

# TIN – Komunikacja frontend- backend (10pkt)

---

Zaimplementuj poniższe wymagania. Zadanie należy wykonać przy pomocy czystego Node.js lub Express.js (w zadaniu **nie należy** korzystać z szablonów EJS). Zadbaj o odpowiednią obsługę błędów oraz odpowiednie wyświetlanie czytelnych komunikatów użytkownikowi (**nie należy** korzystać z funkcji `alert`). Użyj własnej inwencji do uzupełnienia „brakujących” elementów wymagań. Implementacje wszystkich wymagań należy przedstawić w ramach jednej aplikacji.

- 1) Aplikacja powinna serwować formularz (statyczne pliki HTML, CSS, JS) pozwalający na dodawanie nowych danych. Formularz powinien zawierać przynajmniej 3 pola różnych typów. Formularz przed wysłaniem powinien zostać zwalidowany przy pomocy JS. Jeśli dane przejdą walidację powinny zostać wysłane asynchronicznie na backend (np. `POST /users`). Jeśli dane nie przejdą walidacji wyświetlamy odpowiedni komunikat. Na backendzie należy ponownie wykonać na tych danych walidację. Jeśli dane przejdą walidację powinny zostać one „zapisane” po stronie backendu (dane można przechowywać w globalnej zmiennej. Jeśli dane nie przejdą walidacji należy odesłać odpowiedni komunikat do frontendu, a następnie wyświetlić informację użytkownikowi. **(4pkt)**
- 2) Aplikacja powinna serwować listę (statyczne pliki HTML, CSS, JS) pozwalającą na wyświetlenie wszystkich dodanych danych. Po załadowaniu statycznej strony należy wysłać asynchroniczne zapytanie na backend (np. `GET /users`), które powinno zwrócić wszystkie dane - które później należy wyświetlić po stronie frontendu (np. w formie tabeli) poprzez dynamiczną modyfikację struktury HTML. **(3pkt)**
- 3) Rozszerz statyczną stronę z poprzedniego podpunktu o możliwość usuwania poszczególnych, wybranych przez użytkownika rekordów tabeli. W tym celu z frontendu powinno zostać wysłane asynchroniczne zapytanie na backend (np. `DELETE /users/4`). Jeżeli obiekt o danym ID istnieje, powinien on zostać usunięty z listy. Po usunięciu obiektu (uzyskaniu odpowiedzi z API) należy zaktualizować listę z danymi (np. poprzez ponowne pobranie danych). **(3pkt)**