

1. Stwórz servlet, który zwróci zawartość pliku:
 - 1.1. Wczytaj plik `users.txt` (`[first_name],[last_name]`) do listy;
 - 1.2. Utwórz klasę `UsersServlet` i zaimplementuj metodę `doGet()`, aby wyświetliła użytkowników w postaci listy ``;
 - 1.3. Zmapuj servlet na adres `/users`.
2. Stwórz formularz dodawania nowego użytkownika:
 - 2.1. Stwórz statyczną stronę `usersform.htm`, w której znajdzie się struktura formularza;
 - 2.2. Utwórz klasę `AddUserServlet` i w metodzie `doGet()` dodaj użytkownika do listy, wykorzystywanej przez `UsersServlet`. Zastosuj wzorzec Singleton, tak, aby oba serlety współdzieliły jedną listę;
 - 2.3. Gdy użytkownika uda się dodać do listy, zwróć odpowiedź ze statusem `201`.
 - 2.4. Zmapuj servlet na adres `/adduser`
 - 2.5. Zmodyfikuj kod metody `UsersServlet.doGet()`, aby korzystała z tej samej listy, co `AddUserServlet`.
3. Stwórz filtr uwierzytelniający:
 - 3.1. Stwórz bazę danych `servers`, a w niej tabele `admins` (`admin_id`, `login`, `password`, `session_id`) oraz `session` (`session_id`, `token`);
 - 3.2. Stwórz formularz logowania w pliku `login.htm`, który wykona przekierowanie na servlet `UserLogin`;
 - 3.3. W metodzie `doPost()` pobierz z requesta parametry `login` i `password` i sprawdź, czy taki użytkownik występuje w tabeli `admins`;
 - 3.4. Jeśli tak, to wygeneruj za pomocą klasy `Random` dwie wartości typu `long` – `session_id` i `token`. Przypisz je do nagłówków `Token` i `Session_id`. Tymi samymi wartościami zaktualizuj bazę danych.
 - 3.5. Jeśli nie, to zwróć błąd `401`;
 - 3.6. W metodzie `doFilter()` w klasie `AuthenticationFilter` porównaj wartość nagłówka `Token` i `Session_id` i w przypadku niepoprawnej walidacji zwróć kod błędu `403`;
 - 3.7. Dołącz filtr do servletu `UsersServlet`
4. Ciasteczka:
 - 4.1. W przykładzie z zadania 3, w metodzie `UserLogin.doPost()` zapisz w postaci ciasteczek wartość `login` oraz wygenerowane `session_id`. Ustaw czas życia ciasteczek na 2h.
 - 4.2. Usuń tabelę `sessions` i klucz obcy;
 - 4.3. W metodzie `AuthenticationFilter.doFilter()` dokonaj walidacji użytkownika na podstawie `session_id` i loginu z Cookies;
 - 4.4. Utwórz servlet `UserLogout` i w metodzie `doGet()` usuń ciasteczka `login` i `session_id`;
 - 4.5. Zmapuj servlet na adres `/logout`;
 - 4.6. Utwórz prosty formularz, z przyciskiem, po którego kliknięciu, zostanie wywołana akcja `UserLogout.doGet()`
5. Przekierowania:
 - 5.1. W metodzie `UsersServlet.doGet()`, w przypadku niepoprawnego uwierzytelnienia przekieruj zapytanie na stronę `403.htm` zamiast zwracać kod błędu `403`;
 - 5.2. Taki sam mechanizm zastosuj w metodzie `AuthenticationFilter.doFilter()` (`401.htm`);
6. Sesja
 - 6.1. Zmodyfikuj metodę `LoginServlet.doPost()` tak, aby zamiast ciasteczek, po poprawnym uwierzytelnieniu, otworzyła nową sesję z atrybutem `login`;

- 6.2. W singletonie z zadania 2. Zmień globalną listę użytkowników na obiekt typu `Map<String, Map<User>>`, który dla każdego z adminów będzie przechowywał osobną listę dodanych użytkowników;
- 6.3. Zakomentuj metodę `init()` w klasie `UsersServlet` i zmień kod metody `doPost()`, tak aby wyszukiwała w mapie odpowiedniej listy, po kluczu `login` z sesji i do tej listy dodawała nowego użytkownika;
- 6.4. W metodzie `UserLogout.doGet()` zmień kod tak, aby zamiast usuwania ciasteczek unieważnił sesję;
- 6.5. Dodaj kilku użytkowników po zalogowaniu. Wyloguj się i zaloguj jako inny użytkownik – sprawdź zawartość listy. Następnie zaloguj się do poprzedniego konta i zweryfikuj, czy wcześniej dodani użytkownicy pojawili się na liście.