  
(c) Powtórz punkt (b) wykorzystując `fs.readFileAsync()`.
2. (2pkt) Do przykładu aplikacji `mini3` z wykładu dodaj podgląd uczestników czatu. Aby tego dokonać po stronie serwera utrzymuj obiekt zawierający identyfikatory, i nicki zalogowanych klientów. `id` klienta jest nadawany przez serwer i wysyłany do klienta zaraz po utworzeniu gniazdka (`socket`). `nick` jest wysyłany przez klienta do serwera zaraz po uruchomieniu połączenia na kanale `nick`.  
  
Listę należy aktualizować i rozsyłać do wszystkich na osobnym kanale (np `"user list"`) po podłączeniu nowego klienta, oraz po rozłączeniu (`on("disconnect")`). Dla uproszczenia `id` klienta (i ewentualnie jego `nick`) można też przechowywać w obiekcie `socket`.  
  
Po stronie klienta reakcją na otrzymanie wiadomości na tym "user list" powinno być umieszczenie otrzymanego spisu w osobnym elemencie `ul` znajdującym się obok czatu.  
  
Jako format danych można przyjąć (a) JSON zawierający zakodowany obiekt ze spisem klientów, lub też (b) gotowy do wklejenia na stronę kod `html` zawierający wypunktowanie imion klientów. Wypróbuj oba podejścia.
3. (3pkt) Wzorując się (lub nie) na przykładzie `mini6` napisz grę online własnego pomysłu, dla dwóch lub więcej graczy.