**Node.js**

Robimy zadania „The Learning Path Challenge” z https://forum.sealcode.org/t/the-learning-path-challenge/1807

**Zadanie 1 (1pkt)  
a) krok 2. A small project with colors**Utwórz małą aplikację JS przy użyciu node.js. Użyj npm, aby zainstalować bibliotekę chalk i użyj jej, aby dane wyjściowe programu node.js były kolorowe. Aplikacja powinna działać w terminalu (cli) i wyświetlać liczby od 1 do 100. Liczby podzielne przez 3 powinny być czerwone, liczby podzielne przez 5 powinny być niebieskie, a liczby podzielne przez 3 i 5 powinny być fioletowe. Wszystkie pozostałe liczby powinny być białe. Spraw, aby można było uruchomić aplikację za pomocą polecenia npm start, wprowadzając zmiany w pliku package.json.

**b) krok 4. A small web application with koa**Utwórz małą aplikację internetową, która obsługuje dwie strony: 127.0.0.1:8080/hello i 127.0.0.1:8080/goodbye. Jedna z nich powinna mówić "Hello, world", a druga "Goodbye, world".

Tutorial: https://www.digitalocean.com/community/tutorials/how-to-build-a-hello-world-application-with-koa

**Zadanie 2 (1pkt)**

**HTML Forms – kroki 7, 8, 9**

Utwórz plik login.html zawierający formularz logowania. Zawiera on dwa pola - nazwę użytkownika i hasło. Każde pole powinno mieć klikalną etykietę (tak jak w kroku learning path). Formularz powinien mieć przycisk przesyłania "Zaloguj się". Nazwa użytkownika i hasło powinny być wymagane. Hasło powinno mieć co najmniej 8 znaków - jeśli użytkownik spróbuje przesłać krótsze hasło, przesłanie powinno zostać uniemożliwione i powinien zostać wyświetlony odpowiedni komunikat o błędzie. Nie używaj żadnych skryptów JavaScript w tym pliku HTML. Wszystkie opisane powyżej funkcje można osiągnąć za pomocą czystego HTML.

W tym celu : Utwórz małą aplikację koa, dodaj dwa endpointy: GET /login i POST /login. GET /login powinien po prostu odpowiedzieć kodem HTML formularza, POST /login powinien odpowiedzieć kodem HTML zawierającym komunikat "zalogowany!", jeśli użytkownik jest zalogowany, "Nieprawidłowa nazwa użytkownika lub hasło" w przeciwnym razie. Jedynymi działającymi poświadczeniami powinny być login: admin i hasło: adminadmin. (w formularzu upewnij się, że atrybut action elementu <form> wskazuje na /login, a metoda to POST).

Walidacja hasła powinna zostać przeprowadzona zarówno po stronie front-endu, jak i back-endu. Jeśli użytkownik spróbuje przesłać hasło, które jest zbyt krótkie, przeglądarka powinna wyświetlić błąd bez przesyłania formularza, jeśli użytkownik obejdzie to zabezpieczenie i prześle formularz, serwer powinien odpowiedzieć "to hasło jest zbyt krótkie".

Przydatne → https://github.com/koajs/koa-body

**Zadanie 3 (1pkt)**

# **HTTP request method & query params**

Zmodyfikuj stronę z poprzedniego zadania lub utwórz nową, która będzie wykorzystywała pozostałe metody zapytań HTTP – przy każdej z nich może zwrócić prosty komunikat, np. „odebrałem polecenie usuń”. Dodaj funkcjonalność, która będzie odpowiadała na podany parametr, np. uzupełniała login użytkownika w formularzu po ?login=”abc” lub wyświetlała komunikat z parametrem.

Przetestuj stronę (metody HTTP) za pomocą Developer Tools w przeglądarce i za pomocą curl (dodatkowo można za pomocą wtyczki Thunder Client w VC Code) i zrób skrinszoty.

Na 25.05

[mat.jarcz@gmail.com](mailto:mat.jarcz@gmail.com)