**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформаційних систем та технологій**

|  |  |
| --- | --- |
|  |  |

**Курсова робота**

**з дисципліни «Програмування»**

**на тему: «Сайт знайомств»**

Виконав:

студент 1 курсу, групи ІА-33

Кондратюк Віталій Володимирович \_\_\_\_\_\_\_\_\_\_

(підпис)

Керівник:

асистент кафедри ІСТ

Мягкий Михайло Юрійович \_\_\_\_\_\_\_\_\_\_

(підпис)

Засвідчую, що у цій курсовій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент. \_\_\_\_\_\_\_\_\_\_

(підпис)

Київ – 2024 року

ЗМІСТ

[ВСТУП 3](#_Toc158383261)

[1 ВИМОГИ ДО СИСТЕМИ 4](#_Toc158383262)

[1.1 Функціональні вимоги до системи 4](#_Toc158383263)

[1.2 Нефункціональні вимоги до системи 4](#_Toc158383264)

[2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ 5](#_Toc158383265)

[2.1 Діаграма прецедентів 6](#_Toc158383266)

[2.2 Опис сценаріїв використання системи 7](#_Toc158383267)

[3 АРХІТЕКТУРА СИСТЕМИ 24](#_Toc158383268)

[4 РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ 26](#_Toc158383269)

[4.1 Загальна структура проекту 27](#_Toc158383270)

[4.2 Компоненти рівня доступу до даних 27](#_Toc158383271)

[4.3 Компоненти рівня бізнес-логіки 29](#_Toc158383272)

[4.4 Компоненти рівня інтерфейсу користувача 30](#_Toc158383273)

[ВИСНОВКИ 31](#_Toc158383274)

[ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 33](#_Toc158383275)

[ДОДАТОК А Лістінг програми 36](#_Toc158383276)

[ДОДАТОК Б Назва додатку Б (за наявності) 37](#_Toc158383277)

[ДОДАТОК В Назва додатку В (за наявності) 38](#_Toc158383278)

# ВСТУП

У сучасному світі цифрових технологій і глобальної мережі Інтернет онлайн-знайомства стали важливою частиною соціального життя багатьох людей. З розвитком технологій з'явилися можливості для створення платформ, які допомагають людям знаходити нових друзів, партнерів для спілкування чи навіть кохання. Сайти знайомств набули великої популярності завдяки своїй зручності, доступності та широкому функціоналу.

Актуальність створення системи сайту знайомств зумовлена високим попитом на такі послуги та постійно зростаючим числом користувачів, які прагнуть знайти нові знайомства в Інтернеті. Використання сучасних технологій для пошуку партнерів дозволяє не тільки розширити коло спілкування, але й зробити процес знайомства більш зручним і безпечним.

Метою створення системи сайту знайомств є надання користувачам зручного та інтуїтивно зрозумілого інструменту для пошуку нових знайомств. Система повинна забезпечувати можливість редагування власної анкети, перегляду анкет інших користувачів, пошуку анкет за ключовими словами, відправки запрошень на знайомство та перегляду закритої інформації інших користувачів при отриманні підтвердження запрошення.

Для досягнення цієї мети система повинна вирішувати наступні задачі:

1. Реалізація зручного інтерфейсу для реєстрації та авторизації користувачів.

2. Забезпечення можливості редагування та зберігання анкет користувачів.

3. Реалізація функції пошуку анкет за ключовими словами.

4. Надання можливості користувачам відправляти запрошення на знайомство.

5. Забезпечення механізму перегляду закритої інформації інших користувачів при отриманні запрошення чи підтвердження власного запрошення.

6. Гарантування безпеки та конфіденційності даних користувачів.

Таким чином, розробка системи сайту знайомств має на меті створення ефективного інструменту для пошуку та налагодження нових соціальних зв’язків, що відповідає потребам сучасного користувача в умовах стрімкого розвитку інформаційних технологій.

# 1 ВИМОГИ ДО СИСТЕМИ

Розробка системи сайту знайомств вимагає чіткого визначення вимог, які висуває умовний "замовник". Вимоги поділяються на функціональні та нефункціональні.

***- Функціональні вимоги***

1. **Реєстрація та авторизація користувачів**

- Користувачі повинні мати можливість зареєструватися на сайті, заповнивши форму з особистою інформацією.

- Система повинна забезпечувати можливість авторизації зареєстрованих користувачів.

2. **Редагування профілю**

- Користувачі повинні мати можливість створювати та редагувати власні анкети, включаючи завантаження фотографій, додавання опису та вказівки інтересів.

3. **Перегляд анкет**

- Користувачі повинні мати можливість переглядати анкети інших користувачів, включаючи базову відкриту інформацію.

4. **Пошук анкет**

- Система повинна забезпечувати функціональність пошуку анкет за ключовими словами, щоб користувачі могли знаходити потенційних партнерів за спільними інтересами.

5. **Відправка запрошень**

- Користувачі повинні мати можливість відправляти запрошення на знайомство іншим користувачам.

6. **Перегляд закритої інформації**

- Користувачі повинні мати можливість переглядати закриту інформацію інших користувачів лише після отримання запрошення чи підтвердження власного запрошення.

***- Нефункціональні вимоги***

1. **Продуктивність**

- Система повинна забезпечувати швидкий час відгуку на запити користувачів, включаючи пошук анкет та перегляд профілів.

2. **Надійність**

- Система повинна бути надійною, забезпечуючи безперебійну роботу та збереження даних користувачів.

3. **Безпека**

- Система повинна забезпечувати високий рівень безпеки, включаючи захист особистих даних користувачів та запобігання несанкціонованому доступу.

4. **Масштабованість**

- Система повинна бути масштабованою, щоб підтримувати зростаючу кількість користувачів без втрати продуктивності.

5. **Юзабіліті**

- Інтерфейс системи повинен бути інтуїтивно зрозумілим і зручним у використанні, щоб забезпечити позитивний досвід користувачів.

6. **Сумісність**

- Система повинна бути сумісною з основними веб-браузерами та підтримувати адаптивний дизайн для коректного відображення на різних пристроях (ПК, планшети, смартфони).

7. **Конфіденційність**

- Система повинна гарантувати конфіденційність особистої інформації користувачів, зокрема закритої інформації, яка доступна лише за згодою користувача.

Виконання цих вимог є критичним для успішної розробки та функціонування системи сайту знайомств. Вони забезпечують не тільки відповідність потребам користувачів, але й високу якість сервісу, що надається системою.

# 2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ

Изображение выглядит как диаграмма, линия, круг, текст

Автоматически созданное описание

Сценарій 1: Редагування власної анкети

|  |  |
| --- | --- |
| Назва сценарію | Редагування власної анкети |
| Актор | Користувач |
| Передумови | Користувач має обліковий запис та авторизований |
| Основний потік | 1. Користувач відкриває свій профіль |
|  | 2. Користувач натискає кнопку "Редагувати анкету" |
|  | 3. Система відображає форму для редагування анкети |
|  | 4. Користувач вносить зміни до анкети |
|  | 5. Користувач натискає кнопку "Зберегти зміни" |
|  | 6. Система зберігає зміни та оновлює анкету |
| Альтернативні потоки | 3a. Якщо система не може відобразити форму редагування через помилку, вона показує повідомлення про помилку |
|  | 5a. Якщо користувач не заповнив обов'язкові поля, система показує повідомлення про помилку і просить заповнити їх |
| Результат | Анкета користувача оновлена та збережена |

Сценарій 2: Перегляд анкет

|  |  |
| --- | --- |
| Назва сценарію | Перегляд анкет |
| Актор | Користувач |
| Передумови | Користувач має обліковий запис та авторизований |
| Основний потік | 1. Користувач відкриває сторінку перегляду анкет |
|  | 2. Система відображає список анкет інших користувачів |
|  | 3. Користувач обирає анкету для перегляду |
|  | |  | | --- | |  |  |  | | --- | | 4. Система відображає деталі обраної анкети | |
| Альтернативні потоки | 2a. Якщо система не може відобразити список анкет через помилку, вона показує повідомлення про помилку |
| Результат | Користувач бачить деталі анкети обраного користувача |

Сценарій 3: Пошук анкет за ключовими словами

|  |  |
| --- | --- |
| Назва сценарію | Пошук анкет за ключовими словами |
| Актор | Користувач |
| Передумови | Користувач має обліковий запис та авторизований |
| Основний потік | 1. Користувач відкриває сторінку пошуку анкет |
|  | 2. Користувач вводить ключові слова у поле пошуку |
|  | 3. Користувач натискає кнопку "Пошук" |
|  | 4. Система виконує пошук анкет за введеними ключовими словами |
|  | 5. Система відображає результати пошуку у вигляді списку анкет |
| Альтернативні потоки | 4a. Якщо система не знаходить жодної анкети за введеними ключовими словами, вона показує повідомлення про відсутність результатів |
| Результат | Користувач бачить список анкет, які відповідають введеним ключовим словам |

Сценарій 4: Відправка запрошення

|  |  |
| --- | --- |
| Назва сценарію | Відправка запрошення |
| Актор | Користувач |
| Передумови | Користувач має обліковий запис та авторизований |
| Основний потік | 1. Користувач переглядає анкету іншого користувача |
|  | 2. Користувач натискає кнопку "Відправити запрошення" |
|  | 3. Система надсилає запрошення обраному користувачу |
|  | 4. Обраний користувач отримує повідомлення про нове запрошення |
| Альтернативні потоки | 3a. Якщо система не може відправити запрошення через помилку, вона показує повідомлення про помилку |
| Результат | Запрошення успішно відправлено обраному користувачу |

Сценарій 5: Перегляд закритої інформації іншого користувача

|  |  |
| --- | --- |
| Назва сценарію | Перегляд закритої інформації іншого користувача |
| Актор | Користувач |
| Передумови | Користувач має обліковий запис, авторизований і отримав запрошення |
| Основний потік | 1. Користувач отримує запрошення від іншого користувача |
|  | 2. Користувач приймає запрошення |
|  | 3. Система надає доступ до закритої інформації іншого користувача |
|  | 4. Користувач переглядає закриту інформацію іншого користувача |
| Альтернативні потоки | 3a. Якщо система не може надати доступ до закритої інформації через помилку, вона показує повідомлення про помилку |
| Результат | Користувач отримує доступ до закритої інформації іншого користувача і може її переглядати |

# 3 АРХІТЕКТУРА СИСТЕМИ

Система сайту знайомств складається з декількох компонентів, які взаємодіють між собою для забезпечення функціональності, зручності та безпеки користувачів. Нижче описані основні компоненти системи, їх зв'язки та технології, що використовуються.

**Компоненти системи**

1. **Клієнтська частина (Frontend)**
   * **Мова розмітки:** HTML5
   * **Стилізація:** CSS3 (Bootstrap для адаптивності та швидкої розробки)
   * **Анімації:** CSS-анімації
   * **Взаємодія з користувачем:** JavaScript (ES6+)
2. **Серверна частина (Backend)**
   * **Мова програмування:** Node.js
   * **Фреймворк:** Express.js для маршрутизації та обробки запитів
   * **Шаблонізатор:** EJS для динамічного рендерингу сторінок
   * **API:** REST API для взаємодії між клієнтом та сервером
3. **База даних**
   * **Тип:** Реляційна база даних
   * **СУБД:** PostgreSQL для зберігання інформації про користувачів, анкети, запрошення тощо
4. **Система аутентифікації та авторизації**
   * **Технологія:** JWT (JSON Web Tokens) для забезпечення безпеки користувачів
   * **Бібліотеки:** Passport.js для управління аутентифікацією
5. **Зберігання файлів**
   * **Об’єкти:** Фотографії користувачів
   * **Сервіс:** AWS S3 або інший хмарний сервіс для зберігання файлів

**Взаємозв'язок компонентів**

**1. Клієнтська частина**

Клієнтська частина відповідає за відображення інтерфейсу користувача та взаємодію з сервером через API-запити. Основні функції клієнтської частини:

* Реєстрація та авторизація користувачів
* Редагування профілю
* Пошук анкет
* Відправка та отримання запрошень
* Завантаження фотографій

**2. Серверна частина**

Серверна частина обробляє запити від клієнта, взаємодіє з базою даних та забезпечує безпеку. Основні функції серверної частини:

* Обробка HTTP-запитів
* Взаємодія з базою даних
* Генерація та перевірка JWT
* Маршрутизація запитів

**3. База даних**

База даних зберігає інформацію про користувачів, анкети, ключові слова, запрошення та закриту інформацію. Основні функції бази даних:

* Зберігання та обробка даних
* Виконання SQL-запитів

**4. Система аутентифікації та авторизації**

Система аутентифікації забезпечує безпечний доступ до облікових записів користувачів. Основні функції системи:

* Генерація та перевірка токенів доступу
* Управління сесіями користувачів

**5. Зберігання файлів**

Зберігання файлів відповідає за безпечне зберігання та доступ до фотографій користувачів. Основні функції системи:

* Завантаження та збереження фотографій
* Надання доступу до фотографій за запитом

**Рисунок 3.1.**

Изображение выглядит как текст, снимок экрана, линия, Шрифт

Автоматически созданное описание

У цій архітектурі клієнтська частина взаємодіє із серверною через REST API. Серверна частина обробляє ці запити, звертається до бази даних для збереження та отримання необхідної інформації, а також керує аутентифікацією користувачів. Для зберігання файлів використовується зовнішній хмарний сервіс, що забезпечує зручний доступ та масштабованість.

Ця архітектура дозволяє легко підтримувати та розширювати систему, додаючи нові функції та покращуючи існуючі, а також забезпечує високу безпеку та продуктивність.

# 4 РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

**4.1 Загальна структура проекту**

Проект складається з веб-ресурсів, бібліотек та вихідного коду, який в свою чергу можна поділити на компоненти рівня доступу до даних, компоненти бізнес-логіки та веб-компоненти. Загальна структура проекту представлена на рисунку 4.1.

**4.2 Компоненти рівня доступу до даних**

Основні сутності та інтерфейси рівня доступу до даних наведені на рисунку 4.2.

**4.3 Веб-компоненти**

**Реєстрація користувача (register.html)**

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Авторизація користувача (login.html)

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Головна сторінка (index.html)

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

4.4 Стилі (styles.css)

Изображение выглядит как снимок экрана, текст

Автоматически созданное описание

4.5 JavaScript (script.js)

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Графическое программное обеспечение

Автоматически созданное описание

У цій курсовій роботі було розроблено сайт знайомств, який забезпечує користувачам зручний та інтуїтивно зрозумілий інтерфейс для взаємодії. Було використано сучасні веб-технології для реалізації функціоналу та забезпечення привабливого вигляду сайту. У майбутньому можливе розширення функціоналу та інтеграція з базою даних для збереження та управління даними користувачів.

# ВИСНОВКИ

Під час написання курсової роботи було створено сайт знайомств, який дозволяє користувачам реєструватися, входити в систему, редагувати власні анкети, переглядати анкети інших користувачів, здійснювати пошук за ключовими словами, відправляти запрошення та переглядати закриту інформацію інших користувачів після підтвердження запрошень. Розробка сайту включала використання сучасних веб-технологій та фреймворків, що забезпечують зручність та ефективність взаємодії користувачів із системою.

Основними кроками в розробці системи були:

1. Формулювання вимог: На початку роботи було визначено функціональні та нефункціональні вимоги до системи, що дозволило чітко окреслити очікувану поведінку сайту.

2. Вибір технологій: Для реалізації сайту було обрано HTML, CSS (з використанням Bootstrap для стилізації та адаптивності), та JavaScript для динамічної взаємодії на стороні клієнта.

3. Розробка основних сторінок: Було створено декілька HTML-сторінок, зокрема головну сторінку, сторінку реєстрації, сторінку авторизації, сторінку редагування анкети, сторінку перегляду анкет, сторінку пошуку та сторінку відправки запрошень.

4. Додавання стилізації та анімацій: Для покращення візуального вигляду сайту було додано стилі в рожевій кольоровій схемі та анімації CSS для плавного переходу елементів на сторінці.

5. Розробка функціоналу форм: Було додано валідацію форм на стороні клієнта для забезпечення коректного введення даних користувачами.

Загальна оцінка результату:

Сайт знайомств було успішно створено відповідно до визначених вимог. Основні цілі проекту були досягнуті, а саме: створено зручний та функціональний інтерфейс для взаємодії користувачів, забезпечено можливість реєстрації та авторизації, а також реалізовано пошук та перегляд анкет. Додатково було додано візуальні покращення та анімації, що підвищують привабливість сайту для користувачів.

Пропозиції щодо використання та подальшого розвитку:

1. Інтеграція з базою даних: Наступним кроком може бути інтеграція сайту з базою даних для збереження інформації користувачів та анкет, що забезпечить більш ефективне управління даними.

2. Розширення функціоналу: Можна додати додаткові функції, такі як система повідомлень між користувачами, можливість завантаження та перегляду фотографій, а також рейтингова система для анкет.

3. Безпека: Впровадження механізмів для забезпечення безпеки даних користувачів, таких як шифрування паролів та захист від SQL-ін'єкцій.

4. Оптимізація продуктивності: Проведення оптимізації продуктивності сайту для забезпечення швидкого завантаження сторінок та зменшення часу відгуку.

Отже, сумуючи наведене вище, можна сказати, що створена система є гнучкою, зручною у використанні та має великий потенціал для подальшого розвитку та вдосконалення. Завдяки відкритій архітектурі, розширення функціоналу не є проблемою, що дозволяє легко адаптувати сайт до нових вимог та побажань користувачів.

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. W3Schools. HTML Tutorial. <https://www.w3schools.com/html/>

- Використано для ознайомлення з основами HTML та створення структурних елементів веб-сторінок.

2. W3Schools. CSS Tutorial. [https://www.w3schools.com/css/](https://www.w3schools.com/css/%5d(https://www.w3schools.com/css/)

- Використано для стилізації веб-сторінок та додавання анімаційних ефектів.

3. Bootstrap. Introduction.

<https://getbootstrap.com/docs/4.5/getting-started/introduction/>

- Використано для адаптивної верстки та покращення візуального вигляду сайту.

4. Mozilla Developer Network (MDN). JavaScript Guide. <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>

- Використано для додавання інтерактивності та валідації форм на стороні клієнта.

5. ObjectAid. UML Explorer. [https://www.objectaid.com/](https://www.objectaid.com/%5d(https://www.objectaid.com/)

- Використано для генерації UML-діаграм на основі написаного коду, що спрощує розробку та документування проекту.

6. Oracle. Java Documentation. <https://docs.oracle.com/en/java/>

- Використано для написання класів та об'єктів Java, а також для інтеграції з базами даних та створення бізнес-логіки.

7. Hibernate ORM [https://hibernate.org/](https://hibernate.org/](https://hibernate.org/)

- Використано для об'єктно-реляційного відображення (ORM) та управління базами даних.

8. Spring Framework. Spring Data.

<https://spring.io/projects/spring-data>

- Використано для побудови запитів до баз даних та налаштування з'єднання між системою та базою даних.

9. Spring Framework. Spring Security. <https://spring.io/projects/spring-security>

- Використано для налаштування безпеки системи, включаючи авторизацію та автентифікацію користувачів.

10. Javadoc. Java SE Documentation. <https://docs.oracle.com/javase/8/docs/api/>

- Використано для створення документації по написаному коду Java та генерації API-документації.

11. IntelliJ IDEA Documentation. <https://www.jetbrains.com/idea/documentation/>

- Використано як середовище розробки для написання та тестування коду, а також для інтеграції з різними плагінами, такими як ObjectAid.

12. MySQL Documentation. <https://dev.mysql.com/doc/>

- Використано для налаштування та управління базами даних MySQL, які використовуються у проекті.

13. GitHub. <https://github.com/https://github.com/>

- Використано для управління версіями коду та співпраці над проектом.

14. Stack Overflow. <https://stackoverflow.com/>

- Використано для вирішення різних технічних питань та пошуку прикладів коду.

15. GeeksforGeeks. Java Tutorial. <https://www.geeksforgeeks.org/java/>

- Використано для додаткових прикладів та пояснень з програмування на Java.

16. CSS-Tricks. CSS Reference. <https://css-tricks.com/almanac/>

- Використано для поглибленого вивчення CSS та його властивостей.

17. Java Design Patterns. <https://java-design-patterns.com/>

- Використано для ознайомлення з шаблонами проектування на Java.

18. FreeCodeCamp. Responsive Web Design. <https://www.freecodecamp.org/learn/responsive-web-design/>

- Використано для вивчення адаптивного веб-дизайну.

19. DigitalOcean. How to Install and Use MySQL. <https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-20-04>

- Використано для налаштування та управління MySQL на сервері.

20. Spring Guides. <https://spring.io/guides>

- Використано для кращого розуміння та реалізації фреймворку Spring у проекті.

21. Baeldung. A Guide to JPA with Spring.

<https://www.baeldung.com/the-persistence-layer-with-spring-and-jpa>

- Використано для інтеграції JPA (Java Persistence API) зі Spring.

22. Java Code Geeks. <https://www.javacodegeeks.com/>

- Використано для додаткових прикладів та пояснень з програмування на Java.

23. Oracle. MySQL 8.0 Reference Manual. <https://dev.mysql.com/doc/refman/8.0/en/>

- Використано для детальної документації та налаштування MySQL.

24. JetBrains. IntelliJ IDEA. <https://www.jetbrains.com/idea/>

- Використано як середовище розробки для написання та тестування коду.