МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Практична робота № 3**

з дисципліни “**Web-технології та розробка web-систем**”

Виконав:

студент III курсу

групи КП-21

Бураков Д. С.

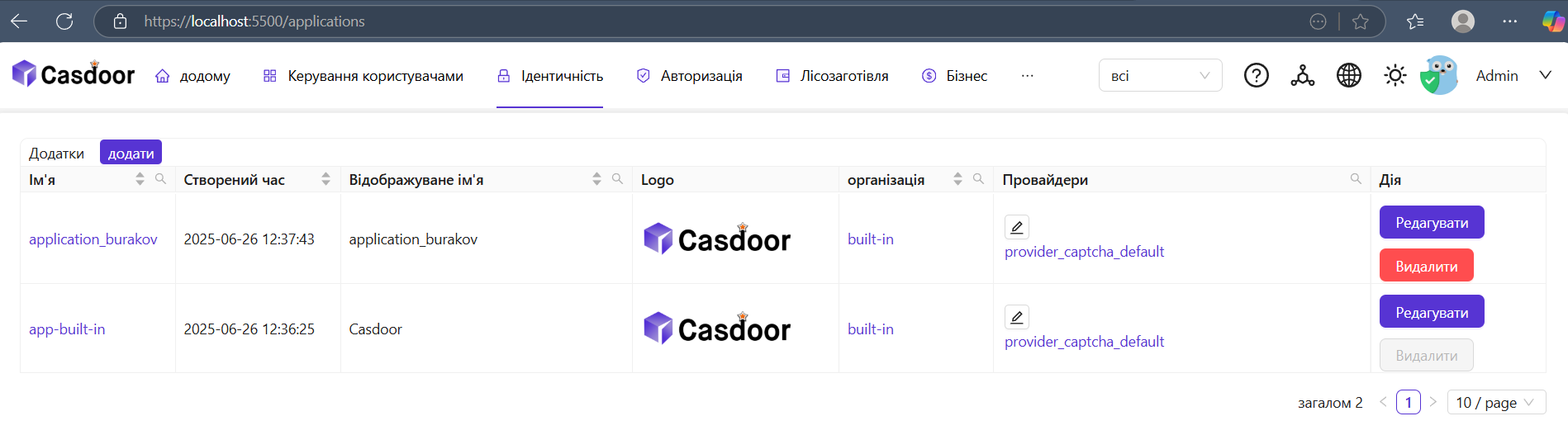
Прийняв:

доцент Дичка А.І.

Київ 2025

У роботі застосовуватимемо готовий Docker-контейнер Casdoor. Оскільки Casdoor за замовчуванням обробляє лише HTTP-запити через порт 8000, для забезпечення коректної взаємодії з сервісом було інтегровано nginx як проксі-сервер, який працює на порті 5500.

Налаштування Casdoor:



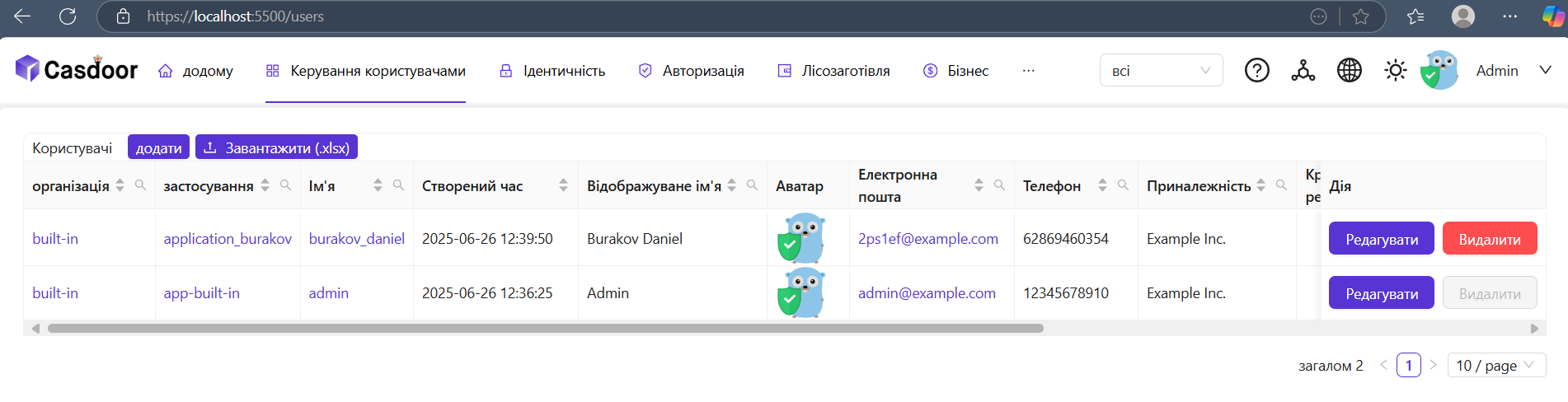


Рис. 1-2. Створення application та користувача на Casdoor

Змінні які використовуються, збережені в application.properties:

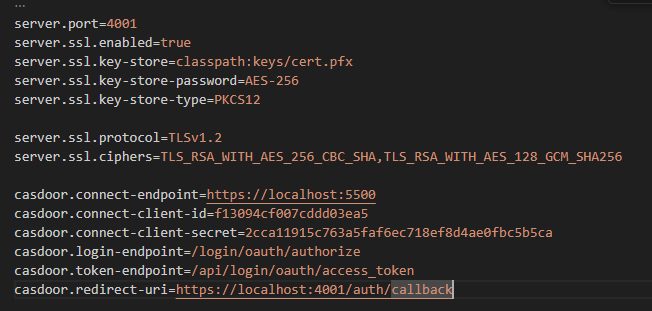


Рис. 3. Визначення змінних в application.properties

Налаштування ендпоінтів для авторизації:

@GetMapping("/login")

public void login(HttpServletResponse response) throws IOException {

System.out.println("Login requested");

String state = UUID.randomUUID().toString();

String redirectUri = casdoorProperties.getRedirectUri();

if (!redirectUri.startsWith("https")) {

redirectUri = redirectUri.replace("http", "https");

}

String authorizeUrl = String.format("%s%s?client\_id=%s&response\_type=code&redirect\_uri=%s&scope=openid profile email&state=%s&nonce=%s",

casdoorProperties.getConnectEndpoint(),

casdoorProperties.getLoginEndpoint(),

casdoorProperties.getConnectClientId(),

redirectUri,

state,

UUID.randomUUID()

);

System.out.println("Authorize URL: " + authorizeUrl);

response.sendRedirect(authorizeUrl);

}

У цьому фрагменті коду налаштовується callback-функція, яка виконується після успішної автентифікації користувача через Casdoor. Ми встановлюємо параметри HttpOnly і Secure для accessTokenCookie, щоб гарантувати його захищене зберігання.:

accessTokenCookie.setHttpOnly(true);

accessTokenCookie.setSecure(true);

Це показник безпеки, але унеможливлює пряме зчитування — саме тому був реалізований окремий ендпоінт /status.

@SuppressWarnings("rawtypes")

@GetMapping("/callback")

public void callback(@RequestParam(required = false) String code,

@RequestParam(required = false) String state,

@RequestParam(required = false) String error,

@RequestParam(required = false) String error\_description,

HttpServletRequest servletRequest, // Renamed from request to servletRequest

HttpServletResponse response) throws IOException {

System.out.println("Callback received. Full URL: " + servletRequest.getRequestURL() + "?" + servletRequest.getQueryString());

if (error != null) {

System.err.println("OAuth error: " + error + " - " + error\_description);

response.sendRedirect("/?error=oauth\_error&message=" + error);

return;

}

if (code == null) {

System.err.println("Missing required 'code' parameter");

response.sendRedirect("/?error=missing\_code");

return;

}

System.out.println("Got code: " + code);

System.out.println("Got state: " + state);

RestTemplate restTemplate = new RestTemplate();

HttpHeaders headers = new HttpHeaders();

headers.setContentType(MediaType.APPLICATION\_FORM\_URLENCODED);

MultiValueMap<String, String> params = new LinkedMultiValueMap<>();

params.add("grant\_type", "authorization\_code");

params.add("code", code);

params.add("redirect\_uri", casdoorProperties.getRedirectUri());

params.add("client\_id", casdoorProperties.getConnectClientId());

params.add("client\_secret", casdoorProperties.getConnectClientSecret());

HttpEntity<MultiValueMap<String, String>> requestEntity = new HttpEntity<>(params, headers); // Renamed from request to requestEntity

ResponseEntity<Map> tokenResponse = restTemplate.postForEntity(

casdoorProperties.getConnectEndpoint() + casdoorProperties.getTokenEndpoint(),

requestEntity,

Map.class

);

if (tokenResponse.getStatusCode().is2xxSuccessful()) {

String accessToken = (String) tokenResponse.getBody().get("access\_token");

if (accessToken != null) {

Cookie accessTokenCookie = new Cookie("access\_token", accessToken);

accessTokenCookie.setPath("/");

accessTokenCookie.setHttpOnly(true);

accessTokenCookie.setSecure(true);

response.addCookie(accessTokenCookie);

System.out.println("Access token stored in cookie");

} else {

System.err.println("Access token not found in response");

}

} else {

System.err.println("Failed to retrieve access token: " + tokenResponse.getStatusCode());

}

response.sendRedirect("/");

}

@GetMapping("/status")

public ResponseEntity<Map<String, Object>> getUserStatus(HttpServletRequest request) {

Map<String, Object> response = new HashMap<>();

Cookie[] cookies = request.getCookies();

boolean loggedIn = false;

if (cookies != null) {

for (Cookie cookie : cookies) {

if ("access\_token".equals(cookie.getName())) {

loggedIn = true;

break;

}

}

}

response.put("loggedIn", loggedIn);

return ResponseEntity.ok(response);

}

Тепер подивимось на перебіг автентифікації з devTools:

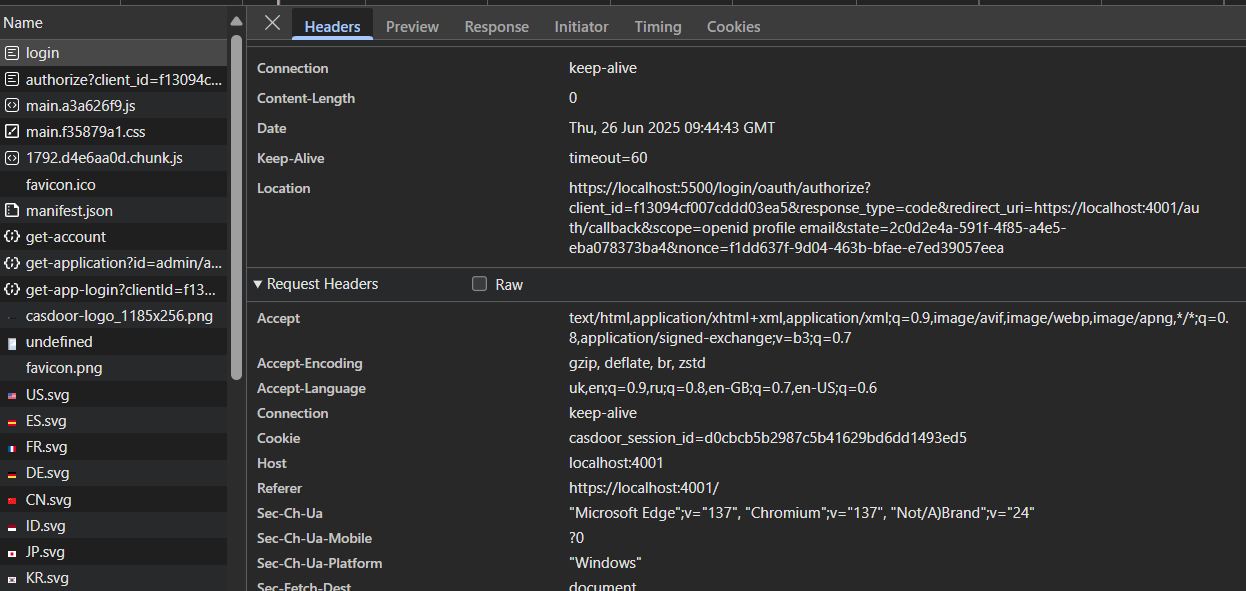


Рис. 4. Запит при натисканні кнопки Login

На рисунку вище, можемо побачити що при спробі залогінитися нас переадресовує на Casdoor, де і відбувається автентифікація. Після цього Casdoor посилає запит назад на callback:

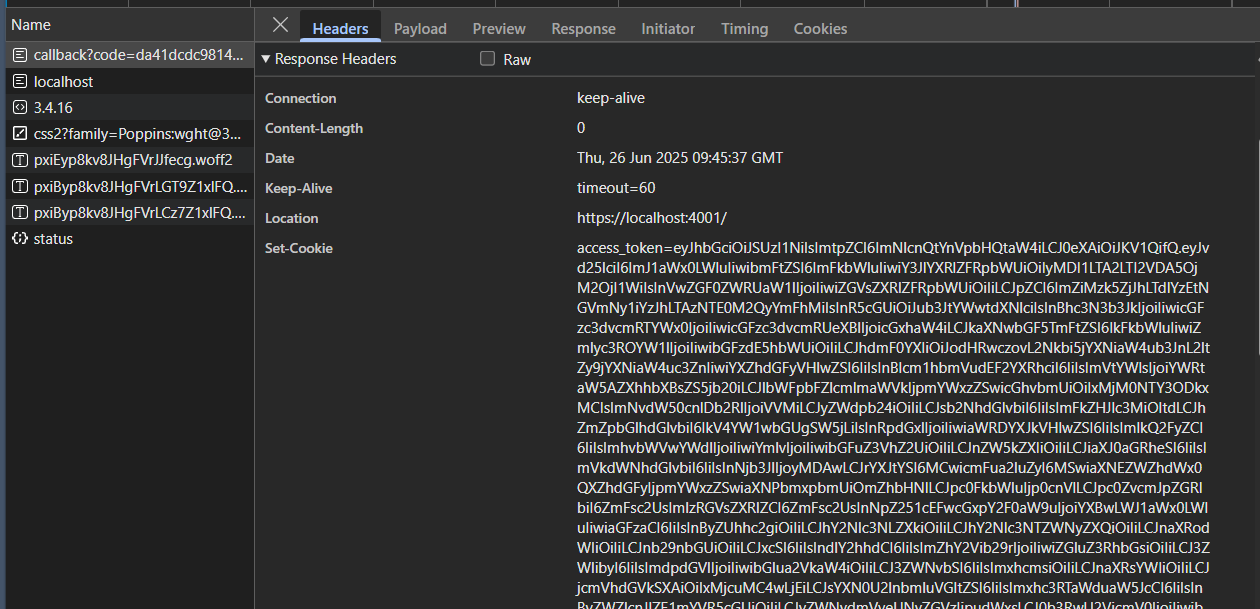


Рис. 5. Виклик ендпоінту Callback і втсановлення Cookies

Надалі робимо запит на сервер вже з Cookies, щоб оновити статус (адже у нас немає доступу):

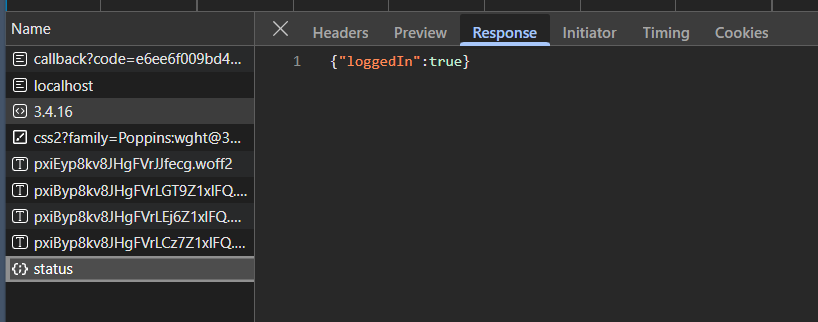


Рис. 6. Запит на перевірку чи встановлені Cookies

Та виконуємо запит щоб отримати інформацію користувача:

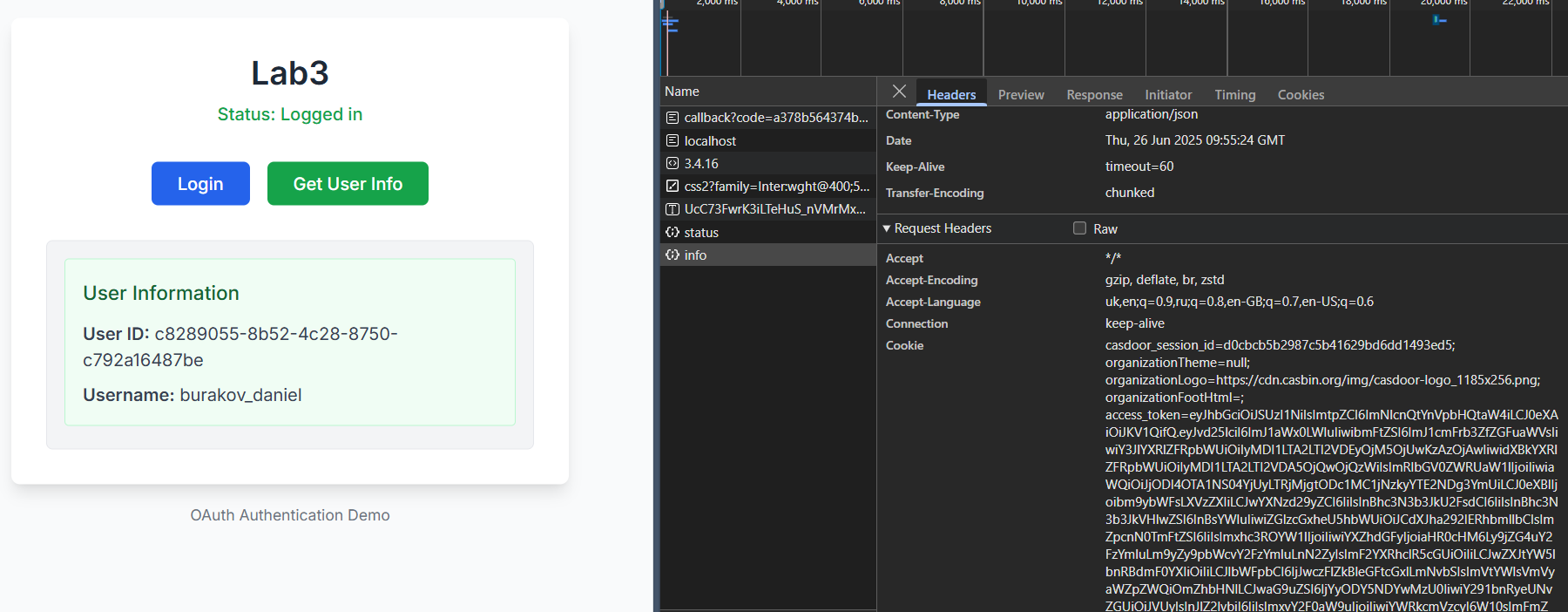


Рис. 7. Запит на отримання даних користувача з Cookies

Користувацький інтерфейс

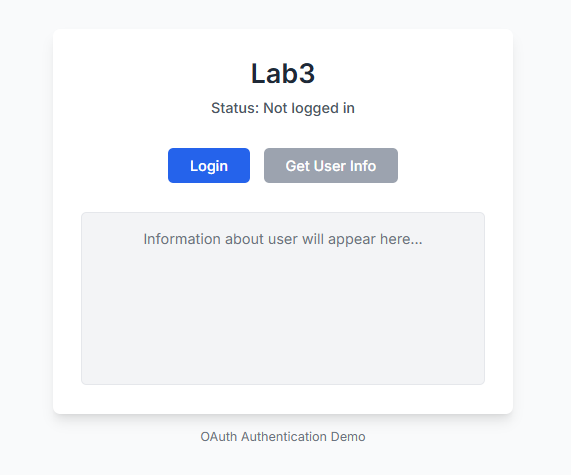


Рис. 8. Головна сторінка, без встановлених Cookies

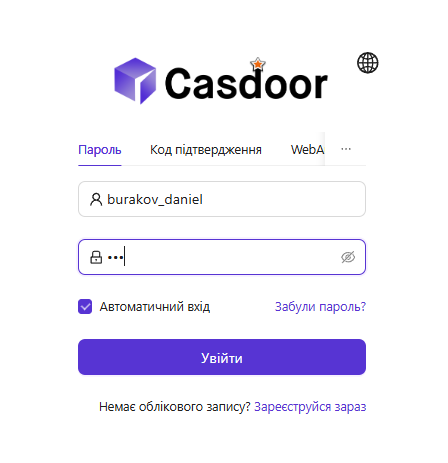


Рис. 9. Процес авторизації в Casdoor

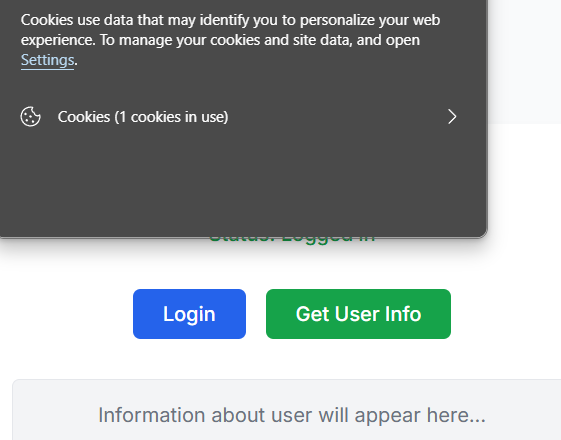


Рис. 10. Головна сторінка, після успішної авторизації і встановлення Cookies

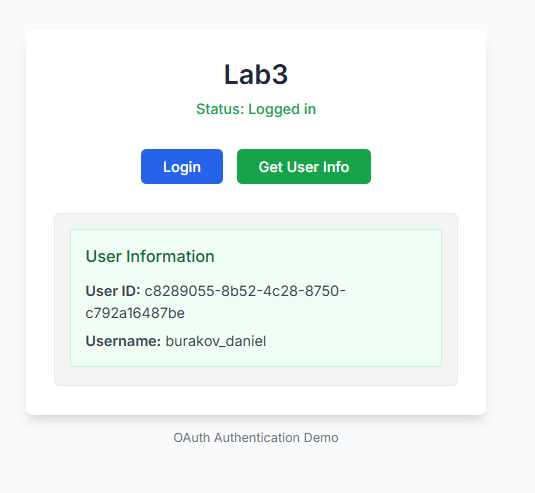


Рис. 11. Успішне отримання інформації залогіненого користувача

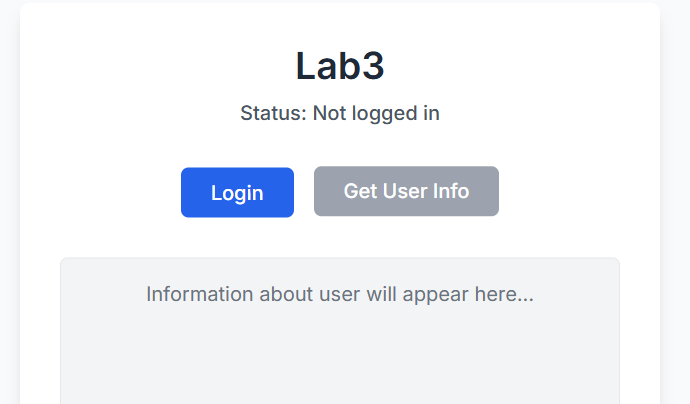


Рис. 12. Спроба отримати незалогіненому користувачу інформацію (кнопка отримати інфо не активна)

**Висновки:**

У ході виконання лабораторної роботи було реалізовано процес автентифікації користувача за допомогою сервісу Casdoor із використанням протоколу OAuth2. Було оброблено callback-запит для отримання токена, забезпечено збереження токена в cookie та реалізовано логіку визначення статусу користувача і отримання його даних на стороні бекенду.  
Окрім цього, створено простий користувацький інтерфейс, який динамічно змінюється залежно від стану авторизації.