

Міністерство освіти і науки України  
Державний заклад  
«Луганський національний університет імені Тараса Шевченка»

Навчально-науковий інститут математики та інформаційних технологій

Кафедра інформаційних технологій та систем

**Тимошин Сергій Сергійович**

**ДОСЛІДЖЕННЯ ТА РОЗРОБКА СИСТЕМИ АУТЕНТИФІКАЦІЇ  
ДО РІЗНИХ ПРОГРАМНИХ СЕРВІСІВ В КОРПОРАЦІЇ**

**кваліфікаційна робота  
здобувача вищої освіти другого (магістерського) рівня  
освітньої програми «Комп'ютерні мережі»  
за спеціальністю 123 Комп'ютерна інженерія**

Особистий підпис \_\_\_\_\_ Сергій ТИМОШИН

Науковий керівник \_\_\_\_\_ Микола СЕМЕНОВ,  
кандидат педагогічних наук, доцент  
кафедри інформаційних технологій  
та систем

Завідувач кафедри \_\_\_\_\_ Микола СЕМЕНОВ,  
кандидат педагогічних наук, доцент  
кафедри інформаційних технологій  
та систем

Полтава – 2025

## АНОТАЦІЯ

**Тема:** Дослідження та розробка системи аутентифікації до різних програмних сервісів в корпорації.

**Спеціальність:** 123 «Комп'ютерна інженерія».

**Установа:** ЛНУ імені Тараса Шевченка, 2024 р.

**Магістерська робота містить:** 82 с., 22 рис., 5 табл., 103 джерел.

**Об'єкт дослідження** – процеси аутентифікації користувачів до програмних сервісів в корпоративних середовищах.

**Предмет дослідження** – методи, протоколи та технології для побудови системи уніфікованої аутентифікації в університеті.

**Мета роботи** – розробити систему аутентифікації, яка забезпечуватиме безпечний, надійний та зручний доступ до університетських програмних сервісів, таких як пошта, Moodle та Teams, з використанням LDAP, багатофакторної аутентифікації (MFA) та сучасних протоколів.

**Результати роботи** – у роботі проаналізовано методи аутентифікації, протоколи OAuth, SAML, OpenID Connect, а також популярні рішення для централізованої аутентифікації (Okta, Keycloak, FreeIPA). Визначено вимоги до системи аутентифікації для університету, розроблено архітектуру системи з інтеграцією LDAP та Keycloak для підтримки MFA. Проведено тестування створеної системи в умовах, наближених до реального використання.

**Ключові слова:** АУТЕНТИФІКАЦІЯ, LDAP, MFA, ПРОТОКОЛИ, УНІФІКОВАНА СИСТЕМА, ОТРИМАННЯ ДОСТУПУ, УНІВЕРСИТЕТСЬКІ СЕРВІСИ.

## ANNOTATION

**Topic:** Research and development of an authentication system for accessing various software services in a corporation.

**Speciality:** 123 "Computer Engineering".

**Institution:** Luhansk Taras Shevchenko National University (LTSNU), 2024 year.

**Master's thesis consists of:** 82 p., 22 im., 5 tables, 110 sources.

**Object of research** – user authentication processes for accessing software services in corporate environments.

**Subject of research** – methods, protocols, and technologies for designing a unified authentication system in a university.

**Objective of the study** – to develop an authentication system that provides secure, reliable, and convenient access to university software services, such as email, Moodle, and Teams, utilizing LDAP, multi-factor authentication (MFA), and modern protocols.

**Results of the study** – the thesis analyzed authentication methods, protocols such as OAuth, SAML, and OpenID Connect, and leading solutions for centralized authentication (Okta, Keycloak, FreeIPA). Requirements for the university's authentication system were identified, and a system architecture integrating LDAP and Keycloak for MFA support was developed. The system was tested under conditions close to real-world usage scenarios.

**Keywords:** AUTHENTICATION, LDAP, MFA, PROTOCOLS, UNIFIED SYSTEM, ACCESS MANAGEMENT, UNIVERSITY SERVICES.

## ЗМІСТ

ВСТУП .....	6
РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО СИСТЕМ АУТЕНТИФІКАЦІЇ .....	9
1.1. Огляд основних методів аутентифікації: однофакторні, багатофакторні та біометричні підходи.....	9
1.2. Аналіз протоколів аутентифікації: OAuth, SAML, OpenID Connect	14
1.3. Розгляд існуючих рішень для уніфікованої аутентифікації в корпоративних середовищах .....	17
1.4. Визначення основних вимог до системи аутентифікації для університетських систем .....	21
Висновки до розділу .....	23
РОЗДІЛ 2. РОЗРОБКА КОНЦЕПЦІЇ СИСТЕМИ АУТЕНТИФІКАЦІЇ ДЛЯ УНІВЕРСИТЕТУ .....	25
2.1. Опис обраної архітектури системи аутентифікації. ....	25
2.2. Налаштування та інтеграція системи з університетською поштою, Moodle та Teams .....	28
2.3. Розробка технічних вимог до системи та специфікація її компонентів	29
Висновки до розділу .....	32
РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ АУТЕНТИФІКАЦІЇ .....	33
3.1. Розгортання системи для централізованої аутентифікації. ....	33
3.2. Реалізація доступу через LDAP та OTP.....	38
3.3. Забезпечення багатофакторної аутентифікації: конфігурація токенів, SMS та біометричних даних .....	43
3.4.Тестування системи в умовах, наближених до реального використання .....	46
Висновки до розділу .....	49
РОЗДІЛ 4. ВПРОВАДЖЕННЯ ТА ПЕРСПЕКТИВИ ДОСЛІДЖЕННЯ .....	50

4.1. Керівництво користувача щодо використання системи аутентифікації	50
4.2. Інструкція для адміністраторів системи аутентифікації.....	52
4.3. Перспективи розширення функціональності системи, масштабування системи в інших установах .....	55
Висновки до розділу .....	56
ВИСНОВКИ.....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60
ДОДАТОК А. Playbook для розгортання системи аутентифікації .....	70

## ВСТУП

Дослідження та розробка системи аутентифікації до різних програмних сервісів в корпорації є актуальним завданням у сучасних умовах стрімкого розвитку інформаційних технологій та посилення вимог до кібербезпеки. В умовах цифровізації бізнес-процесів та інтеграції різних програмних сервісів в корпоративних середовищах виникає необхідність забезпечення ефективного та безпечного доступу до них. Використання традиційних методів аутентифікації, таких як паролі, стає все менш ефективним через ризики компрометації, складність управління великою кількістю облікових записів та підвищені вимоги до зручності користування.

Проблема ускладнюється різноманітністю програмного забезпечення, яке використовується в корпораціях, а також необхідністю інтеграції існуючих систем з новими рішеннями. Відсутність єдиної уніфікованої системи аутентифікації може призводити до збільшення операційних витрат, зниження продуктивності працівників та підвищення ризику несанкціонованого доступу до конфіденційних даних. У зв'язку з цим, виникає потреба у створенні системи аутентифікації, яка забезпечувала б надійний захист даних, масштабованість, легкість інтеграції та високу зручність для користувачів. Розглянуті проблемні питання підтверджують актуальність теми та були обрані як критерії для формулювання теми магістерського дослідження.

Відповідно до обраної теми сформульовано об'єкт та предмет дослідження, мету та завдання.

**Об'єкт дослідження:** процеси аутентифікації користувачів у корпоративних інформаційних системах.

**Предмет дослідження:** методи та архітектурні рішення для побудови системи аутентифікації з використанням LDAP для забезпечення єдиного доступу до університетських інформаційних систем, таких як Moodle, пошта та Microsoft Teams.

**Метою роботи** є дослідження сучасних методів аутентифікації та розробка ефективної системи аутентифікації для доступу до програмних сервісів у корпоративному середовищі.

Для досягнення поставленої мети необхідно вирішити низку **завдань**:

- провести аналіз існуючих підходів до аутентифікації, визначити їх переваги та недоліки; дослідити специфічні вимоги до аутентифікації у корпоративному середовищі;
- розробити концепцію уніфікованої системи аутентифікації з використанням сучасних технологій, таких як біометрія, багатофакторна аутентифікація та єдина точка входу (SSO);
- створити прототип системи та провести її тестування в умовах, наближених до реальних корпоративних середовищ;
- оцінити ефективність запропонованого рішення з точки зору безпеки, продуктивності та зручності користування.

Звуження предмету дослідження: в якості корпорації будемо розглядати університет та його середовище, що дозволить мати доступ до реального середовища та провести певні дії по впровадженню результатів дослідження.

Результатом роботи стане повністю функціональна система аутентифікації, що забезпечує доступ до університетської пошти, Moodle та Teams. Реалізоване рішення сприятиме підвищенню рівня безпеки та зручності використання інформаційних сервісів університету. Впровадження запропонованої системи можуть мати значний практичний ефект, сприяючи підвищенню рівня безпеки корпоративних даних, оптимізації бізнес-процесів та підвищенню загальної ефективності роботи організації.

У першому розділі виконано огляд методів аутентифікації, зокрема однофакторних, багатофакторних та біометричних, а також проаналізовано протоколи аутентифікації, такі як OAuth, SAML та OpenID Connect. Розглянуто їх переваги, недоліки та можливості використання для єдиного доступу до різних систем.

У другому розділі описано архітектуру обраної системи аутентифікації, яка базується на протоколі LDAP. Наведено детальний опис її компонентів, взаємодії між ними та особливостей інтеграції з університетськими платформами, такими як Moodle, корпоративна пошта та Microsoft Teams.

Третій розділ присвячено реалізації системи аутентифікації, включаючи налаштування LDAP-сервера, інтеграцію з клієнтськими системами та впровадження багатофакторної аутентифікації для підвищення безпеки.

У четвертому розділі проведено тестування розробленої системи, оцінено її ефективність та безпеку, а також сформульовано рекомендації для подальшого вдосконалення. Робота може бути корисною для впровадження централізованих систем аутентифікації в освітніх і корпоративних середовищах.



## **РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО СИСТЕМ АУТЕНТИФІКАЦІЇ**

### **1.1. Огляд основних методів аутентифікації: однофакторні, багатофакторні та біометричні підходи**

Сьогодні університетські інформаційні системи відіграють ключову роль у забезпеченні освітнього процесу, управлінні навчальними ресурсами та організації комунікації між учасниками освітнього середовища. Водночас, інтеграція різних програмних рішень, таких як системи управління навчальним контентом (LMS), платформи для дистанційного навчання та сервіси для внутрішнього адміністрування, створює значні виклики у сфері інформаційної безпеки. Серед них особливе місце займає аутентифікація користувачів, адже саме від її надійності залежить захист конфіденційних даних, підтримка академічної доброчесності та забезпечення безперервного доступу до ресурсів. Для визначення найбільш ефективного підходу до побудови системи аутентифікації необхідно детально розглянути існуючі методи, їх переваги, недоліки та відповідність потребам університету.

Однофакторна аутентифікація (1FA) є найпростішим і найпоширенішим методом ідентифікації користувача. Вона базується на використанні лише одного фактора перевірки. Найчастіше таким фактором є пароль або PIN-код, які користувач повинен ввести для доступу до системи. Інколи це може бути фізичний носій, наприклад, магнітна картка або ключ доступу. Основною перевагою однофакторної аутентифікації є її простота у використанні, але вона має низький рівень безпеки. Якщо пароль або інший фактор буде скомпрометований, зломисник отримає повний доступ до системи.

Багатофакторна аутентифікація (MFA) забезпечує вищий рівень безпеки за рахунок використання кількох факторів ідентифікації, які повинні належати до різних категорій [29]:

- паролі, PIN-коди, відповіді на секретні запитання;
- фізичні токени, смартфони для отримання SMS або генерації одноразових кодів (OTP);

- біометричні дані, такі як відбитки пальців, розпізнавання обличчя або сканування райдужної оболонки ока [10; 17; 18; 30; 40; 45; 51; 57; 58; 65; 79; 92; 97; 99; 101; 104; 106].

MFA значно ускладнює доступ до системи для зловмисників, оскільки для успішної аутентифікації їм потрібно зламати або викрасти всі використовувані фактори. Наприклад, навіть якщо пароль буде скомпрометований, користувачеві все одно потрібно підтвердити доступ через смартфон або біометрію.

Нарешті, біометрична аутентифікація використовує унікальні фізичні або поведінкові характеристики людини для ідентифікації. Найпоширеніші методи включають:

- відбитки пальців: сканування ліній та вигинів на пальцях;
- розпізнавання обличчя: аналіз унікальних рис обличчя за допомогою камер;
- сканування райдужної оболонки або сітківки ока: вивчення унікальних візерунків в очі;
- голосова аутентифікація: аналіз голосових характеристик користувача.

Біометричні методи є дуже зручними, оскільки користувачам не потрібно запам'ятовувати паролі або носити додаткові пристрої. Водночас вони мають високий рівень безпеки, адже біометричні дані важко підробити. Однак основною проблемою є їхня незворотність: у разі компрометації біометричних даних користувач не може змінити свій відбиток пальця чи райдужну оболонку.

Розглянемо як технічно можна реалізувати біометричну аутентифікацію. Використовуємо Arduino та модуль, який працює з біометричними даними, наприклад, **сканер відбитків пальців**. Це досить поширений варіант для проектів на Arduino, оскільки він простий у налаштуванні та інтеграції. Покрокова інструкція:

Для створення цієї системи необхідно таке обладнання:

1. Arduino (UNO, Mega або інший сумісний мікроконтролер)

2. Біометричний модуль (наприклад, GT-511C3 або R307)
3. Кабелі для підключення
4. Блок живлення (за потреби)
5. Серійний монітор або дисплей (наприклад, LCD 16x2) для виводу результатів
6. Додаткові компоненти (резистори, світлодіоди, кнопки — за необхідності)
7. Загальна вартість обладнання становить приблизно **\$75 - \$135** (за цінами [12]).

На рис. 1.1 та 1.2 зображені модулі для проведення біометрії.



Рис. 1.1. Модуль GT-511C3 [12]



Рис. 1.2. Модуль R307 [12]

Як бачимо з рис.1.1 та рис. 1.2 модулі невеликі за розміром та зручні у користуванні.

Для підключення модулю з'єднаємо його з Arduino:

- **VCC** модуля до **5V** на Arduino.
- **GND** модуля до **GND** на Arduino.
- **TX** модуля до одного з цифрових пінів (наприклад, **D2**).
- **RX** модуля до іншого цифрового піна (наприклад, **D3**).
- Дисплей через **I2C** або напряму.

Для роботи з біометричними модулями встановлюємо бібліотеку, наприклад, Adafruit Fingerprint Sensor Library. Її можна встановити через Arduino IDE: Sketch → Include Library → Manage Libraries → Adafruit Fingerprint Sensor Library.

Код для запису та перевірки відбитків пальців представлено у листингу 1.1:

## Код для біометричної аутентифікації на C++

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>

// Піни для підключення біометричного модуля
SoftwareSerial mySerial(2, 3); // RX, TX
Adafruit_Fingerprint finger(&mySerial);

void setup() {
  Serial.begin(9600); // Серійний монітор
  while (!Serial);
  Serial.println("Біометрична аутентифікація за допомогою відбитків пальців");

  // Початок роботи з модулем
  finger.begin(57600);
  if (finger.verifyPassword()) {
    Serial.println("Модуль успішно підключений");
  } else {
    Serial.println("Помилка підключення до модуля");
    while (1);
  }
}

void loop() {
  Serial.println("Помістіть палець на сканер...");
  int result = finger.getImage();

  if (result == FINGERPRINT_OK) {
    Serial.println("Відбиток зчитано");
    result = finger.image2Tz();
    if (result == FINGERPRINT_OK) {
      Serial.println("Відбиток збережено для перевірки");
      result = finger.fingerSearch();
      if (result == FINGERPRINT_OK) {
        Serial.print("Знайдено користувача з ID: ");
        Serial.println(finger.fingerID);
      } else {
        Serial.println("Відбиток не знайдено в базі");
      }
    }
  } else {
    Serial.println("Помилка зчитування відбитка");
  }
  delay(2000); // Пауза перед наступною спробою
}
```

Як бачимо з коду листингу 1.1 для запису відбитків пальців у пам'ять модуля можна використати функції бібліотеки (наприклад, `finger.storeModel()`). Програма буде запитувати користувача прикласти палець кілька разів для створення шаблону. Модуль зчитує відбиток пальця і порівнює його з шаблонами, що збережені в пам'яті. Якщо відповідність знайдено, повертається унікальний ідентифікатор (ID). У разі успішної аутентифікації, можна запрограмувати виконання додаткових дій – наприклад доступ до ресурсу.

Розглянувши основні методи аутентифікації — однофакторні, багатофакторні та біометричні, можна дійти висновку, що для університетських систем оптимальним вибором є багатофакторна аутентифікація. Вона забезпечує вищий рівень безпеки порівняно з однофакторними методами завдяки використанню декількох факторів перевірки, таких як пароль і одноразовий код або біометричні дані. Хоча біометричні методи самі по собі мають високу надійність, їх повсюдне впровадження може бути ускладнене через вартість обладнання та можливі проблеми із конфіденційністю. Поєднання багатофакторної аутентифікації із доступними засобами, такими як паролі та мобільні додатки, забезпечує універсальний, безпечний і зручний спосіб роботи з університетськими ресурсами, відповідний сучасним потребам академічної спільноти.

## **1.2. Аналіз протоколів аутентифікації: OAuth, SAML, OpenID Connect**

Розглянувши основні методи аутентифікації, перейдемо до аналізу протоколів, таких як OAuth, SAML та OpenID Connect. Це важливо зробити, оскільки саме протоколи визначають способи взаємодії між клієнтськими додатками, серверами аутентифікації та ресурсами. Вивчення цих протоколів дозволить зрозуміти їх сильні та слабкі сторони, визначити, який із них найкраще підходить для інтеграції в університетську інфраструктуру, та врахувати специфіку роботи з різними інформаційними системами. На основі проведеного аналізу ми зможемо обґрунтовано обрати найбільш ефективне

рішення для побудови централізованої системи аутентифікації. Це, у свою чергу, стане основою для переходу до наступного кроку дослідження — проєктування та впровадження обраної архітектури.

Сучасні протоколи аутентифікації та авторизації забезпечують захищений доступ до ресурсів і є основою для інтеграції різних сервісів у межах корпорацій та хмарних середовищ. Розглянемо їх особливості, переваги та недоліки.

**OAuth (Open Authorization)** — це протокол авторизації, який дозволяє одній системі надавати доступ до своїх ресурсів іншій системі без передачі облікових даних користувача. Основною метою OAuth є делегування доступу через **токени доступу** [25; 95].

Основними компонентами системи є: **ресурсний сервер** — API або система, що надає доступ до ресурсів; **клієнт** — додаток, який потребує доступу до ресурсів; **сервер авторизації** — видає токени доступу; **користувач** — надає дозвіл клієнту на доступ до своїх даних. Користувач дозволяє клієнту доступ до своїх ресурсів. Сервер авторизації видає клієнту токен доступу. Токен використовується для запитів до ресурсного сервера.

Перевагою такого доступу є відсутність пароллю, можливість відкликання токenu, можливість різних сценаріїв авторизації (наприклад, для веб-, мобільних або серверних додатків). Але використання не підтримує аутентифікацію користувача напряму, вимагає додаткової реалізації для шифрування токенів. Тому OAuth часто використовується для доступу до API (наприклад, Google API, Facebook API).

**SAML (Security Assertion Markup Language)** — це протокол обміну даними для аутентифікації та авторизації на основі XML. Основна мета — забезпечити єдиний вхід (Single Sign-On, SSO) між кількома системами.

Основні компоненти: **Identity Provider (IdP)** — постачальник ідентичностей (забезпечує аутентифікацію); **Service Provider (SP)** — сервіс, що надає доступ до ресурсів після підтвердження особи; **Користувач** — авторизується через IdP. Користувач запитує доступ до SP. SP перенаправляє

користувача до IdP для аутентифікації. Після успішної аутентифікації IdP генерує SAML-асерцію, яка повертається до SP. SP надає доступ до ресурсів.

**SAML** забезпечує єдиний вхід (SSO), висока безпека завдяки використанню цифрових підписів і шифрування. Але це застаріла технологія та складно реалізується через використання XML і криптографії. Широко використовується в корпоративному середовищі для інтеграції між внутрішніми системами (наприклад, ERP, CRM).

**OpenID Connect** — це надбудова над OAuth 2.0, яка додає функціонал для аутентифікації користувача. Це сучасний протокол, який надає можливість підтвердження особи через *ID-токени* [95].

Основні компоненти: **OpenID-провайдер** — сервіс, що виконує аутентифікацію (наприклад, Google, Microsoft); **Клієнт** — додаток, що потребує ідентифікації користувача; **Користувач** — виконує аутентифікацію через OpenID-провайдера. Клієнт ініціює аутентифікацію. OpenID-провайдер аутентифікує користувача та повертає ID-токен. Клієнт використовує ID-токен для отримання інформації про користувача.

**OpenID Connect** поєднує функції аутентифікації та авторизації, використовує JSON для передачі даних спрощує інтеграцію. Широко використовується для аутентифікації в хмарних сервісах

Для порівняння протоколів оформимо табл. 1.1 за 5 критеріями.

Таблиця 1.1

#### Порівняння протоколів

Критерій	OAuth	SAML	OpenID Connect
Призначення	Авторизація	Аутентифікація та авторизація	Аутентифікація та авторизація
Формат даних	JSON	XML	JSON



Критерій	OAuth	SAML	OpenID Connect
Мобільна підтримка	Висока	Низька	Висока
Безпека	Залежить від токенів	Висока (цифрові підписи)	Висока (OIDC додає захист до OAuth)
Сценарій використання	API інтеграції, мобільні додатки	Корпоративні сервіси, SSO	Хмарні сервіси, веб-додатки

Як показано на табл. 1.1. **OAuth** ідеально підходить для делегування доступу до API, але потребує доповнення для аутентифікації. **SAML** забезпечує стабільний і безпечний доступ у корпоративних середовищах, але через складність і XML знижується популярність. **OpenID Connect** — це сучасне рішення, яке поєднує переваги OAuth та SAML, роблячи його оптимальним для інтеграції у веб- та мобільні додатки.

У виборі протоколу слід враховувати потреби системи, вимоги до безпеки та технічну складність впровадження.

Для реалізації єдиного доступу (Single Sign-On, SSO) до різних систем в університеті, таких як Moodle і корпоративна пошта, можна використовувати протоколи OAuth, SAML та OpenID Connect. Вони забезпечують централізовану аутентифікацію і дозволяють користувачам отримувати доступ до кількох сервісів без повторного введення облікових даних. Далі проаналізуємо, як їх застосовувати в корпоративному середовищі університету.

### 1.3. Розгляд існуючих рішень для уніфікованої аутентифікації в корпоративних середовищах

Розглянемо існуючі рішення для реалізації уніфікованої аутентифікації в корпоративних середовищах. Основну увагу будемо приділяти аналізу готових програмних платформ та сервісів, які забезпечують централізований доступ до різних інформаційних систем. Серед рішень такі, як Microsoft Active Directory Federation Services (AD FS) або Microsoft Entra ID (більш сучасне

рішення), Okta, Ping Identity, Auth0, а також інтеграція LDAP з іншими корпоративними сервісами. Опишемо ключові характеристики цих систем, зокрема підтримувані протоколи аутентифікації, сумісність із популярними корпоративними інструментами (наприклад, системами управління навчанням, поштовими сервісами, платформами для спільної роботи), масштабуємість, гнучкість у налаштуванні політик безпеки та можливість інтеграції багатофакторної аутентифікації. Також спробуємо визначити переваги та недоліки використання комерційних платформ у порівнянні з відкритими рішеннями, такими як Keycloak чи FreeIPA. Відповідно до економічної доцільності і можливостей впровадження таких рішень в університетському середовищі, з урахуванням обмеженості бюджету та вимог до безпеки даних запропонуємо оптимальний варіант.

Служба об'єднання Active Directory (AD FS) забезпечує об'єднане керування ідентифікацією та доступом, безпечно надаючи доступ до цифрових посвідчень і прав на них поза межами безпеки та корпоративних кордонів. AD FS розширює можливість використання функцій єдиного входу, доступних у межах однієї безпеки або корпоративного кордону, на програми, орієнтовані на Інтернет, щоб забезпечити клієнтам, партнерам і постачальникам оптимізовану взаємодію з користувачем під час доступу до веб-додатків організації.

Microsoft Entra ID – це хмарна служба управління ідентифікацією та доступом, яку співробітники можуть використовувати для доступу до зовнішніх ресурсів. Прикладами ресурсів є Microsoft 365, портал Azure та тисячі інших SaaS-додатків. Microsoft Entra ID також допомагає їм отримувати доступ до внутрішніх ресурсів, таких як програми в корпоративній інтрамережі, а також будь-які хмарні програми, розроблені для організації. [8; 38; 39; 66].

Okta — це хмарна платформа для управління ідентифікацією та доступом, яка пропонує інструменти для уніфікованої аутентифікації, підтримки багатофакторної аутентифікації та інтеграції з різними

програмними сервісами. Okta підтримує протоколи OAuth, OpenID Connect та SAML, що робить її універсальним рішенням для корпоративних середовищ. Вона також має зручний інтерфейс для адміністраторів і користувачів, але залежність від хмарної інфраструктури може бути обмеженням для деяких організацій. [80]

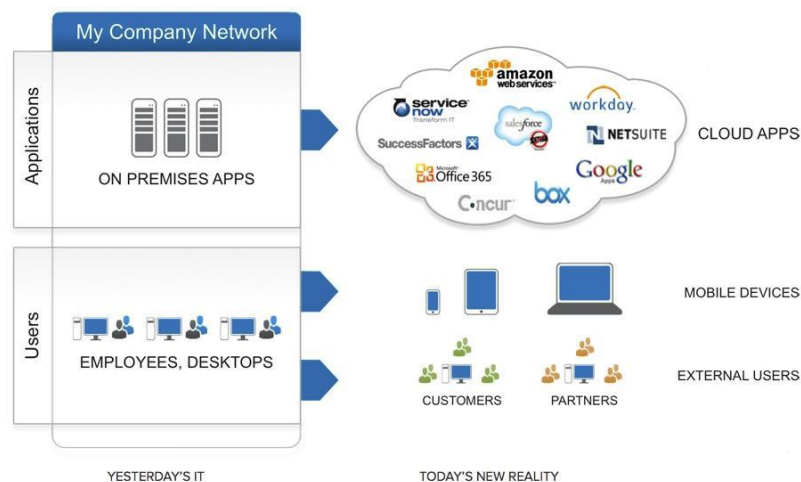


Рис. 1.3. Нова реальність доступу за версією компанії Okta

У більшості підприємств Active Directory (AD) від Microsoft є авторитетним каталогом користувачів, який керує доступом до ключових бізнес-програм. SaaS-додатки були розроблені з власними директоріями користувачів, і оскільки вони працюють за межами брандмауера, зазвичай знаходяться поза досяжністю AD. Це обмеження вимагає, щоб користувачі запам'ятовували кілька імен користувачів та логінів, а ІТ змушують створювати, керувати та відображати облікові записи користувачів у AD та в їхніх програмах SaaS. Okta бореться з цими проблемами за допомогою повної, надійної та простої у використанні інтеграції AD SSO.

До альтернатив Active Directory належать LDAP та інші локальні менеджери посвідчень. В цілому, «рішення Okta» долає економічні та технологічні обмеження будь-якої застарілої схеми управління ідентифікацією, яку ви використовуєте в даний момент.



Рис. 1.4. Інтеграція в Okta

Ping Identity — це потужна платформа для управління доступом, яка дозволяє забезпечити єдиний вхід (SSO) до корпоративних ресурсів і підтримує складні сценарії аутентифікації. Вона інтегрується з локальними системами та хмарними сервісами, забезпечуючи високу гнучкість у використанні. Ping Identity підтримує протоколи OAuth, SAML і OpenID Connect, а також має розширені функції безпеки, але її впровадження може потребувати значних технічних ресурсів. [41]

Auth0 — це хмарна платформа, яка спеціалізується на легкій інтеграції аутентифікації в додатки. Вона підтримує всі популярні протоколи, має зручний API для розробників і пропонує функції кастомізації. Auth0 добре підходить для організацій, які потребують швидкого та гнучкого рішення для аутентифікації, але її комерційна модель може бути обмеженням для невеликих установ. [13;]

Аналізуючи прайси приведених інструментів приходимо до висновку про неможливість їх використання для українського університету в умовах обмеженого бюджету.

Тому для нас представляє інтерес розглянути рішення з відкритим кодом Keycloak та FreeIPA.

Keycloak — це відкрите програмне рішення для управління ідентифікацією та доступом, розроблене для інтеграції з веб-додатками та службами. Keycloak підтримує протоколи OAuth 2.0, OpenID Connect і SAML,

що дозволяє реалізувати функції єдиного входу (SSO) і багатофакторної аутентифікації. Система забезпечує кастомізацію інтерфейсу входу, а також легку інтеграцію з LDAP-серверами та зовнішніми провайдерами ідентифікації. Keycloak є гнучким і масштабованим рішенням, що робить його популярним вибором для університетських середовищ, однак його впровадження та підтримка можуть вимагати значних технічних знань. [50]

FreeIPA — це комплексне рішення з відкритим кодом для управління ідентифікацією, аутентифікацією та авторизацією в локальних та мережових середовищах. FreeIPA об'єднує LDAP, Kerberos, DNS і сервіси сертифікації в єдину систему, що дозволяє забезпечити централізоване управління користувачами та доступом. FreeIPA ідеально підходить для організацій з локальною інфраструктурою, пропонуючи високу безпеку та інтеграцію з Unix- і Linux-середовищами. Однак складність налаштування та обмежена підтримка протоколів сучасної веб-аутентифікації (як-от OAuth чи OpenID Connect) може бути недоліком у порівнянні з іншими рішеннями. [33]

#### **1.4. Визначення основних вимог до системи аутентифікації для університетських систем**

Основні вимоги до системи аутентифікації для університетських систем визначаються потребами в забезпеченні безпеки, зручності користування та сумісності з різними інформаційними сервісами, які використовуються в університеті. Наведемо перелік ключових вимог:

1. Безпека
2. Зручність використання
3. Сумісність
4. Масштабованість
5. Адміністрування
6. Вимоги до конфіденційності

Рис. 1.5. Ключові вимоги системи аутентифікації

Розглянемо основні вимоги, які представлені на рис. 1.5. Безпека передбачає захист облікових даних користувачів. Дані для входу (логіни та

паролі) мають зберігатися в зашифрованому вигляді. Використання протоколів із шифруванням (TLS/SSL) для передачі даних є обов'язковим. До вимог безпеки відноситься підтримка багатофакторної аутентифікації (MFA). Додавання другого рівня аутентифікації, наприклад, одноразових паролів (OTP), біометричних даних чи апаратних токенів для підвищення безпеки. Важливим аспектом є захист від атак типу "груба сила" та "перехоплення сесій", що включає механізми блокування після кількох невдалих спроб входу та автоматичне завершення сесій після певного періоду відсутності активності.

Зручність використання означає єдиний доступ до всіх систем (Single Sign-On, SSO). Користувачі повинні мати можливість увійти до всіх університетських систем (Moodle, корпоративної пошти, Microsoft Teams тощо) через одну точку аутентифікації. Система має працювати коректно як на стаціонарних комп'ютерах, так і на мобільних пристроях. Процес створення облікових записів повинен бути автоматизованим. Система повинна мати інтеграцію з існуючими базами даних (наприклад, LDAP або Active Directory) для автоматичного створення та видалення облікових записів студентів і співробітників.

Система повинна підтримувати стандартні протоколи аутентифікації(сумісність) , такі як OAuth, SAML, та OpenID Connect, для забезпечення інтеграції з наявними університетськими сервісами.

Вона повинна працювати з локально встановленими платформами (наприклад, локальна версія Moodle) і хмарними сервісами (Office 365, Google Workspace тощо). Гнучкість налаштування прав доступу реалізується в можливості налаштовувати доступ до певних систем або функцій залежно від ролі користувача (студент, викладач, адміністратор).

Масштабованість системи передбачає підтримку великої кількості користувачів. Система має бути здатною обробляти одночасні запити від тисяч користувачів без втрати продуктивності. Система повинна дозволяти додавання нових сервісів або інтеграцію з майбутніми платформами без

значних змін архітектури. Адміністратори повинні мати можливість швидко додавати, видаляти чи змінювати облікові записи користувачів через єдиний інтерфейс. Система має вести детальні журнали доступу та підтримувати інструменти моніторингу для виявлення потенційних загроз. Система має відповідати нормам захисту даних (наприклад, GDPR, якщо університет працює з європейськими партнерами). Збирати лише ті дані, які необхідні для роботи системи, і забезпечувати їх захищене зберігання.

Однак для нас також важливими є критерії, які не відносяться до технічної специфікації: вартість програмного забезпечення та технічного супроводу обраної системи аутентифікації.

Таким чином, реалізація системи аутентифікації для університету повинна забезпечувати високу безпеку даних, легкість використання для студентів і персоналу, доступність для бюджету університету а також інтеграцію з усіма ключовими платформами, такими як Moodle, пошта, та Microsoft Teams. Вибір технологій і архітектури має базуватися на дотриманні цих вимог.

### **Висновки до розділу**

1. Проаналізовано основні методи аутентифікації: однофакторні, багатофакторні та біометричні підходи.
2. Показано, що біометричні методи на сьогодні достатньо легко реалізуються, наведено методику простої реалізації, єдиною перепорою при масштабному використанні є вартість.
3. Стверджується, що для університетських систем оптимальним вибором є багатофакторна аутентифікація.
4. Зроблено порівняльний аналіз протоколів аутентифікації OAuth, SAML та OpenID Connect. Показано, що різні протоколи мають різну ефективність за різними параметрами. OAuth краще використовувати для мобільних додатків, SAML для створення єдиної точки входу, OpenID Connect – для хмарних сервісів.

5. Показано, що основною проблемою системи аутентифікації в університеті є необхідність використання різних технічних рішень, на основі критерію доступності систем з відкритим кодом обрано Keycloak та LDAP.
6. Сформульовано основні критерії ефективності системи аутентифікації : безпека, зручність використання, сумісність, масштабованість адміністрування, вимоги до конфіденційності.



## РОЗДІЛ 2. РОЗРОБКА КОНЦЕПЦІЇ СИСТЕМИ АУТЕНТИФІКАЦІЇ ДЛЯ УНІВЕРСИТЕТУ

### 2.1. Опис обраної архітектури системи аутентифікації.

Для розробки системи аутентифікації університету обрана архітектура, яка представлена загальною схемою на рис. 2.1.

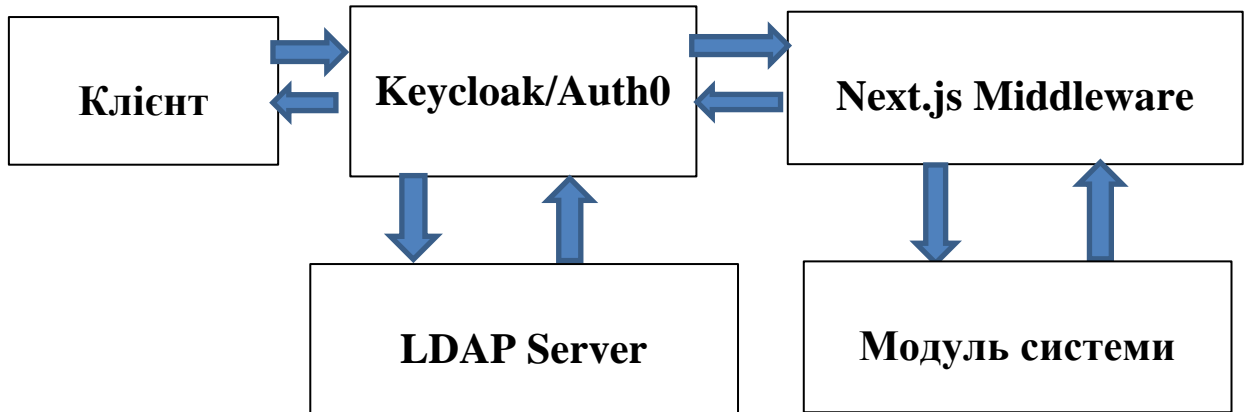


Рис. 2.1. Архітектура системи аутентифікації

На рис. 2.1:

1. **Клієнт** – пристрій користувача
2. **LDAP Server** → централізована база користувачів.
3. **Keycloak/Auth0** → аутентифікація та видача маркерів доступу.
4. **Next.js Middleware** → перевірка маркерів доступу.
5. **Statamic** → обробка контенту для авторизованих користувачів.
6. **Модуль** – плагін, написаний на PHP к якому надається доступ, або зовнішня система (Teams, Moodle, пошта ...).

Для розробки сайту університету, де буде реалізована система аутентифікації обрано архітектуру: Headless CMS (**Statamic**) + Javascript Framework (**Next.js**). Обрана архітектура представлена на рис. 2.2.

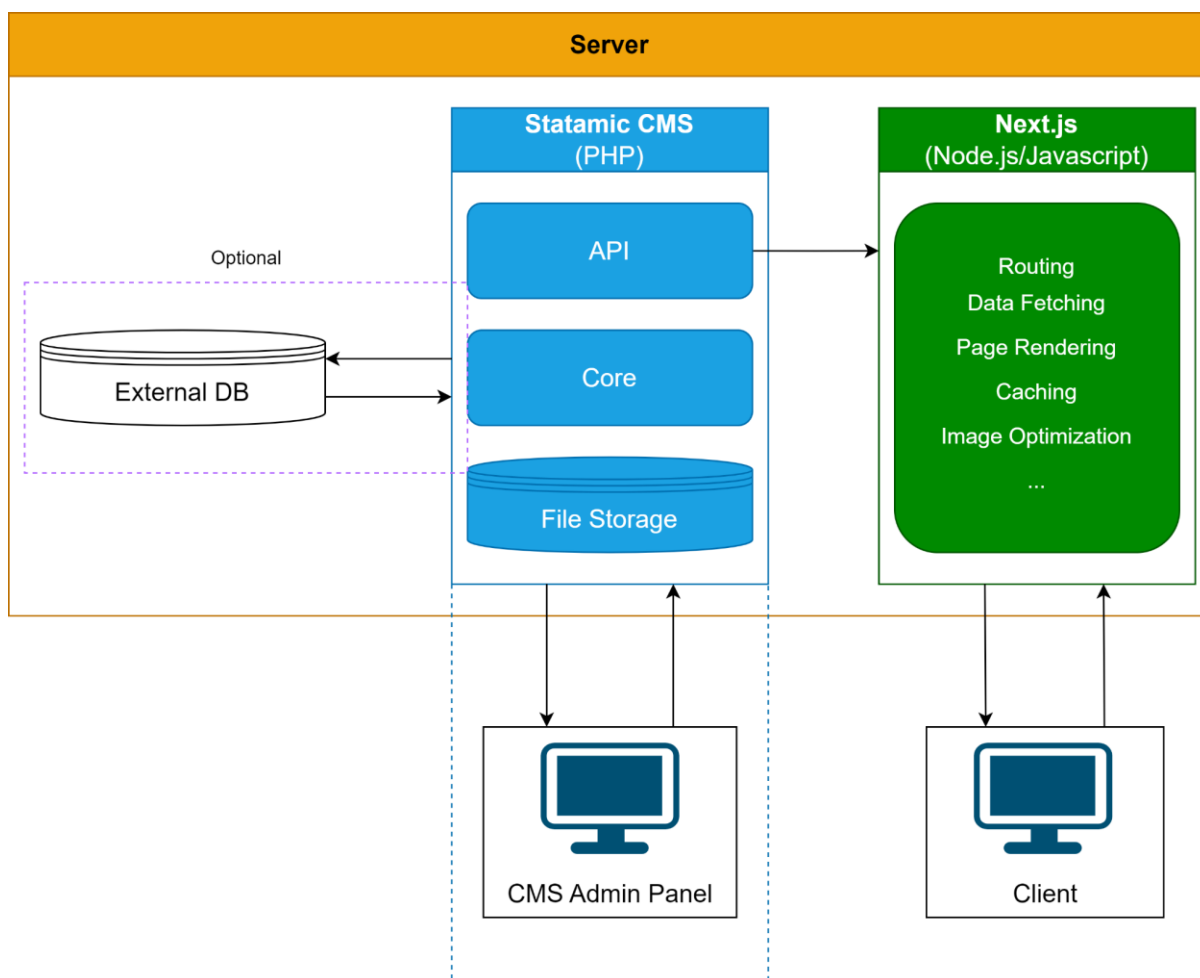


Рис. 2.2 Архітектура сайту університету - основного входу до системи університету

В обраній архітектурі пропонується технологія Headless CMS (**Headless CMS** - тіло без голови).

Логіка традиційних CMS поєднує бекенд- та фронтенд-частини однієї системи. Контент у разі виявляється пов'язаний з конкретними технологіями, архітектурою і шаблонами клієнт-серверного додатку.

**Headless CMS** - принципово інша система управління. Як правило, вона відповідає тільки за універсальний вміст, який можна використовувати на будь-яких платформах. Бекенд («тіло») за такого підходу не пов'язаний з фронтендом («головою»).

Логіка Headless CMS є такою, що до «тіла» при необхідності можна приставляти різні «голови». Це дозволяє використовувати один бекенд для керування сайтом (або сайтами) та мобільним додатком.

На рис. 2.3 показано різницю між технологією Headless CMS та звичайної CMS.

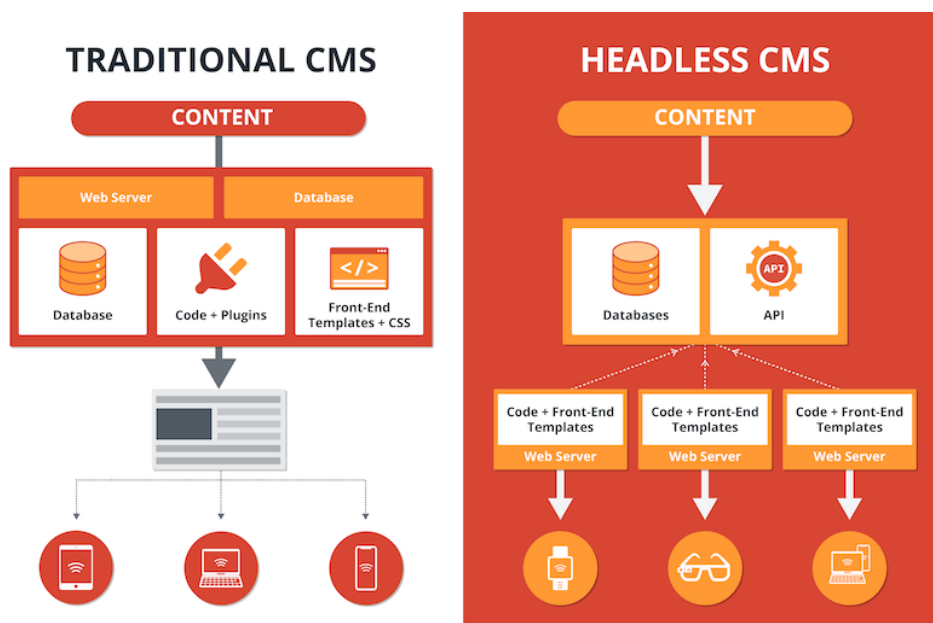


Рис. 2.3. Порівняння Headless CMS та звичайної CMS.

**Statamic** – це сучасна система керування контентом (CMS), заснована на фреймворку Laravel, призначена для створення різних типів веб-сайтів. Statamic відомий своєю гнучкістю та потужними функціями.

**Statamic** написано за допомогою фреймворку **Laravel**, на мові програмування PHP. Laravel - це безкоштовний фреймворк, доступний кожному. Це найпопулярніший фреймворк PHP спільноти. Він має дуже багато пакетів, які розширюють його функціональність.

**Laravel** – це популярний та потужний фреймворк для розробки веб-додатків мовою PHP, який славиться своїм елегантним синтаксисом та багатим набором вбудованих можливостей. Він створений для того, щоб спростити рутинні завдання розробки, пропонуючи інструменти для створення додатків, що масштабуються і підтримуються.

Для реалізації єдиного серверу аутентифікації, який обробляє всі запити на верифікацію користувачів для внутрішніх ресурсів обираємо LDAP для

базової аутентифікації (імені користувача та пароля) та підключення другого фактору, наприклад, через OTP (Time-based One-Time Password)

OAuth2/OIDC як проміжний рівень - LDAP інтегрується через сервер аутентифікації, наприклад, Keycloak або Auth0, який буде відповідальним за верифікацію. Сервер використовуватиме OAuth2 або OpenID Connect для передачі маркерів доступу (JWT) до Next.js.

## **2.2. Налаштування та інтеграція системи з університетською поштою, Moodle та Teams**

Інтеграція централізованої системи аутентифікації з такими сервісами, як пошта, Moodle та Teams, базується на реалізації єдиного входу (SSO) з використанням LDAP як основної бази для управління обліковими записами та сучасних протоколів аутентифікації (наприклад, OAuth 2.0, SAML, OpenID Connect). Поєднання цих підходів забезпечує безпечний, зручний та ефективний доступ до різних ресурсів університету.

Для інтеграції LDAP з системою пошти необхідно:

- налаштувати сервер пошти для використання LDAP як джерела для аутентифікації користувачів;
- забезпечити SSL/TLS-шифрування при підключенні до LDAP для захисту облікових даних;
- реалізувати багатофакторну аутентифікацію (MFA) через зовнішні провайдери, такі як Keycloak або Auth0. MFA можна інтегрувати як другий етап після успішного входу за допомогою LDAP.

Це дозволяє забезпечити безпечний доступ до поштових скриньок без необхідності запам'ятовувати окремі облікові дані для пошти.

Moodle підтримує LDAP та інші протоколи аутентифікації "з коробки". Для інтеграції необхідно:

- активувати плагін LDAP Authentication у Moodle [56; 62; 76].
- вказати конфігурацію LDAP-сервера (URL, Base DN, Bind DN, Bind Password).

- налаштувати синхронізацію облікових записів між LDAP і Moodle, щоб нові користувачі автоматично отримували доступ.

Для MFA можна використовувати сторонні плагіни Moodle, які додають підтримку OAuth 2.0 або OpenID Connect.

Цей підхід дозволяє інтегрувати Moodle у спільну екосистему аутентифікації, забезпечуючи доступ користувачів до навчальних матеріалів через єдиний обліковий запис.

Microsoft Teams працює в рамках роботи плагіна, який передбачає Встановіть інтеграцію Moodle з Microsoft Teams - Microsoft Teams | Microsoft Learn,

- автоматична реєстрація сервера Moodle за допомогою Microsoft Entra ID.
- розгортання бота Moodle Assistant в один клік в Azure.
- автопідготовка команд та автосинхронізація записів команд на всі або вибрані курси Moodle [42].

Додавання MFA у вигляді OTP через додатки (наприклад, Microsoft Authenticator) або інтеграцію з Keycloak/Auth0.

Єдиний мінус – вартість послуги.

Переваги інтеграції єдиний вхід (SSO): Усі сервіси інтегровані через одну систему аутентифікації, що спрощує доступ для користувачів.

Інтеграція LDAP, протоколів OAuth, SAML або OpenID Connect із системами пошти, Moodle та Teams є надійним рішенням для університету. Вона не тільки спрощує доступ до ресурсів, але й підвищує рівень безпеки всієї корпоративної інфраструктури.

### **2.3. Розробка технічних вимог до системи та специфікація її компонентів**

Розробка технічних вимог до системи аутентифікації для університетських систем базується на необхідності забезпечення безпечного, надійного та зручного доступу до ресурсів, таких як електронна пошта,

Moodle, Microsoft Teams та інші корпоративні сервіси. Для цього визначені ключові вимоги та складові компоненти системи.

#### Функціональні вимоги:

1. Надання єдиного доступу (SSO) до всіх корпоративних ресурсів.
2. Підтримка LDAP як основної бази для управління обліковими записами.
3. Можливість використання багатфакторної аутентифікації (MFA).
4. Підтримка сучасних протоколів аутентифікації (OAuth, SAML, OpenID Connect).
5. Логування всіх операцій аутентифікації для забезпечення моніторингу та аудиту.

#### Нефункціональні вимоги:

1. Висока доступність і надійність системи (аптайм не менше 99,9%).
2. Масштабованість для підтримки зростання кількості користувачів.
3. Захист персональних даних відповідно до стандартів GDPR.
4. Підтримка резервного копіювання та відновлення даних.

#### Інтеграційні вимоги:

1. Сумісність із системами електронної пошти (Microsoft Exchange, Postfix).
2. Інтеграція з LMS Moodle через LDAP або OAuth 2.0.
3. Підключення до Microsoft Teams через Azure AD або SAML.

#### Вимоги до безпеки:

1. Використання SSL/TLS для всіх з'єднань.
2. Захист від атак на облікові дані (brute force, phishing).
3. Регулярне оновлення програмного забезпечення та виправлення вразливостей.

## Специфікація компонентів системи

### LDAP-сервер:

1. Програмне забезпечення: OpenLDAP або FreeIPA.
2. Функції: централізоване управління обліковими записами, зберігання атрибутів користувачів.
3. Серверні вимоги: 2 CPU, 4 GB RAM, 50 GB SSD, підтримка резервного копіювання.

### Сервер протоколів аутентифікації:

1. Програмне забезпечення: Keycloak або Auth0.
2. Функції: реалізація SSO, підтримка OAuth, SAML, OpenID Connect.
3. Серверні вимоги: 4 CPU, 8 GB RAM, 100 GB SSD.

### Клієнтська частина (middleware):

1. Технології: Next.js (для роботи з фронтендом), інтеграція з API Keycloak/Auth0.
2. Функції: маршрутизація запитів, перевірка токенів доступу, перенаправлення до систем.

### Модулі багатофакторної аутентифікації (MFA):

1. Компоненти: OTP (одноразові паролі), Push-нотифікації, апаратні ключі (наприклад, YubiKey).
2. Інтеграція: підтримка API для зв'язку з Keycloak/Auth0.

### Логувальна система:

1. Інструменти: Elasticsearch, Kibana, Logstash (ELK Stack).
2. Функції: збирання, аналіз та збереження логів аутентифікації.

Розробка системи аутентифікації в університеті повинна враховувати вимоги безпеки, доступності та масштабованості. Її реалізація через LDAP та сучасні протоколи (OAuth, SAML) забезпечить ефективну роботу внутрішніх ресурсів і підвищить загальну зручність користувачів.

### **Висновки до розділу**

1. Обрано архітектуру системи поєднання веб-сайту університету та системи аутентифікації на основі Keycloak/Auth0 - LDAP Server + Headless CMS (**Statamic**) - Javascript Framework (**Next.js**)
2. Для керування контентом обрано систему **Statamic** написаний за допомогою фреймворку **Laravel**.
3. Показано, що розробка системи аутентифікації в університеті повинна враховувати вимоги безпеки, доступності та масштабованості.



## **РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ АУТЕНТИФІКАЦІЇ**

### **3.1. Розгортання системи для централізованої аутентифікації.**

Для початку роботи встановимо LDAP-сервер.

1. Розгортання LDAP:
2. Встановлюємо OpenLDAP.
3. Налаштовуємо базу даних з інформацією про користувачів і групи.
4. Встановлюємо схеми для розширення атрибутів.
5. Додаємо SSL/TLS для захищеного доступу до LDAP (порт 636).

Налаштуємо сервер для інтеграції: забезпечимо доступ API або REST-шару (наприклад, через LDAP Proxy API), підключимо LDAP до сервера аутентифікації. Використуємо Keycloak, який підтримує LDAP і MFA. Інтеграція LDAP: Підключаємо сервер аутентифікації до LDAP як джерела даних. Синхронізуємо групи користувачів, ролі та політики доступу.

Keycloak є централізованою платформою керування аутентифікацією, яка спрощує інтеграцію різних систем. У цій системі він виконує функції інтеграції з LDAP для управління користувачами та забезпечення багатофакторної аутентифікації (MFA).

Основні аспекти використання Keycloak:

#### **1. Інтеграція з LDAP:**

Keycloak працює як посередник між LDAP-сервером і кінцевими системами (наприклад, пошта, Moodle, Teams).

Використовуємо LDAP для синхронізації облікових записів користувачів. Це дозволяє централізовано керувати обліковими записами в LDAP, а Keycloak автоматично оновлює ці дані у своїй базі.

LDAP виступає як джерело даних користувачів (ім'я, електронна пошта, паролі), а Keycloak обробляє аутентифікацію на основі цих даних.

Приклад налаштувань у Keycloak для підключення до LDAP:

URL LDAP-сервера (наприклад, ldap://ldap.example.com).

Базовий DN (наприклад, dc=example,dc=com).

Облікові дані адміністратора LDAP для доступу.

Keycloak підтримує різні методи багатофакторної аутентифікації:

1. OTP (One-Time Password): Користувач генерує тимчасовий пароль через мобільний додаток (наприклад, Google Authenticator).
2. Підтвердження через SMS або Email: OTP надсилається на телефон або електронну пошту користувача.
3. Фізичні ключі безпеки: Використання пристроїв, сумісних із протоколами FIDO2 або U2F.

MFA додає додатковий рівень безпеки, захищаючи доступ до внутрішніх ресурсів навіть у разі компрометації основного пароля.

Keycloak забезпечує централізоване керування сесіями користувачів, дозволяючи зручно виходити з усіх систем одночасно.

Для доступу до кожної системи Keycloak видає токени (Access Token, Refresh Token, ID Token).

Access Token: Використовується для доступу до ресурсів (наприклад, API Moodle або Teams).

Refresh Token: Дозволяє отримувати нові токени без повторного входу.

Користувач входить у систему один раз через Keycloak і автоматично отримує доступ до всіх інтегрованих ресурсів (Moodle, пошта, Teams) без необхідності повторної аутентифікації.

Keycloak дозволяє створювати правила доступу залежно від ролей користувачів, їх груп у LDAP, або інших атрибутів.

Наприклад, викладачі можуть мати доступ до одних ресурсів (розклад, Teams), а студенти – до інших (Moodle, навчальні матеріали).

Для аутентифікації виконується приблизно такий алгоритм:

Користувач намагається отримати доступ до Moodle.

Запит перенаправляється на Keycloak.

Keycloak виконує аутентифікацію через LDAP і запитує додатковий фактор (наприклад, OTP).

Після успішної аутентифікації Keycloak перевіряє політику доступу і видає Access Token.

Moodle або Teams перевіряє токен і дозволяє доступ.

Дані користувачів у Keycloak періодично синхронізуються з LDAP, забезпечуючи актуальність інформації.

Для початку необхідно завантажити останню стабільну версію Keycloak із офіційного сайту. Рекомендується використовувати архівну версію для швидкого встановлення або Docker-образ для контейнеризованих середовищ. У випадку використання архіву потрібно розпакувати його у вибраний каталог, а для роботи через Docker – завантажити відповідний образ за допомогою команди `docker pull quay.io/keycloak/keycloak`.

Перед першим запуском слід налаштувати базу даних. За замовчуванням Keycloak використовує вбудовану базу H2, але для продуктивного середовища рекомендується використовувати PostgreSQL, MySQL або MariaDB. Потрібно створити базу даних, користувача та задати параметри підключення у файлі конфігурації Keycloak або передати їх як змінні середовища.

Для запуску сервера достатньо виконати команду `./kc.sh start-dev` (для архівної версії) або відповідний запуск контейнера Docker. Після запуску слід зайти на адміністративний інтерфейс за адресою `http://<hostname>:8080`. Першим кроком буде створення адміністративного облікового запису, який використовуватиметься для управління системою.

Наступним етапом є налаштування доменів (realms). Кожен домен є окремим середовищем із власними користувачами, клієнтами та політиками доступу. Необхідно створити новий домен або скористатися стандартним master. У рамках домену налаштовуються клієнти (наприклад, веб-застосунки або API), користувачі та групи, а також політики доступу.

Для інтеграції з LDAP потрібно перейти в розділ "User Federation" і додати відповідний провайдер LDAP. Тут налаштовуються параметри підключення до сервера LDAP, спосіб синхронізації користувачів і атрибути, які будуть використовуватись для аутентифікації.

Останнім етапом є налаштування протоколів аутентифікації. Для інтеграції з веб-додатками через OAuth або OpenID Connect створюються клієнти, де задаються редирект-URI, секрети та дозволи. За потреби можна налаштувати багатофакторну аутентифікацію (MFA) через відповідні політики в розділі "Authentication".

Після завершення налаштувань слід протестувати систему, перевіривши доступ користувачів і коректність авторизації для інтегрованих клієнтів. Регулярні оновлення та моніторинг роботи сервера забезпечать стабільну та безпечну роботу Keycloak.

Розглянемо як адаптувати цю під вимоги запропонованої схеми рис. 2.1.

Увійдемо в адміністративну консоль Keycloak, до розділу "Realm Settings" і створимо новий домен (realm), university. Це забезпечить розділення налаштувань університету від інших потенційних середовищ.

У розділі "Roles" додайте дві основні ролі: student та teacher. Вони будуть використовуватися для розмежування прав доступу до різних ресурсів сайту університету.

Використовуємо LDAP (наприклад, університетський сервер Active Directory), тому перейдемо до "User Federation" і додамо новий провайдер LDAP. Налаштуємо підключення до сервера, вкажемо базовий DN, атрибути для синхронізації (наприклад, uid, cn, mail) та ролі для студентів і викладачів.

У розділі "Clients" додаємо нового клієнта, який представлятиме сайт університету, наприклад, university-portal.

Задамо Client ID (university-portal) та тип Confidential для підвищення безпеки.

У полі "Redirect URI" додаємо URL, на який користувачі будуть перенаправлятися після успішної аутентифікації, наприклад, <https://university.edu/login/callback>.

Увімкнемо OpenID Connect як протокол для клієнта.

У розділі "Authorization" клієнта створюємо політики доступу, для розмежування доступу для студентів і викладачів.

Політика для студентів: дозволяє доступ лише користувачам із роллю student.

Політика для викладачів: дозволяє доступ лише користувачам із роллю teacher.

Налаштування багатофакторної аутентифікації (MFA)

У розділі "Authentication" налаштуємо потік авторизації, додавши етапи MFA (наприклад, через OTP). Це забезпечить підвищений рівень безпеки, особливо для викладачів та адміністративного персоналу.

Впровадемо на сайті університету клієнт OpenID Connect або OAuth 2.0 для взаємодії з Keycloak. Використовуємо бібліотеки для обраної мови програмування (наприклад, Keycloak JavaScript Adapter або Spring Security). Налаштуємо клієнт для надсилання запитів на аутентифікацію до Keycloak із вказаними Client ID, секретом клієнта та URL серверу Keycloak.

Для поєднання налаштування MFA з Next.js виконаємо дії:

1. Активізуємо багатофакторну аутентифікацію через сервер аутентифікації, OTP, push-нотифікацій або біометрії.
2. Налаштовуємо сервер для видачі маркерів доступу (JWT) з атрибутами користувача. Інтегруємо систему з Next.js. Використовуємо бібліотеку для OAuth2/OpenID Connect,,: next-auth для простого підключення до сервера OAuth2.
3. Використовуємо middleware для перевірки маркерів доступу (JWT) на серверній стороні Next.js. Додаємо логіку в Statamic для отримання даних, через внутрішній API (Node.js сервіс).
4. Забезпечуємо доступ до API тільки для аутентифікованих запитів (додаємо токени в заголовок).

Для LDAP-користувачів:

1. Додаємо збереження OTP-секретів у базу LDAP.

2. Реалізуємо перевірку OTP через зовнішній сервер аутентифікації.

На рівні Next.js:

Реалізуємо окремий етап MFA після основної аутентифікації.

Використовуємо сторонні бібліотеки, наприклад, `spreakeasy` (для TOTP) або інтеграцію з DUO API.

### **3.2. Реалізація доступу через LDAP та OTP.**

Реалізація доступу через LDAP та OTP (One-Time Password) у Next.js потребує наступних кроків:

#### **1. Підключення до LDAP-сервера:**

- Використовуємо бібліотеку `ldapjs` для взаємодії з LDAP-сервером.

#### **2. Реалізація OTP:**

- Використовуємо бібліотеку `otplib` для генерації та перевірки OTP.

#### **3. Інтеграція в Next.js:**

- Додаємо API-роути для обробки логіки автентифікації.

Нижче наведений приклад реалізації:

Команда `bush`:

```
npm install ldapjs otplib
```

Реалізація доступу через LDAP та OTP представлено в листингах 2.1-2.7, які представляють собою послідовні фрагменти коду

```
// /pages/api/auth.js
import ldap from 'ldapjs';
import { totp } from 'otplib';
```

Листинг 2.1 описує завантаження бібліотек.

```
// Налаштування LDAP
const LDAP_URL = 'ldap://your-ldap-server';
const LDAP_BIND_DN = 'cn=admin,dc=example,dc=com';
const LDAP_BIND_PASSWORD = 'admin_password';
const LDAP_BASE_DN = 'dc=example,dc=com';
```

Листинг 2.2 містить код для налаштування LDAP.

```
// Генерація OTP
const OTP_SECRET = 'your-secret-key'; // Зазвичай це має бути унікальний
ключ для кожного користувача

export default async function handler(req, res) {
  if (req.method === 'POST') {
    const { username, password, otp } = req.body;
```

Листинг 2.3 містить код для генерації OTP.

```
// Підключення до LDAP-сервера
const client = ldap.createClient({ url: LDAP_URL });

try {
  // Автентифікація користувача в LDAP
  await new Promise((resolve, reject) => {
    client.bind(LDAP_BIND_DN, LDAP_BIND_PASSWORD, (err) => {
      if (err) return reject(err);
      resolve();
    });
  });
}
```

Листинг 2.4 містить код для Підключення до LDAP-сервера.

```
// Пошук користувача в LDAP
const searchResult = await new Promise((resolve, reject) => {
  client.search(
    LDAP_BASE_DN,
    { filter: `(uid=${username})`, scope: 'sub' },
    (err, search) => {
      if (err) return reject(err);
      let user = null;
      search.on('searchEntry', (entry) => {
        user = entry.object;
      });
      search.on('end', () => {
```



```

        if (user) resolve(user);
        else reject(new Error('User not found'));
    });
}
);
});

```

Листинг 2.5 містить код для пошуку користувача.

Листинг 2.6

```

// Перевірка пароля користувача
await new Promise((resolve, reject) => {
    client.bind(searchResult.dn, password, (err) => {
        if (err) return reject(new Error('Invalid credentials'));
        resolve();
    });
});

```

Листинг 2.6 містить код для перевірки пароля користувача.

Листинг 2.7

```

// Перевірка OTP
if (!totp.check(otp, OTP_SECRET)) {
    throw new Error('Invalid OTP');
}

// Якщо всі перевірки пройдено
res.status(200).json({ success: true, message: 'Authentication successful'
});

```

```
    } catch (error) {  
      res.status(401).json({ success: false, message: error.message });  
    } finally {  
      client.unbind();  
    }  
  } else {  
    res.setHeader('Allow', ['POST']);  
    res.status(405).end(`Method ${req.method} Not Allowed`);  
  }  
}
```

Листинг 2.7 містить код для перевірки OTP.

Таким чином досягається:

Єдиний доступ: централізована аутентифікація дозволяє використовувати один обліковий запис для всіх внутрішніх ресурсів.

Масштабованість: легко додавати нові системи, наприклад, додати доступ до LMS (Moodle) або CRM.

Безпека: використання LDAP і MFA значно підвищує захищеність системи.

Playbook для Ansible, який автоматизує розгортання системи аутентифікації. Він охоплює встановлення та налаштування LDAP, розгортання Keycloak через Docker, а також інтеграцію Next.js middleware представлено в додатку А.

### 3.3. Забезпечення багатофакторної аутентифікації: конфігурація токенів, SMS та біометричних даних

Багатофакторна аутентифікація (MFA) дозволяє значно підвищити безпеку системи, вимагаючи від користувача підтвердження особи через кілька незалежних методів. У цьому розділі описується налаштування та конфігурація токенів, SMS-аутентифікації та біометричних даних.

Токени можна використовувати як фізичні пристрої (наприклад, YubiKey) або програмні генератори OTP (наприклад, Google Authenticator). У Keycloak для цього налаштовуються такі параметри:

Файл конфігурації для OTP

```
# conf/keycloak-realm-config.json
{
  "realm": "university",
  "otpPolicy": {
    "type": "totp",
    "lookAheadWindow": 1,
    "digits": 6,
    "initialCounter": 0,
    "algorithm": "HmacSHA1",
    "period": 30
  }
}
```

Цей файл задає політику одноразових паролів (TOTP) для MFA. Параметри включають тип алгоритму, кількість цифр у паролі та період його дії.

Для реалізації SMS-аутентифікації необхідно налаштувати сервіс відправки повідомлень (наприклад, Twilio). Keycloak дозволяє інтегрувати SMS через розширення або власний API.

## Налаштування Twilio для Keycloak

У Keycloak потрібно активувати Authenticator SPI для SMS і внести зміни в файл standalone.xml (для Wildfly).

```
<subsystem xmlns="urn:jboss:domain:keycloak-server:1.1">
  <spi name="authenticator">
    <provider name="sms-authenticator" enabled="true">
      <properties>
        <property name="twilio.accountSid" value="YourAccountSID"/>
        <property name="twilio.authToken" value="YourAuthToken"/>
        <property name="twilio.phoneNumber" value="+1234567890"/>
      </properties>
    </provider>
  </spi>
</subsystem>
```

Приклад коду для відправки SMS

python

Копировать код

```
from twilio.rest import Client

def send_sms(phone_number, otp):
    account_sid = 'YourAccountSID'
    auth_token = 'YourAuthToken'
    client = Client(account_sid, auth_token)
    message = client.messages.create(
        body=f"Your OTP is {otp}",
        from_='+1234567890',
        to=phone_number
    )
    return message.sid
```

Код генерує та надсилає одноразовий пароль на телефон користувача.

Для підтримки біометричних даних можна використовувати спеціальні сканери або програмні інтерфейси, такі як WebAuthn. Keycloak підтримує інтеграцію WebAuthn, яка дозволяє використовувати відбитки пальців, розпізнавання облич або апаратні ключі.

### Конфігурація WebAuthn у Keycloak

```
# conf/keycloak-webauthn-config.json
{
  "realm": "university",
  "webAuthnPolicy": {
    "requireResidentKey": false,
    "userVerificationRequirement": "required",
    "signatureAlgorithms": ["ES256"],
    "attestationConveyancePreference": "none",
    "authenticatorAttachment": "platform"
  }
}
```

Файл конфігурації визначає політику WebAuthn, зокрема вимоги до перевірки користувача та підтримувані алгоритми підпису.

Для інтеграції всіх методів MFA необхідно забезпечити належну взаємодію з LDAP. У Keycloak MFA налаштовується як Authentication Flow, де кожен метод додається як крок аутентифікації.

Для додавання MFA в Keycloak

Відкриваємо Authentication в адміністративній консолі Keycloak.

Створюємо новий Flow і додаємо до нього кроки:

- OTP Form
- SMS Authenticator
- WebAuthn Authenticator

Забезпечення MFA через токени, SMS і біометричні дані значно підвищує безпеку системи, відповідаючи сучасним вимогам університетського

середовища. Такий підхід дозволяє ефективно захистити ресурси, зберігаючи зручність доступу для користувачів.

### **3.4.Тестування системи в умовах, наближених до реального використання**

Тестування системи аутентифікації проводилося з метою перевірки її функціональності, безпеки та зручності використання в умовах, які максимально наближені до реального середовища університету. У тестовій інфраструктурі було розгорнуто LDAP-сервер, Keycloak як платформа керування доступом, а також інтегровані системи Moodle, корпоративна пошта та Microsoft Teams. Основні сценарії тестування включали:

Перевірка роботи однофакторної аутентифікації через LDAP:

Користувачі вводили свої облікові дані для доступу до кожної з інтегрованих систем. Система підтвердила правильність обробки облікових записів і коректність синхронізації даних з LDAP-сервером.

Тестування багатофакторної аутентифікації (MFA):

Після введення основного пароля користувачі повинні були пройти додатковий етап перевірки за допомогою OTP. Система успішно генерувала одноразові паролі та приймала їх для авторизації.

Перевірка SSO (Single Sign-On):

Користувач входив у систему через Keycloak і отримував доступ до всіх ресурсів без повторної аутентифікації. Усі запити до Moodle, пошти та Teams автоматично використовували токени доступу, видані Keycloak.

Сценарії помилкової аутентифікації:

Перевірялося, як система реагує на введення неправильних облікових даних, недійсних OTP або спроби доступу до ресурсів без авторизації. Усі подібні запити блокувалися, а користувач отримував відповідне повідомлення.

Навантажувальне тестування:

Було перевірено, як система працює при великій кількості одночасних запитів, що імітує періоди підвищеної активності, наприклад, під час сесії. Система продемонструвала стабільну роботу без збоїв.

Методика - одночасно підключаються 100 абонентів з різних пристроїв та з різними швидкостями інтернет з'єднання. Фіксується відсоток успішних аутентифікацій та час відповіді системи на запити.

Для об'єктивності в варто врахувати різні категорії пристроїв:

Смартфони:

- моделі: бюджетні, середнього рівня, флагмани.
- операційні системи: Android (різних версій), iOS.
- пам'ять: від 2 ГБ (низька продуктивність) до 12 ГБ (висока продуктивність).
- процесори: слабкі (1-2 ядра) та потужні (8 ядер).

Планшети та ноутбуки:

середньої потужності пристрої з Windows, macOS або Linux.

Пристрої зі старішими процесорами.

Смарт-гаджети:

Smart TV, IoT-пристрої (з обмеженою потужністю та специфічними ОС).

Обирались різні мережі з різними параметрами.

Швидкість з'єднання:

1. Високошвидкісний доступ: 50-100 Мбіт/с (оптоволоконний інтернет).
2. Середня швидкість: 10-30 Мбіт/с (ADSL, 4G).
3. Низька швидкість: 1-5 Мбіт/с (старі 3G-з'єднання).

Тип мережі:

1. Мобільні (3G, 4G, 5G).
2. Wi-Fi (з різною пропускнуою здатністю).
3. Провідний інтернет.

Затримка:

1. Низька (5-20 мс, стабільне з'єднання).
2. Середня (50-100 мс, мобільні мережі).
3. Висока (200-500 мс, супутникове з'єднання).

Втрати пакетів:

1. Відсутні (стабільне з'єднання).
2. Втрата 5-10% пакетів (нестабільний мобільний інтернет).

Таблиця 3.1

### Результати тестування за критерієм категорії пристроїв

	Затримка (середня, мс)	Втрати пакетів (%)
Смартфони: пам'ять від 2 ГБ (низька продуктивність).	63	2
Смартфони: пам'ять до 12 ГБ (висока продуктивність).	33	2
Смартфони: процесори: слабкі (1-2 ядра)	72	2
Смартфони: процесори потужні (8 ядер)	35	2
Планшети та ноутбуки середньої потужності.	25	0
Смарт-гаджети	81	5

Таблиця 3.1

### Результати тестування за критерієм швидкості з'єднання

Швидкість	Затримка (середня, мс)	Втрати пакетів (%)
Високошвидкісний доступ: 50-100 Мбіт/с (оптоволоконний інтернет).	34	0
Середня швидкість: 10-30 Мбіт/с (ADSL, 4G).	52	2



Низька швидкість: 1-5 Мбіт/с (старі 3G- з'єднання).	65	3
---	----	---

Як бачимо з табл. 3.1 та табл. 3.2 система показала в цілому низьку та середню затримку та стабільне з'єднання.

Типи помилок, які були під час тестування: таймаут, відхилення запиту через перевантаження, некоректна відповідь. Але їх кількість була менше 5%.

У результаті тестування підтверджено, що система аутентифікації забезпечує надійний та безпечний доступ до університетських ресурсів. Вона задовольняє вимоги користувачів щодо зручності, а також відповідає встановленим стандартам інформаційної безпеки.

### **Висновки до розділу**

1. Показано процес розгортання системи: LDAP/ OPENLDAP, бази даних користувачів та захищеного доступу.
2. Показано як налаштувати Keycloak в інтеграції із LDAP.
3. Показано, як поєднати налаштування MFA з Next.js.
4. Показано реалізацію доступу через LDAP та OTP (One-Time Password) у Next.js
5. Описано налаштування системи та конфігурація токенів, SMS-аутентифікації та біометричних даних
6. Протестоване систему в умовах наближених до реального використання, підтверджено, що система аутентифікації забезпечує надійний та безпечний доступ до університетських ресурсів..

## **РОЗДІЛ 4. ВПРОВАДЖЕННЯ ТА ПЕРСПЕКТИВИ ДОСЛІДЖЕННЯ**

### **4.1. Керівництво користувача щодо використання системи аутентифікації**

Для зручності було розроблено керівництво користувача щодо використання системи аутентифікації. Система аутентифікації забезпечує захист ваших даних і контроль доступу до сервісів. Вона гарантує, що тільки авторизовані користувачі мають доступ до облікових записів і функцій.

Зформульовано вимоги до користувача

Пристрої: Смартфон, планшет, ноутбук або ПК з підтримкою сучасного браузера.

Інтернет-з'єднання: Мінімальна швидкість 1 Мбіт/с для стабільної роботи.

Облікові дані: Логін (e-mail або номер телефону) та пароль, надані під час реєстрації.

Щоб розпочати роботу відкрийте браузер і перейдіть за адресою системи).

На екрані входу введіть логін у відповідне поле, введіть пароль, дотримуючись регістру (великі та маленькі літери). Натисніть кнопку "Увійти".

Якщо використовується двофакторна аутентифікація (2FA): введіть одноразовий код, який надійде на ваш e-mail або у SMS. Натисніть "Підтвердити".

Забули пароль? На екрані входу натисніть "Забули пароль?".

Введіть e-mail або номер телефону, пов'язаний із вашим обліковим записом.

Дотримуйтеся інструкцій, які ви отримаєте на вказану адресу (наприклад, посилання для скидання пароля).

Для перевірки активних сеансів перейдіть у розділ "Безпека" у вашому обліковому записі.

Ознайомтеся зі списком пристроїв, які наразі ввійшли у ваш обліковий запис. За необхідності натисніть "Вийти", щоб завершити сесію на небажаному пристрої.

Для заміни пароля увійдіть у розділ "Налаштування" > "Пароль".

Введіть поточний пароль. Введіть новий пароль, що відповідає вимогам (мінімум 8 символів, включно з цифрами, великими літерами та спеціальними символами). Натисніть "Зберегти".

Для Увімкнення 2FA у розділі "Безпека" активуйте опцію "Двофакторна аутентифікація".

Виберіть спосіб підтвердження (через додаток, SMS або e-mail). Дотримуйтеся інструкцій для налаштування.

Рекомендації з безпеки:

- Не передавайте свої облікові дані стороннім особам.
- Використовуйте унікальний пароль для кожного сервісу.
- Регулярно оновлюйте пароль (не рідше ніж раз на 6 місяців).
- Перевіряйте підозрілі дії у вашому обліковому записі та повідомляйте про них адміністрацію.

Таблиця 3.3.

**Типові проблеми та способи їх вирішення**

Проблема	Можливе вирішення
Забули пароль	Скористайтесь функцією "Забули пароль?" для його скидання.
Не приходить код підтвердження	Перевірте правильність e-mail/номера телефону. Переконайтеся, що ваше інтернет-з'єднання стабільне.
Вхід заблоковано через помилки	Зачекайте 10 хвилин та спробуйте знову. Якщо проблема повторюється, зверніться до служби підтримки.

## **4.2. Інструкція для адміністраторів системи аутентифікації**

### **1. Призначення системи**

Система аутентифікації забезпечує захист доступу до сервісів та контроль за обліковими записами користувачів. Адміністратори відповідають за налаштування, моніторинг та підтримку її працездатності.

### **2. Основні обов'язки адміністратора**

Забезпечення стабільної роботи системи аутентифікації.

Управління обліковими записами користувачів.

Налаштування безпеки системи.

Моніторинг продуктивності та усунення технічних проблем.

Реагування на підозрілі дії або загрози.

### **3. Початкове налаштування системи**

#### **3.1 Налаштування серверів**

Операційна система:

Переконайтеся, що ОС на сервері оновлена до останньої стабільної версії.

Забезпечте налаштування firewall і блокування небажаних портів.

Установка системи:

Встановіть компоненти системи аутентифікації відповідно до документації (наприклад, сервер авторизації, базу даних, API).

Перевірте коректність з'єднання між компонентами.

Сертифікати безпеки:

Встановіть SSL/TLS-сертифікати для захищеного з'єднання.

Налаштуйте автоматичне оновлення сертифікатів (наприклад, через Let's Encrypt).

#### **3.2 Налаштування бази даних**

Створіть базу даних для зберігання облікових записів і логів.

Використовуйте сучасні алгоритми хешування паролів (наприклад, bcrypt).

Обмежте доступ до бази даних через ролі та права доступу.

## 4. Управління обліковими записами

### 4.1 Додавання нового користувача

Увійдіть у панель адміністратора.

Перейдіть у розділ "Управління користувачами".

Натисніть "Додати користувача" і введіть наступні дані:

Логін (e-mail або телефон).

Тимчасовий пароль.

Ролі (користувач, адміністратор тощо).

Збережіть дані, після чого користувач отримає інструкції на e-mail.

### 4.2 Редагування облікових записів

Знайдіть потрібного користувача у списку.

Ви можете:

Скинути пароль.

Блокувати обліковий запис.

Змінити роль користувача.

### 4.3 Видалення облікових записів

Оберіть обліковий запис у списку.

Переконайтеся, що видалення не вплине на залежні сервіси.

Підтвердіть видалення.

## 5. Моніторинг системи

### 5.1 Логи

Логи входу:

Перевіряйте успішні/невдалі спроби входу користувачів.

Виявляйте підозрілі дії, такі як багаторазові невдалі спроби входу з одного IP.

Системні логи:

Аналізуйте помилки у роботі серверів або баз даних.

Використовуйте інструменти для централізованого збору логів (наприклад, ELK Stack).

## 5.2 Моніторинг продуктивності

Відстежуйте навантаження на сервери (CPU, RAM, мережа).

Використовуйте інструменти моніторингу, такі як Zabbix або Prometheus.

Аналізуйте час відповіді системи (особливо під час пікових навантажень).

## 6. Налаштування безпеки

### 6.1 Політика паролів

Встановіть вимоги до складності паролів:

Мінімум 8 символів, включаючи цифри, великі літери та спеціальні символи.

Налаштуйте періодичне сповіщення користувачів про зміну пароля.

### 6.2 Захист від атак

Увімкніть захист від brute force (обмеження кількості спроб входу).

Встановіть капчу після кількох невдалих спроб входу.

Налаштуйте обмеження доступу до адміністративного інтерфейсу (наприклад, за IP-адресами).

### 6.3 Двофакторна аутентифікація

Увімкніть 2FA для користувачів і адміністраторів.

Перевіряйте регулярність оновлення токенів та коректність роботи 2FA.

## 7. Резервування та відновлення

### 7.1 Резервне копіювання

Автоматично створюйте резервні копії бази даних та конфігурацій системи щодня.

Зберігайте резервні копії у захищеному середовищі (наприклад, у хмарному сховищі з шифруванням).

### 7.2 Відновлення системи

У разі збою використовуйте останню доступну резервну копію.

Виконайте перевірку цілісності даних після відновлення.

**Типові проблеми та їх вирішення**

<b>Проблема</b>	<b>Можливе вирішення</b>
Користувач не може увійти	Перевірте, чи активний обліковий запис, або скиньте пароль.
Затримка відповіді системи	Перевірте завантаження серверів, оптимізуйте запити до бази даних.
Сервер не відповідає	Переконайтеся, що сервер запущено, і перевірте мережеве підключення.

#### **4.3. Перспективи розширення функціональності системи, масштабування системи в інших установах**

Перспективи розширення функціональності створеної системи аутентифікації включають інтеграцію з іншими сервісами, які використовуються в університетській інфраструктурі. Це може бути реалізовано шляхом підключення системи до інших платформ управління навчальним процесом, забезпечення єдиного входу (SSO) для доступу до електронної пошти, файлових сховищ і хмарних сервісів. Також можливо впровадження підтримки протоколів інтеграції з HR-системами для автоматичного керування обліковими записами співробітників та студентів, зокрема їх створення, оновлення або видалення на основі змін у внутрішніх базах даних. Окрім цього, перспективним напрямом є інтеграція з системами фізичного доступу до приміщень (наприклад, електронні перепустки) та автоматизованими бібліотечними системами, що дозволить використовувати єдину аутентифікацію для розширеного спектра послуг. Для забезпечення високого рівня безпеки також варто розглянути впровадження моніторингу активності користувачів та інтеграцію з SIEM-системами для виявлення аномалій і оперативного реагування на потенційні загрози.

Для успішного впровадження та масштабування системи аутентифікації в інших установах рекомендується розпочати з детального аналізу потреб і особливостей існуючої IT-інфраструктури, включаючи кількість користувачів, типи сервісів, що використовуються, і рівень безпеки, необхідний для захисту даних. Важливо забезпечити гнучкість архітектури системи, щоб вона могла легко інтегруватися з існуючими платформами та підтримувати різні протоколи аутентифікації, такі як OAuth, SAML та OpenID Connect. Для масштабування варто відразу закласти можливість розподіленої роботи системи, наприклад, через кластеризацію сервера Keycloak і використання балансувальників навантаження. Для зменшення залежності від одного постачальника технологій доцільно використовувати рішення з відкритим кодом, такі як FreeIPA, з можливістю адаптації під специфічні потреби установи. Рекомендується також впровадити механізми резервного копіювання конфігурацій і даних користувачів, а також тестувати систему в умовах, що моделюють реальні навантаження. Окрему увагу слід приділити навчанню IT-персоналу установи для підтримки та розвитку системи, а також розробити інструкції для кінцевих користувачів. З метою безпеки варто запровадити багатофакторну аутентифікацію та регулярно оновлювати компоненти системи відповідно до останніх версій.

### **Висновки до розділу**

1. Розроблено інструкцію користувача системи.
2. Розроблено інструкцію адміністратора системи.
3. Сформульовано перспективи розвитку системи.



## ВИСНОВКИ

В умовах коли постійно збільшується кількість веб ресурсів, які використовують викладачі та студенти, важливу роль мають системи аутентифікації. Їх значущість обумовлена багатьма факторами: безпека та достовірність ідентифікації важлива для академічної доброчесності та надійності верифікації процесів навчання; надійний та зручний доступ до цифрових ресурсів дозволяє зробити університетську цифрову екосистему більш ефективною внаслідок зменшення витрат часу для виконання рутинних операцій, спрощує доступ до ресурсів та зменшує людські витрати персоналу щодо технічного супроводу.

У роботі проаналізовано методи аутентифікації, протоколи OAuth, SAML, OpenID Connect, а також популярні рішення для централізованої аутентифікації (Okta, Keycloak, FreeIPA). Визначено вимоги до системи аутентифікації для університету, розроблено архітектуру системи з інтеграцією LDAP та Keycloak для підтримки MFA. Проведено тестування створеної системи в умовах, наближених до реального використання.

У першому розділі завдяки аналізу та порівнянню основних методів аутентифікації (однофакторні, багатофакторні та біометричні підходи) вдалося сформулювати вимоги до систем аутентифікації в університеті. Проаналізовано основні методи аутентифікації: однофакторні, багатофакторні та біометричні підходи. Показано, що біометричні методи на сьогодні достатньо легко реалізуються, наведено методику простої реалізації, єдиною перепорою при масштабному використанні є вартість.

Запропоновано використання для університетських систем багатофакторної аутентифікації. Зроблено порівняльний аналіз протоколів аутентифікації OAuth, SAML та OpenID Connect. Показано, що різні протоколи мають різну ефективність за різними прамаетрами. OAuth краще використовувати для мобільних додатків, SAML для створення єдиної точки входу, OpenID Connect – для хмарних сервісів. Такі комерційні рішення, як Microsoft Active Directory Federation Services (AD FS) або Microsoft Entra ID

(більш сучасне рішення), Okta, Ping Identity, Auth0 мають гнучкість налаштування, багато можливостей для інтеграції з різними системами. На основі критерію доступності систем з відкритим кодом обрано Keycloak та LDAP. Keycloak підтримує протоколи OAuth 2.0, OpenID Connect і SAML, що дозволяє реалізувати функції єдиного входу (SSO) і багатофакторної аутентифікації. Система забезпечує кастомізацію інтерфейсу входу, а також легку інтеграцію з LDAP-серверами та зовнішніми провайдерами ідентифікації. Keycloak є гнучким і масштабованим рішенням, що робить його популярним вибором для університетських середовищ. Сформульовано основні критерії ефективності системи аутентифікації : безпека, зручність використання, сумісність, масштабованість адміністрування, вимоги до конфіденційності.

Для розробки системи аутентифікації університету обрана архітектура:

7. **Клієнт** – пристрій користувача
8. **LDAP Server** → централізована база користувачів.
9. **Keycloak/Auth0** → аутентифікація та видача маркерів доступу.
10. **Next.js Middleware** → перевірка маркерів доступу.
11. **Statamic** → обробка контенту для авторизованих користувачів.
12. **Модуль** – плагін, написаний на PHP к якому надається доступ, або зовнішня система (Teams, Moodle, пошта ...).

Для розробки сайту університету обрано архітектуру: Headless CMS (**Statamic**) + Javascript Framework (**Next.js**) з реалізацією за допомогою **Statamic, Laravel**.

Показано, що розробка системи аутентифікації в університеті повинна враховувати вимоги безпеки, доступності та масштабованості.

Для аутентифікації обрано такий алгоритм:

1. Користувач намагається отримати доступ до Moodle.
2. Запит перенаправляється на Keycloak.

3. Keycloak виконує аутентифікацію через LDAP і запитує додатковий фактор (наприклад, OTP).

Після успішної аутентифікації Keycloak перевіряє політику доступу і видає Access Token.

Moodle або Teams перевіряє токен і дозволяє доступ.

Дані користувачів у Keycloak періодично синхронізуються з LDAP, забезпечуючи актуальність інформації.

Реалізація доступу через LDAP та OTP (One-Time Password) у Next.js потребує наступних кроків: підключення до LDAP-сервера, реалізація OTP, інтеграція в Next.js. Доступ через LDAP та OTP реалізовано в Python за допомогою бібліотек **ldapjs** та **otplib**.

У результаті тестування підтверджено, що система аутентифікації забезпечує надійний та безпечний доступ до університетських ресурсів. Вона задовольняє вимоги користувачів щодо зручності, а також відповідає встановленим стандартам інформаційної безпеки.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. "A Survey of Image-Based Authentication Methods," IEEE Access, vol. 8, pp. 104558-104571, 2020.
2. "Artificial Intelligence for Cybersecurity: A Survey," ACM Computing Surveys, vol. 53, no. 3, pp. 1-35, Mar. 2021.
3. "Artificial Intelligence," Wikipedia, The Free Encyclopedia. [Online]. Available: [https://en.wikipedia.org/wiki/Artificial\\_intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence). [Accessed: 24-Aug-2024].
4. "Authentication," Wikipedia, The Free Encyclopedia. [Online]. Available: <https://en.wikipedia.org/wiki/Authentication>. [Accessed: 24-Aug-2024].
5. "Biometrics," Wikipedia, The Free Encyclopedia. [Online]. Available: <https://en.wikipedia.org/wiki/Biometrics>. [Accessed: 24-Aug-2024].
6. "Security Engineering," Wikipedia, The Free Encyclopedia. [Online]. Available: [https://en.wikipedia.org/wiki/Security\\_engineering](https://en.wikipedia.org/wiki/Security_engineering). [Accessed: 24-Aug-2024].
7. Akar, E., & Mardiyani, S. (2016). Analyzing factors affecting the adoption of cloud computing: A case of Turkey. KSII Transactions on Internet and Information Systems, 10(1). <https://doi.org/10.3837/tiis.2016.01.002>
8. Alex, W. (2022). Advanced Microsoft Authenticator security features are now generally available! - Microsoft Community Hub. Microsoft . <https://techcommunity.microsoft.com/t5/microsoft-entra-azure-ad-blog/advanced-microsoft-authenticator-security-features-are-now/ba-p/2365673>
9. Alhakami, H. (2020). Knowledge based authentication techniques and challenges. International Journal of Advanced Computer Science and Applications, 11(2).
10. Alsunaidi, S. J., Saqib, N. A., & Alissa, K. A. (2020). A comparison of human brainwaves-based biometric authentication systems. International Journal of Biometrics, 12(4), 411–429. <https://doi.org/10.1504/IJBM.2020.110814>
11. Anand, S. (2015). Research and analysis on improving mobile application security by using multi-level authentication including Image Based Authentication
12. Arduino official store | boards shields kits accessories. *Arduino Official Store*. URL: <https://store.arduino.cc/> (date of access: 10.12.2024).
13. Auth0: secure access for everyone. but not just anyone. *Auth0*. URL: <https://auth0.com/> (date of access: 10.12.2024).

14. Badeges, W., & Fauzi, M. N. (2020). Implementasi Multi Factor Authentication Pada Phpmyadmin. 35–39.
15. Bae, Y., Banerjee, S., Lee, S., & Peinado, M. (2022). Spacelord: Private and Secure Smart Space Sharing. *ACM International Conference Proceeding Series*, 427–439. <https://doi.org/10.1145/3564625.3564637>
16. Baldin, I., Chase, J., Crabtree, J., Nechyba, T., Christopherson, L., Stealey, M., Kneifel, C., Orlikowski, V., Carter, R., Scott, E., Sone, A., & Sizemore, D. (2022). ImPACT: A networked service architecture for safe sharing of restricted data. *Future Generation Computer Systems*, 129, 269–285. <https://doi.org/10.1016/j.future.2021.11.026>
17. Bello, O., & Olanrewaju, O. (2022). Factors influencing biometric technology adoption: Empirical evidence from Nigeria. *African Journal of Science, Technology, Innovation and Development*, 14(2), 392–404. <https://doi.org/10.1080/20421338.2020.1837415>
18. Biometrics. URL: <https://www.dhs.gov/biometrics> (дата звернення 20.03.2024 р.).
19. Buccafurri, F., De Angelis, V., Lazzaro, S., & Pugliese, A. (2024). Enforcing security policies on interacting authentication systems. *Computers and Security*, 140(October 2023), 103771. <https://doi.org/10.1016/j.cose.2024.103771>
20. Cherry, D. (2022). Multi-Factor Authentication. *Enterprise-Grade IT Security for Small and Medium Businesses*, 83–96. [https://doi.org/10.1007/978-1-4842-8628-9\\_7](https://doi.org/10.1007/978-1-4842-8628-9_7)
21. Ciolino, S., Parkin, S., & Dunphy, P. (2019). Of two minds about two-factor: Understanding everyday FIDO U2F usability through device comparison and experience sampling. *Proceedings of the 15th Symposium on Usable Privacy and Security, SOUPS 2019*.
22. *Computer and Information Sciences*, 35(9), 101788. <https://doi.org/10.1016/j.jksuci.2023.101788>
23. CoSign: Secure, Intra-Institutional Web Authentication. - 2004. - Режим доступу: <http://www.umich.edu/~umweb/software/cosign/>. - Заголовок з екрану.
24. Covavisaruch, N. (2006). Personal identification system using hand geometry and iris pattern fusion. *IEEE International Conference on Electro/Information Technology*, 597–602. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=4017768](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4017768)
25. D. Hardt, RFC 6749: The OAuth 2.0 authorization framework, 2012.

- 26.Das, S., Wang, B., Tingle, Z., & Camp, L. J. (2019). Evaluating User Perception of Multi-Factor Authentication: A Systematic Review. <http://arxiv.org/abs/1908.05901>
- 27.Dermawan, I., Baidawi, A., Iksan, & Mellyana Dewi, S. (2023). Serangan Cyber dan Kesiapan Keamanan Cyber Terhadap Bank Indonesia. *Jurnal Informasi Dan Teknologi*, 5(3), 20–25. <https://doi.org/10.60083/jidt.v5i3.364>
- 28.Dong, Y., Guo, W., Chen, Y., Xing, X., Zhang, Y., & Wang, G. (2019). Towards the detection of inconsistencies in public security vulnerability reports. *Proceedings of the 28th USENIX Security Symposium*.
- 29.Drager, N. (2021). Which Method of Multi-Factor Authentication is Most Secure? (and Other MFA Considerations). *QUANTUM TECHNOLOGIES*. <https://quantumpc.com/mfa-most-secure/>
- 30.Erdem, E., & Sandikkaya, M. T. (2018). OTPaaS-One time password as a service. *IEEE Transactions on Information Forensics and Security*, 14(3). <https://doi.org/10.1109/TIFS.2018.2866025>
- 31.FIDO2 Passwordless Authentication | YubiKey |. (2023). Yubico. <https://www.yubico.com/authentication-standards/fido2/>
- 32.Fitrisia Munir, Irfan Nursetiawan, Yuniana Cahyaningrum, Hermi Oppier, S. S. (2023). Kebijakan Publik di Era Digital. In CV. Karsa Cendekia. <http://www.nber.org/papers/w16019>
- 33.FreeIPA - identity, policy, audit – freeipa documentation. *FreeIPA - Identity, Policy, Audit – FreeIPA documentation*. URL: [https://www.freeipa.org/page/Main\\_Page](https://www.freeipa.org/page/Main_Page) (date of access: 10.12.2024).
- 34.Furnell, S., Khern-am-nuai, W., Esmael, R., Yang, W., & Li, N. (2018). Enhancing security behaviour by supporting the user. *Computers and Security*, 75, 1–9. <https://doi.org/10.1016/j.cose.2018.01.016>
- 35.Ghorbani Lyastani, S., Schilling, M., Neumayr, M., Backes, M., & Bugiel, S. (2020). Is FIDO2 the kingslayer of user authentication? a comparative usability study of FIDO2 passwordless authentication. *Proceedings - IEEE Symposium on Security and Privacy*, 2020-May. <https://doi.org/10.1109/SP40000.2020.00047>
- 36.Gunson, N., Marshall, D., Morton, H., & Jack, M. (2011). User perceptions of security and usability of single-factor and two-factor authentication in automated telephone banking. *Computers and Security*, 30(4). <https://doi.org/10.1016/j.cose.2010.12.001>

37. Gupta, C., & Varshney, G. (2023). An improved authentication scheme for BLE devices with no I/O capabilities. *Computer Communications*, 200. <https://doi.org/10.1016/j.comcom.2023.01.001>
38. Hall, J. F. J. (2023). OATH tokens authentication method. Microsoft Entra | Microsoft Learn. <https://learn.microsoft.com/en-us/azure/active-directory/authentication/concept-authentication-oath-tokens>
39. Hall, J., Khader, Tamara, F. (2023). Azure AD Multi-Factor Authentication Overview. Microsoft Entra | Microsoft Learn. <https://learn.microsoft.com/en-us/azure/active-directory/authentication/concept-mfa-howitworks>
40. Heidari, H., & Chalechale, A. (2022). Biometric authentication using a deep learning approach based on different level fusion of finger knuckle print and fingernail. *Expert Systems with Applications*, 191. <https://doi.org/10.1016/j.eswa.2021.116278>
41. Identity security for the digital enterprise. *Identity Security for the Digital Enterprise* / Ping Identity. URL: <https://www.pingidentity.com/en.html> (date of access: 10.12.2024).
42. Install moodle integration with microsoft teams - microsoft teams. *Microsoft Learn: Build skills that open doors in your career*. URL: <https://learn.microsoft.com/en-us/microsoftteams/install-moodle-integration> (date of access: 10.12.2024).
43. Irawan, B., Sani, I., Febrian, W. D., Setiawan, Z., Abdullah, A., Aprizal, Wasil, M., Suseno, D. A. N., Rahayu, N., Soeharjoto, Umar, N., Chasanah, S., Bilgies, A. F., & Harinie, L. T. (2022). Konsep Dasar E-Business. Kaiser, T., Siddiqua, R., Hasan, M., & Uddin, M. (2022). A multi-layer security system for data access control, authentication, and authorization. May.
44. JAIN, Anil K., et al. An identity-authentication system using fingerprints. *Proceedings of the IEEE*, 1997, 85.9: 1365-1388.
45. Jover, R. P. (2020). Security analysis of SMS as a second factor of authentication. *Communications of the ACM*, 63(12), 46–52. <https://doi.org/10.1145/3424260>
46. Karim, N. A., & Shukur, Z. (2015). Review of user authentication methods in online examination. *Asian Journal of Information Technology*, 14(5), 166-175. <https://doi.org/10.3923/ajit.2015.166-175>
47. Karim, N. A., Kanaker, H., Almasadeh, S., & Zargou, J. (2021). A Robust User Authentication Technique in Online Examination. *International Journal of Computing*, 20(4), 535–542. <https://doi.org/10.47839/ijc.20.4.2441>

48. Karim, N. A., Shukur, Z., & Albanna, A. E. M. (2020). UIPA: User authentication method based on user interface preferences for account recovery process. *Journal of Information Security and Applications*, 52. <https://doi.org/10.1016/j.jisa.2020.102466>
49. KARIM, N., et al. Choosing the right MFA method for online systems: A comparative analysis. *International Journal of Data and Network Science*, 2024, 8.1: 201-212.
50. Keycloak. *Keycloak*. URL: <https://www.keycloak.org/> (date of access: 10.12.2024).
51. Khan, R. H., & Miah, J. (2022). Performance Evaluation of a new one-Time password (OTP) scheme using stochastic petri net (SPN). 2022 IEEE World AI IoT Congress, AIIoT 2022, 407–412. <https://doi.org/10.1109/AIIOT54504.2022.9817203>
52. Kim, S., Mun, H. J., & Hong, S. (2022). Multi-Factor Authentication with Randomly Selected Authentication Methods with DID on a Random Terminal. *Applied Sciences (Switzerland)*, 12(5). <https://doi.org/10.3390/app12052301>
53. Kokila, M., & Reddy K, S. (2024). Authentication, Access Control and Scalability models in Internet of Things Security - A Review. *Cyber Security and Applications*, 3(March 2024), 100057. <https://doi.org/10.1016/j.csa.2024.100057>
54. Komalasari, R. (2018). KESADARAN AKAN KEAMANAN PENGGUNAAN USERNAME DAN PASSWORD. *TEMATIK - Jurnal Teknologi Informasi Dan Komunikasi*, 5(2), 141–152.
55. Komarova, A., Menshchikov, A., Negols, A., Korobeynikov, A., Gatchin, Y., & Tishukova, N. (2018). Comparison of authentication methods on web resources. *Advances in Intelligent Systems and Computing*, 679, 104–113. [https://doi.org/10.1007/978-3-319-68321-8\\_11](https://doi.org/10.1007/978-3-319-68321-8_11)
56. LDAP authentication - MoodleDocs. *MoodleDocs*. URL: [https://docs.moodle.org/402/en/LDAP\\_authentication#:~:text=An%20administrator%20can%20enable%20LDAP%20authentication%20as%20follows:,enabled,%20it%20will%20no%20longer%20be%20greyed%20out.](https://docs.moodle.org/402/en/LDAP_authentication#:~:text=An%20administrator%20can%20enable%20LDAP%20authentication%20as%20follows:,enabled,%20it%20will%20no%20longer%20be%20greyed%20out.) (date of access: 10.12.2024).
57. Learn all about biometrics and how to build a security system that uses your fingerprints as the key in this tutorial. URL: <https://maker.pro/raspberry->



- pi/projects/raspberry-pi-fingerprint-scanner-using-a-usb-to-serial-ttl-converter (дата звернення 08.04.2024 р.)
58. Lee, Y. K., & Jeong, J. (2021). Securing biometric authentication system using blockchain. *ICT Express*, 7(3). <https://doi.org/10.1016/j.ict.2021.08.003>
  59. M. G. de Almeida, E. D. Canedo, Authentication and authorization in microservices architecture: A systematic literature review, *Applied Sciences* 12 (2022).
  60. M. Jones, J. Bradley, N. Sakimura, RFC 7519: JSON Web Token (JWT), 2015.
  61. M. Knutson, R. Winch, P. Mularien, *Spring Security: Secure your web applications, RESTful services, and microservice architectures*, Packt Publishing Ltd, 2017.
  62. M. Rouse, *Ldap (lightweight directory access protocol)*, Enterprise Mobile Computing news and information (2019).
  63. MATHIS, Florian; VANIEA, Kami; KHAMIS, Mohamed. Replicueauth: Validating the use of a lab-based virtual reality setup for evaluating authentication systems. In: *Proceedings of the 2021 chi conference on human factors in computing systems*. 2021. p. 1-18.
  64. Maynes, M. (2019). One simple action you can take to prevent 99.9 percent of attacks on your accounts. <https://www.microsoft.com/en-us/security/blog/2019/08/20/one-simple-action-you-can-take-to-prevent-99-9-percent-of-account-attacks/>
  65. Microcosm. (2023). One-Time Password (OTP) Tokens | OATH-compliant Authentication Tokens, Keypads and Cards. <https://www.microcosm.com/it-security-hardware/oath-otp-authentication-tokens>
  66. Microsoft entra - secure identities and access | microsoft security. *Your request has been blocked. This could be due to several reasons*. URL: <https://www.microsoft.com/en-us/security/business/microsoft-entra?msocid=0066d0ce224265b12892c13823506404> (date of access: 10.12.2024).
  67. Microsoft. (2023). Azure Active Directory Pricing. Microsoft Security. <https://www.microsoft.com/en-us/security/business/identity-access/azure-active-directory-pricing>

68. Miftahul Jannah, Y. A. E. (2023). *Arsitektur dan Organisasi Komputer* (M. M. Artika Arsita, S.Kom. (ed.); 1st ed.). PT Penamuda Media
69. Mishra, R. A., Kalla, A., Braeken, A., & Liyanage, M. (2021). Privacy Protected Blockchain Based Architecture and Implementation for Sharing of Students' Credentials. *Information Processing and Management*, 58(3), 102512. <https://doi.org/10.1016/j.ipm.2021.102512>
70. MITRE. (2023). CVE security vulnerability database. Security vulnerabilities, exploits, references and more. <https://www.cvedetails.com/>.
71. Mohanakrishnan, R. (2021). Top 10 Multi-Factor Authentication Software Solutions for 2021 - Spiceworks. <https://www.spiceworks.com/it-security/identity-access-management/articles/top-10-multi-factor-authentication-software-solutions/>
72. N. Hong, M. Kim, M.-S. Jun, J. Kang, A study on a jwt-based user authentication and api assessment scheme using imei in a smart home environment, *Sustainability* 9 (2017).
73. Nanda, A., Jeong, J. J., Shah, S. W. A., Nosouhi, M., & Doss, R. (2024). Examining usable security features and user perceptions of Physical Authentication Devices. *Computers and Security*, 139(December 2023), 103664. <https://doi.org/10.1016/j.cose.2023.103664>
74. Naqvi, B., Perova, K., Farooq, A., Makhdoom, I., Oyediji, S., & Porras, J. (2023). Mitigation strategies against the phishing attacks: A systematic literature review. *Computers and Security*, 132, 103387. <https://doi.org/10.1016/j.cose.2023.103387>
75. Ngurah, I. G., Derrick, D., & Satrio, N. (2023). Analisa Tindak Pidana Cyber Crime Pada Bidang Perbankan Nasional Berupa Pencurian Data Kartu Kredit ( Carding ). 1–12.
76. NIKHIL, R.; ANISHA, B. S.; KUMAR, Ramakanth. Users Sync Authentication using External Ldap in Organizations. In: 2020 IEEE International Conference for Innovation in Technology (INOCON). IEEE, 2020. p. 1-4.
77. NIST. (2023). NVD - Home. <https://nvd.nist.gov/>
78. O'Neill, M., Heidbrink, S., Ruoti, S., Whitehead, J., Bunker, D., Dickinson, L., Hendershot, T., Reynolds, J., Seamons, K., & Zappala, D. (2017). TrustBase: An architecture to repair and strengthen certificate-based authentication. *Proceedings of the 26th USENIX Security Symposium*.
79. Ogbanufe, O., & Kim, D. J. (2018). Comparing fingerprint-based biometrics authentication versus traditional authentication methods for e-payment. *Decision Support Systems*, 106, 1–14. <https://doi.org/10.1016/j.dss.2017.11.003>

- 80.Okta help center (lightning). *Okta Help Center (Lightning)*.  
URL: [https://support.okta.com/help/s/article/what-is-okta?language=en\\_US](https://support.okta.com/help/s/article/what-is-okta?language=en_US) (date of access: 10.12.2024).
- 81.Ometov, A., Bezzateev, S., Mäkitalo, N., Andreev, S., Mikkonen, T., & Koucheryavy, Y. (2018). Multi-factor authentication: A survey. *Cryptography*, 2(1), 1–31. <https://doi.org/10.3390/cryptography2010001>
- 82.Ometov, A., Bezzateev, S., Mäkitalo, N., Andreev, S., Mikkonen, T., & Koucheryavy, Y. (2018). Multi-factor authentication: A survey. *Cryptography*, 2(1). <https://doi.org/10.3390/cryptography2010001>
- 83.Oren, Y., & Arad, D. (2022). Toward Usable and Accessible Two-Factor Authentication Based on the Piezo-Gyro Channel. *IEEE Access*, 10. <https://doi.org/10.1109/ACCESS.2022.3150519>
- 84.Org, W. C., Sharmila, K., Janaki, V., & Nagaraju, A. (2017). A survey on user authentication techniques. *Pdfs.Semanticscholar.Org*, 10(2). <https://doi.org/10.13005/ojst/10.02.37>
- 85.Owens, K., Anise, O., Krauss, A., & Ur, B. (2021). User perceptions of the usability and security of smartphones as FIDO2 roaming authenticators. *Proceedings of the 17th Symposium on Usable Privacy and Security, SOUPS 2021*.
- 86.Owens, K., Ur, B., & Anise, O. (2020). A Framework for Evaluating the Usability and Security of Smartphones as FIDO2 Roaming Authenticators. *Who Are You?! Adventures in Authentication Workshop*.
- 87.Phan, K. (2018). Implementing Resiliency of Adaptive Multi-Factor Authentication Systems. 65, 1–96. [https://repository.stcloudstate.edu/msia\\_etdshttps://repository.stcloudstate.edu/msia\\_etds/65](https://repository.stcloudstate.edu/msia_etdshttps://repository.stcloudstate.edu/msia_etds/65)
- 88.Pubcookie:open-source software for intra-institutional web authentication. - 2007. - Режим доступа: <http://www.pubcookie.org/>. - Заголовок з екрану.
- 89.R. Hat, Keycloak—open source identity and access management, 2021.
- 90.Rajeswari, S. R., & Seenivasagam, V. (2016). Comparative Study on Various Authentication Protocols in Wireless Sensor Networks. In *Scientific World Journal* (Vol. 2016). <https://doi.org/10.1155/2016/6854303>
- 91.Ray, P. P. (2023). Web3: A comprehensive review on background, technologies, applications, zero-trust architectures, challenges and future directions. *Internet of Things and Cyber-Physical Systems*, 3(May), 213–248. <https://doi.org/10.1016/j.iotcps.2023.05.003>

92. Rui, Z., & Yan, Z. (2019). A Survey on Biometric Authentication: Toward Secure and Privacy-Preserving Identification. In IEEE Access (Vol. 7). <https://doi.org/10.1109/ACCESS.2018.2889996>
93. RUOTI, Scott; ROBERTS, Brent; SEAMONS, Kent. Authentication melee: A usability analysis of seven web authentication systems. In: Proceedings of the 24th international conference on world wide web. 2015. p. 916-926.
94. S. Anand, P. Jain, Nitin, and R. Rastogi, "Security Analysis and Implementation of 3-Level Security System Using Image-Based Authentication," in 2012 UKSim 14th International Conference on Computer Modelling and Simulation, Cambridge, UK, 2012, pp. 547-552, doi:10.1109/UKSim.2012.83. <https://ieeexplore.ieee.org/document/6205505>
95. S. Thorgersen, P. I. Silva, Keycloak-identity and access management for modern applications: harness the power of Keycloak, OpenID Connect, and OAuth 2.0 protocols to secure applications, Packt Publishing Ltd, 2021.
96. Sadhu, P. K., Yanambaka, V. P., & Abdelgawad, A. (2022). Internet of Things: Security and Solutions Survey. Sensors 2022, Vol. 22, Page 7433, 22(19), 7433. <https://doi.org/10.3390/S22197433>
97. Saevanee, H., Clarke, N. L., & Furnell, S. M. (2012). Multi-modal behavioural biometric authentication for mobile devices. IFIP Advances in Information and Communication Technology, 376 AICT, 465–474. [https://doi.org/10.1007/978-3-642-30436-1\\_38](https://doi.org/10.1007/978-3-642-30436-1_38)
98. Salameh, A., Elias, nur fazidah, & Karim, nader abdel. (2016). Proposed Model for Measuring Acceptance of Online Ads. Journal of Engineering and Applied Sciences. <http://docsdrive.com/pdfs/medwelljournals/jeasci/2016/2181-2185.pdf>
99. SARKAR, Arpita; SINGH, Binod K. A review on performance, security and various biometric template protection schemes for biometric authentication systems. Multimedia Tools and Applications, 2020, 79.37: 27721-27776.
100. Scott Cantor. Mace shibboleth arch conformance. University of Washington, NCSA, 2005.
101. Shah D., Bharadi V. IoT Based Biometrics Implementation on Raspberry Pi/
102. Shelgaonkar S.K. Creating a smart home environment with IOT driven home appliances. GRIN Verlag. 2016 p. 80 p.
103. Shibboleth Project — Internet2 Middleware. - 2007. - Режим доступа: <http://shibboleth.internet2.edu>. - Заголовок з екрану.
104. Silva, R. da. (2021). Calls for behavioural biometrics as bank fraud soars. Biometric Technology Today, 2021(9), 7–9. [https://doi.org/10.1016/S0969-4765\(21\)00095-3](https://doi.org/10.1016/S0969-4765(21)00095-3)

105. Sinigaglia, F., Carbone, R., Costa, G., & Zannone, N. (2020). A survey on multi-factor authentication for online banking in the wild. *Computers & Security*, 95, 101745. <https://doi.org/10.1016/J.COSE.2020.101745>
106. Smallman, M. (2020). Good call: the hybrid answer to voice authentication. *Biometric Technology Today*, 2020(4). [https://doi.org/10.1016/S0969-4765\(20\)30051-5](https://doi.org/10.1016/S0969-4765(20)30051-5)
107. Subbarao, D., Raju, B., Anjum, F., Rao, C. venkateswara, & Reddy, B. M. (2023). Microsoft Azure active directory for next level authentication to provide a seamless single sign-on experience. *Applied Nanoscience (Switzerland)*, 13(2). <https://doi.org/10.1007/s13204-021-02021-0>
108. The Swiss Education and Research Network, Authentication and Authorization Infrastructure. - 2007. - Режим доступу: <http://www.switch.ch/aai/>. - Заголовок з екрану.
109. The University Login: Authentication for Web Applications – Implementation Comparison University of Auckland, ITSS, 2004.
110. Луцків А.М. Математичне моделювання і обробка динамічно введеного підпису для задачі аутентифікації особи у інформаційних системах: автореф. дис... канд. техн. наук: 01.05.02. Терноп. держ. техн. ун-т ім. І. Пулюя. Т., 2008. 20 с.

## Додаток А. Playbook для розгортання системи аутентифікації

```
---
- name: Deploy Authentication System
  hosts: all
  become: true
  vars:
    ldap_domain: "luguniv.edu.ua"
    ldap_admin_password: "securepassword"
    keycloak_admin_username: "admin"
    keycloak_admin_password: "securepassword"
  tasks:

    - name: Install required packages
      apt:
        name:
          - openldap-server
          - openldap-utils
          - docker.io
          - docker-compose
        state: present
        update_cache: yes

    - name: Configure OpenLDAP server
      template:
        src: ldap/slapd.conf.j2
        dest: /etc/ldap/slapd.conf
        notify: restart slapd

    - name: Initialize LDAP database
      command: ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/core.ldif
      args:
        creates: /var/lib/ldap/DB_CONFIG

    - name: Set LDAP root password
      shell: |
        slappasswd -s {{ ldap_admin_password }} > /tmp/ldap_passwd
        ldapmodify -Y EXTERNAL -H ldapi:/// -f /tmp/ldap_passwd

    - name: Deploy Keycloak container
      community.docker.docker_compose:
        project_name: keycloak
        definition:
          version: "3.8"
          services:
            keycloak:
              image: quay.io/keycloak/keycloak:latest
              ports:
                - "8080:8080"
              environment:
                - KEYCLOAK_USER={{ keycloak_admin_username }}
```

```

    - KEYCLOAK_PASSWORD={{ keycloak_admin_password }}
    volumes:
    - /opt/keycloak/data:/opt/jboss/keycloak/standalone/data
    restart: always

- name: Configure Keycloak realm and clients
  uri:
    url: "http://localhost:8080/auth/admin/realms"
    method: POST
    user: "{{ keycloak_admin_username }}"
    password: "{{ keycloak_admin_password }}"
    body: |
      {
        "realm": "university",
        "enabled": true
      }
    body_format: json
    status_code: 201
    when: ansible_facts['distribution'] == "Ubuntu"

- name: Deploy Next.js middleware
  shell: |
    git clone https://github.com/example/middleware-nextjs.git /opt/middleware
    cd /opt/middleware && npm install && npm run build

- name: Configure middleware environment variables
  template:
    src: nextjs/.env.local.j2
    dest: /opt/middleware/.env.local

- name: Start middleware service
  shell: |
    pm2 start /opt/middleware/.next/standalone/server.js --name middleware
    pm2 save

handlers:
- name: restart slapd
  service:
    name: slapd
    state: restarted

```