

**БАКАЛАВРСЬКА РОБОТА**

**БР. ІІІ - 44.00.00.000 ІІЗ**

**Група ІІІ-21-3**

**Гуцул Роман**

**2025**

**Івано-Франківський національний технічний університет нафти і газу**

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

**Гуцул Роман Русланович**

(прізвище, ім'я, по батькові)

УДК 004  
(індекс)

## **БАКАЛАВРСЬКА РОБОТА**

**Реалізація ресурсу соціального нетворкінгу для взаємодії ІТ-**

**розробників**

(назва роботи)

**Інженерія програмного забезпечення**

(назва освітньої програми)

**121 - Інженерія програмного забезпечення**

(шифр і назва спеціальності)

**Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело**

Здобувач освітнього рівня Гуцул Р.Р.  
(підпис, ініціали та прізвище здобувача)

Науковий керівник Ваврик Тетяна Олександрівна , асистент  
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту  
Завідувач кафедри

доц. Бандура В.В.  
(посада) (підпис) (дата) (ініціали та прізвище)

**Івано-Франківськ – 2025**

**Івано-Франківський національний технічний університет нафти і газу**

Інститут, факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Спеціальність 121 – Інженерія програмного забезпечення

**ЗАТВЕРДЖУЮ:**

Зав. кафедрою ІІЗ

доц.

В.В. Бандура

“     ”     2025 р.

## **ЗАВДАННЯ**

### **НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ**

**Гуцулу Роману Руслановичу**

(прізвище, ім'я, по-батькові)

**1. Тема проекту (роботи) “Реалізація ресурсу соціального нетворкінгу для взаємодії ІТ-розробників”**

керівник проекту (роботи) Ваврик Т.О., асистент

затвержені наказом закладу вищої освіти від “ 28 ” квітня 2025 р. № 264/7

**2. Строк подання студентом проекту (роботи) 10 червня 2025 р.**

**3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики**

**4. Зміст розрахунково - пояснювальної записки (перелік питань, які потрібно розробити)**

1. Аналіз предметної області розробки ресурсів соціального нетворкінгу

2. Моделювання структури платформи соціального нетворкінгу для взаємодії ІТ-розробників

3. Реалізація діаграм варіантів використання системи нетворкінгу

4. Розробка діаграми послідовності

5. Програмна розробка ресурсу соціального нетворкінгу для взаємодії ІТ-розробників

**5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)**

1. Огляд загальної процедури роботи Codementor (рис. 1.1)

2. Вигляд платформи Codementor (рис. 1.2)

3. Інтерфейс платформи Chegg (рис. 1.3)

4. Онлайн-платформа MentorCruise (рис. 1.4)

5. Вигляд веб-ресурсу Stack Overflow (рис. 1.5)

## 6. Консультанти розділів проекту (роботи)

| Розділ | Консультант | Підпис, дата   |                  |
|--------|-------------|----------------|------------------|
|        |             | Завдання видав | Завдання прийняв |
|        |             |                |                  |
|        |             |                |                  |
|        |             |                |                  |

7. Дата видачі завдання 28 квітня 2025 р.

Керівник \_\_\_\_\_

(підпис)

Завдання прийняв до виконання \_\_\_\_\_

(підпис)

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проекту (роботи)   | Строк виконання етапів проекту | Примітка |
|-------|--|--------------------------------|----------|
| 1     | Аналіз предметної області розробки ресурсів соціального нетворкінгу                  | 04.05.2025                     | виконано |
| 2     | Моделювання структури платформи соціального нетворкінгу для взаємодії ІТ-розробників | 15.05.2025                     | виконано |
| 3     | Реалізація діаграм варіантів використання системи нетворкінгу                        | 21.05.2025                     | виконано |
| 4     | Розробка діаграми послідовності  | 28.05.2025                     | виконано |
| 5     | Програмна розробка ресурсу соціального нетворкінгу для взаємодії ІТ-розробників      | 03.06.2025                     | виконано |
| 6     | Оформлення пояснювальної записки дипломної роботи завідувачем кафедри                | 10.06.2025                     | виконано |

Студент – дипломник \_\_\_\_\_

(підпис)

Керівник роботи \_\_\_\_\_

(підпис)

## АНОТАЦІЯ

Бакалаврська робота містить 75 сторінок, 23 рисунки, список використаних джерел із 33 найменуваннями, 1 додаток.

**Мета роботи** - розробка та реалізація ресурсу соціального нетворкінгу, призначеного для професійної взаємодії ІТ-розробників, що включає архітектуру системи, інтерфейси, механізми автентифікації, управління контентом і можливості для розширення функціоналу.

**Об'єкт дослідження** - процес професійної взаємодії між ІТ-розробниками в цифровому середовищі.

**Предмет дослідження** - архітектурні, функціональні та програмні рішення, що забезпечують розробку ресурсу соціального нетворкінгу для ІТ-розробників.

**В першому розділі** проведено аналіз сучасного стану соціальних платформ для ІТ-сфери та обґрунтовано доцільність створення нової системи

**В другому розділі** розроблено архітектуру, визначено вимоги до системи та виконано моделювання її структури

**В третьому розділі** реалізовано функціональний прототип ресурсу та проведено тестування його роботи.

**Висновок:** результати роботи можуть бути використані при розробці та впровадженні спеціалізованих соціальних платформ у ІТ-компаніях, стартап-інкубаторах, навчальних ІТ-програмах, а також для створення професійних спільнот ІТ-розробників у межах open-source та комерційних ініціатив

**КЛЮЧОВІ СЛОВА:** СОЦІАЛЬНИЙ НЕТВОРКІНГ, ІТ-РОЗРОБНИКИ, ПЛАТФОРМА, ПРОФЕСІЙНА ВЗАЄМОДІЯ, ВЕБ-РОЗРОБКА, АРХІТЕКТУРА СИСТЕМИ, МЕНТОРСТВО, FRONTEND, BACKEND, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

## ANNOTATION

The Bachelor's thesis comprises 75 pages, 23 figures, a list of 33 references, and 1 appendix.

**The purpose of this work** is the development and implementation of a social networking resource designed for professional interaction among IT developers, encompassing system architecture, interfaces, authentication mechanisms, content management, and possibilities for functional expansion.

**Object of Research:** The object of research is the process of professional interaction among IT developers in a digital environment.

**Subject of Research:** The subject of research encompasses the architectural, functional, and software solutions that ensure the development of a social networking resource for IT developers.

**In the first chapter**, an analysis of the current state of social platforms for the IT sector was conducted, and the expediency of creating a new system was substantiated.

**In the second chapter**, the architecture was developed, system requirements were defined, and its structure was modeled.

**In the third chapter**, a functional prototype of the resource was implemented, and its operation was tested.

**Conclusion:** The results of this work can be utilized in the development and implementation of specialized social platforms within IT companies, startup incubators, educational IT programs, as well as for creating professional communities of IT developers within open-source and commercial initiatives.

**KEYWORDS:** SOCIAL NETWORKING, IT DEVELOPERS, PLATFORM, PROFESSIONAL INTERACTION, WEB DEVELOPMENT, SYSTEM ARCHITECTURE, MENTORSHIP, FRONTEND, BACKEND, SOFTWARE.

## ЗМІСТ

|  |    |
|--|----|
| ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ .....  | 9  |
| ВСТУП .....  | 10 |
| <br>   |    |
| РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗРОБКИ РЕСУРСІВ<br>СОЦІАЛЬНОГО НЕТВОРКІНГУ .....                 | 14 |
| 1.1. Передумови розробки проекту соціального нетворкінгу для ІТ-<br>розробників.....                   | 14 |
| 1.1.1. Архітектура та функціонал .....   | 14 |
| 1.1.2. Аналіз існуючих рішень та обґрунтування необхідності .....                                      | 15 |
| 1.2. Актуальність розробки ресурсу соціального нетворкінгу для взаємодії<br>ІТ-розробників.....        | 15 |
| 1.3. Аналіз існуючих платформ обміну досвідом в області ІТ розробки ....                               | 17 |
| 1.3.1.Онлайн-сервіс Codementor .....   | 17 |
| 1.3.2. Платформа Chegg .....   | 21 |
| 1.3.3. Онлайн-платформа MentorCruise.....  | 24 |
| 1.3.4. Веб-ресурс Stack Overflow.....  | 27 |
| 1.3.5. Онлайн платформа Amazing Design People List .....   | 30 |
| <br>   |    |
| РОЗДІЛ 2. МОДЕЛЮВАННЯ СТРУКТУРИ ПЛАТФОРМИ СОЦІАЛЬНОГО<br>НЕТВОРКІНГУ ДЛЯ ВЗАЄМОДІЇ ІТ-РОЗРОБНИКІВ..... | 34 |
| 2.1. Запропонована архітектура ресурсу соціального нетворкінгу для ІТ-<br>розробників.....             | 34 |
| 2.1.1. Інтерфейс користувача та функціонал автентифікації .....  | 34 |
| 2.1.2. Взаємодія у спільноті (розділ публікацій).....  | 35 |
| 2.2. Специфікація вимог до системи соціального нетворкінгу .....                                       | 35 |

|           |      |              |        |      |  |                 |      |         |
|-----------|------|--------------|--------|------|--|-----------------|------|---------|
|           |      |              |        |      | БР.ІП – 44.00.00.000 ПЗ  |                 |      |         |
| Змн.      | Арк. | № докум.     | Підпис | Дата |  |                 |      |         |
| Розроб.   |      | Гуцул Р.Р.   |        |      | Реалізація ресурсу соціального нетворкінгу для взаємодії ІТ-розробників<br><b>Пояснювальна записка</b> | Літ.            | Арк. | Акрушіє |
| Перевір.  |      | Ваврик Т.О.  |        |      |  |                 | 6    |         |
| Реценз.   |      |              |        |      |  | ІФНТУНГ ІІ-21-3 |      |         |
| Н. Контр. |      | Піх М.М.     |        |      |  |                 |      |         |
| Затверд.  |      | Бандура В.В. |        |      |  |                 |      |         |

|  |    |
|--|----|
| 2.2.1. Апаратні та програмні вимоги до серверної інфраструктури .....    | 36 |
| 2.2.2. Програмні вимоги до розробки та функціонування .....              | 36 |
| 2.2.3. Вимоги до клієнтського середовища .....                           | 37 |
| 2.3. Використані програмні інструменти та технології .....               | 37 |
| 2.3.1. Технології реалізації фронтенду .....                             | 37 |
| 2.3.2. Технології реалізації бекенду .....                               | 39 |
| 2.3.3. Управління даними .....   | 41 |
| 2.3.4. Система аутентифікації .....                                      | 43 |
| 2.4. Реалізація діаграм варіантів використання системи нетворкінгу ..... | 44 |
| 2.5. Розробка діаграм потоку даних .....                                 | 46 |
| 2.5.1. Діаграма потоку даних для входу та реєстрації .....               | 46 |
| 2.5.2. Потік даних для розділу публікацій .....                          | 48 |
| 2.6. Розробка діаграми послідовності .....                               | 51 |

### РОЗДІЛ 3. ПРОГРАМНА РОЗРОБКА РЕСУРСУ СОЦІАЛЬНОГО

|   |    |
|---|----|
| НЕТВОРКІНГУ ДЛЯ ВЗАЄМОДІЇ ІТ-РОЗРОБНИКІВ.....                             | 55 |
| 3.1. Загальний вигляд та навігаційні елементи головної сторінки системи   | 55 |
| 3.2. Функціональність перегляду профілів розробників .....                | 56 |
| 3.2.1. Доступ до опцій розробників .....                                  | 56 |
| 3.3.1. Екран входу до системи.....  | 58 |
| 3.3.2. Екран реєстрації нових користувачів .....                          | 59 |
| 3.4. Деталізація профілю: розробка форм додавання досвіду та освіти ..... | 60 |
| 3.5. Функціональність панелі управління та розділу публікацій.....        | 63 |
| 3.5.1. Панель управління профілем користувача .....                       | 63 |
| 3.5.2. Розділ публікацій (стрічка публікацій) .....                       | 64 |
| 3.6. Тестування системи соціального нетворкінгу для ІТ-розробників.....   | 66 |
| 3.6.1. Документація з тестування програмного забезпечення .....           | 66 |
| 3.6.2. Модульне тестування (Unit Testing).....                            | 67 |

|  |    |
|--|----|
| 3.6.3. Тестування на прийнятність користувачами (User Acceptance Testing, UAT) ..... | 67 |
| 3.6.4. Результати тестування .....   | 67 |
| 3.7. Напрямки майбутнього розвитку проекту нетворкінгу для ІТ-розробників.....       | 68 |
| ВИСНОВКИ.....  | 70 |
| ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....  | 72 |
| ДОДАТКИ  |    |
| БІБЛІОГРАФІЧНА ДОВІДКА   |    |

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 8    |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

SPA – Single Page Application – Односторінковий додаток

Q&A – Questions & Answers – Запитання та відповіді

DBaaS – Database-as-a-Service – База даних як сервіс

UAT – User Acceptance Testing – Тестування на прийнятність користувачами

REST – Representational State Transfer – Передача репрезентативного стану

API – Application Programming Interface – Інтерфейс програмування додатків

DFD – Data Flow Diagram – Діаграма потоків даних

JSON – JavaScript Object Notation – Об'єктна нотація JavaScript

ODM – Object Data Modeling – Об'єктне моделювання даних

XSRF – Cross-Site Request Forgery – Міжсайтова підробка запитів

PaaS – Platform as a Service – Платформа як сервіс

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 9    |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

## ВСТУП

У сучасному інформаційному суспільстві зростає попит на ефективні засоби комунікації між фахівцями в різних професійних галузях, зокрема в галузі інформаційних технологій (ІТ). Розвиток цифрових платформ значно спростив процеси спілкування, проте професійне мережування (networking) все ще залишається фрагментарним і недостатньо охоплює потреби ІТ-спільноти. Більшість існуючих рішень орієнтовані або на обмін кодом і технічною допомогою, або на вузькоспеціалізовані менторські сервіси. Водночас інтегроване середовище для професійної взаємодії, обміну досвідом, навчання та підтримки проєктної діяльності ІТ-розробників є надзвичайно актуальним.

У цьому контексті виникає потреба у створенні ресурсу соціального нетворкінгу нового типу — такого, що забезпечить не лише спілкування, а й умови для формування спільнот, розвитку кар'єри, пошуку менторів або команд для реалізації ІТ-проєктів. Реалізація подібного веб-ресурсу вимагає не лише продуманого архітектурного рішення, але й врахування специфіки взаємодії ІТ-фахівців як цільової аудиторії.

Дана дипломна робота спрямована на створення повноцінної платформи соціального нетворкінгу для ІТ-розробників із функціональністю автентифікації, управління профілями, стрічкою публікацій, інтерфейсами взаємодії та засобами адміністрування. У процесі дослідження було виконано моделювання системи, реалізацію програмної частини та тестування результатів, що дозволяє оцінити життєздатність проєкту та його потенціал у практичному застосуванні.

### **Актуальність роботи**

У ХХІ столітті розвиток цифрових технологій значною мірою вплинув на всі аспекти людської діяльності, зокрема на професійне спілкування та обмін знаннями. Особливо це стосується галузі інформаційних технологій,

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 10   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

яка характеризується високою динамікою, глобальністю та постійною необхідністю навчання й удосконалення навичок. IT-фахівці, як представники однієї з найбільш швидко зростаючих і мобільних професійних спільнот, потребують ефективних платформ для взаємодії, які сприяють професійному зростанню, обміну досвідом, спільному вирішенню технологічних завдань і створенню нових проєктів.

Існуючі сервіси, як-от Stack Overflow, Codementor, Chegg або MentorCruise, виконують окремі функції: надають технічну підтримку, організовують менторські сесії або дають змогу ставити запитання. Проте ці платформи не забезпечують комплексної, постійної, цілеспрямованої взаємодії між розробниками в контексті професійного нетворкінгу. Крім того, у більшості подібних сервісів відсутні засоби для побудови довготривалих зв'язків, формування спільнот за інтересами, обміну не лише технічною інформацією, а й кар'єрними можливостями, а також інструменти для спільного розвитку ідей та проєктів.

Також спостерігається зростаючий інтерес до індивідуального навчання, менторства та кооперації у межах open-source або фриланс-спільнот, що потребує відповідних цифрових платформ з розширеною соціальною функціональністю. Зважаючи на це, виникає потреба у створенні такого ресурсу, який буде не лише інформаційним, а й інтерактивним — із можливістю будувати професійний профіль, демонструвати власні навички, ділитися знаннями, знаходити менторів або учнів, працювати в команді над спільними завданнями, організовувати спільноти за напрямками спеціалізації (frontend, backend, data science тощо).

Особливу актуальність проєкт набуває у контексті глобалізації ринку праці, коли IT-фахівці з різних країн і культур працюють віддалено та потребують нових ефективних способів організації спілкування й кооперації. Створення такої платформи також сприятиме формуванню культурного капіталу, збереженню знань і підтримці новачків у професії.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 11   |

Таким чином, реалізація ресурсу соціального нетворкінгу, спеціально адаптованого до потреб ІТ-розробників, є актуальним напрямом дослідження й практичної розробки. Такий ресурс може стати потужним інструментом у професійному розвитку фахівців, підтримці менторства, просуванні інновацій і формуванні нових підходів до організації комунікації в межах ІТ-галузі.

**Мета роботи** - розробка та реалізація ресурсу соціального нетворкінгу, призначеного для професійної взаємодії ІТ-розробників, що включає архітектуру системи, інтерфейси, механізми автентифікації, управління контентом і можливості для розширення функціоналу.

#### **Завдання дослідження**

1. Провести аналіз предметної області соціального нетворкінгу та існуючих рішень для ІТ-сфери.
2. Визначити функціональні та технічні вимоги до ресурсу.
3. Розробити архітектурну модель системи.
4. Побудувати діаграми варіантів використання, потоків даних та послідовності.
5. Реалізувати клієнтську та серверну частину ресурсу.

**Об'єкт дослідження** - процес професійної взаємодії між ІТ-розробниками в цифровому середовищі.

**Предмет дослідження** - архітектурні, функціональні та програмні рішення, що забезпечують розробку ресурсу соціального нетворкінгу для ІТ-розробників.

#### **Методи дослідження**

- аналітичний метод (аналіз існуючих платформ і технологій);
- системний аналіз (побудова архітектури та специфікація вимог);
- методи об'єктно-орієнтованого моделювання (UML-діаграми);
- методи програмної реалізації (frontend/backend розробка);
- експериментальні методи (модульне та користувачське тестування).

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 12   |

### **Наукова новизна**

Запропоновано архітектуру спеціалізованої платформи соціального нетворкінгу, орієнтованої виключно на потреби ІТ-розробників, що поєднує функції професійного спілкування, менторства, профілювання компетенцій та управління контентом у межах одного інтегрованого рішення.

### **Практичне застосування**

Результати роботи можуть бути використані при розробці та впровадженні спеціалізованих соціальних платформ у ІТ-компаніях, стартап-інкубаторах, навчальних ІТ-програмах, а також для створення професійних спільнот ІТ-розробників у межах open-source та комерційних ініціатив.

Бакалаврська робота містить 75 сторінок, 23 рисунки, 3 розділи список використаних джерел із 33 найменуваннями, 1 додаток.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 13   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

# РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗРОБКИ РЕСУРСІВ СОЦІАЛЬНОГО НЕТВОРКІНГУ

## 1.1. Передумови розробки проекту соціального нетворкінгу для ІТ-розробників

Даний проект сфокусований на розробці та імплементації веб-орієнтованої дошки оголошень, що функціонуватиме як спеціалізований ресурс соціального нетворкінгу для ІТ-розробників. Основною метою є створення ефективної платформи для взаємодії між початківцями у сфері комп'ютерних наук та досвідченими розробниками, сприяючи прискореному освоєнню програмних дисциплін.

На платформі "Соціальна мережа для розробників програмного забезпечення" розробники-початківці, які стикаються з труднощами у програмуванні, зможуть отримувати кваліфіковану допомогу від своїх старших колег. Автентифіковані користувачі матимуть можливість висловлювати власні думки щодо кожного розміщеного матеріалу шляхом додавання коментарів. Кожен пост, а також коментарі до нього, будуть оснащені функціоналом "подобається" та "не подобається". Це дозволить користувачам швидко оцінювати релевантність та корисність інформації, приймаючи рішення на основі кумулятивної кількості позитивних оцінок, без необхідності детального перегляду всіх коментарів.

### *1.1.1. Архітектура та функціонал*

Система передбачає надійні механізми безпеки, включаючи систему автентифікації користувачів та контрольовані маршрути доступу. Це означає, що доступ до певних функціональних можливостей та інформації буде обмежений відповідно до статусу автентифікації користувача.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 14   |

Важливою функціональною особливістю проекту є можливість для користувачів створювати та розвивати персональні портфоліо. Ці портфоліо слугуватимуть ключовим інструментом для представлення професійних навичок та досягнень, що в подальшому може сприяти отриманню стажування та працевлаштування у сфері ІТ.

Інтеграція зазначених функціональних можливостей покликана забезпечити, що "Соціальна мережа для розробників програмного забезпечення" суттєво оптимізує часові витрати майбутніх розробників на вивчення та освоєння комп'ютерного програмування.

### *1.1.2. Аналіз існуючих рішень та обґрунтування необхідності*

На сучасному етапі значна кількість молодих фахівців у галузі розробки програмного забезпечення стикається з обмеженими можливостями для отримання оперативної допомоги та вирішення проблемних питань. Існуючі платформи, такі як Stack Overflow, хоча й надають широкі можливості для взаємодії між розробниками, вимагають від нових користувачів попереднього активного внеску у вигляді відповідей на запитання перед тим, як вони зможуть ставити власні. Альтернативні ресурси, наприклад Codementor та Chegg, функціонують за моделлю платної підписки, що може бути бар'єром для початківців. Таким чином, існує потреба у створенні доступного та ефективного ресурсу соціального нетворкінгу, орієнтованого на взаємодопомогу в середовищі ІТ-розробників.

## **1.2. Актуальність розробки ресурсу соціального нетворкінгу для взаємодії ІТ-розробників**

Розробка спеціалізованого ресурсу соціального нетворкінгу для взаємодії ІТ-розробників є актуальною в сучасному динамічному світі технологій, особливо з огляду на наступні фактори:

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 15   |

1. Прискорений розвиток ІТ-індустрії та постійна потреба в кваліфікованих кадрах.

ІТ-сектор демонструє експоненційне зростання, що створює постійний попит на нових, добре підготовлених спеціалістів. Проте, швидкість технологічних змін вимагає безперервного навчання та адаптації. Новачкам часто бракує системних знань та практичного досвіду, що уповільнює їхнє входження в професію. Створення платформи, що забезпечує швидкий доступ до знань та менторства, є ключовим для прискореної підготовки високоякісних фахівців.

2. Дефіцит ефективних механізмів менторства та підтримки для початківців.

Існуючі освітні програми та онлайн-ресурси часто не можуть повністю задовольнити потребу в індивідуальній допомозі та менторстві. Новачки стикаються з реальними проектними завданнями та проблемами, для яких універсальні відповіді є недостатніми. Платформи на кшталт Stack Overflow, хоча й корисні, не завжди надають глибоке, контекстуалізоване менторство. Платні консультації є доступними не для всіх, особливо для студентів або тих, хто тільки починає свій шлях. Розробка безкоштовного або недорогого ресурсу для менторства заповнює цю прогалину.

3. Необхідність формування професійних спільнот та обміну досвідом.

Успішна кар'єра в ІТ значною мірою залежить від здатності до нетворкінгу та обміну знаннями. Можливість спілкуватися з досвідченими колегами, отримувати зворотний зв'язок, ділитися власними напрацюваннями та вчитися на чужому досвіді є неоціненною. Платформа, яка сприяє формуванню активної спільноти, де знання циркулюють вільно та ефективно, посилює професійний розвиток кожного учасника.

4. Сприяння формуванню персонального бренду та портфоліо.

У конкурентному середовищі ІТ-ринку праці наявність якісного портфоліо та активний профіль у професійній спільноті є значною

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 16   |

перевагою. Можливість демонструвати свої проекти, отримувати рекомендації та оцінки від інших розробників на єдиній платформі спрощує процес пошуку стажування та працевлаштування, а також дозволяє компаніям швидше знаходити талановитих кандидатів.

#### 5. Оптимізація процесу навчання та вирішення проблем.

Замість витратити години на пошук інформації та відлагодження коду, початківці можуть отримати швидку та цільову допомогу від досвідчених розробників. Це не тільки економить час, але й запобігає розчаруванню, що часто виникає при самотійному вирішенні складних завдань. Ефективний обмін знаннями через коментарі, лайки/дизлайки та рейтинги коментарів дозволяє швидко виділяти найбільш корисні рішення.

#### 6. Зміцнення локальних та глобальних ІТ-екосистем.

Створення такого ресурсу сприяє не лише індивідуальному розвитку, але й зміцненню всієї ІТ-екосистеми. Він може стати платформою для колаборації над проектами, обміну ідеями, проведення вебінарів чи онлайн-зустрічей, що сприятиме інноваціям та розвитку галузі в цілому, особливо в регіонах з активним розвитком ІТ (наприклад, в Україні).

Узагальнюючи, розробка даного ресурсу є не просто створенням чергового веб-додатку, а стратегічною інвестицією в розвиток людського капіталу в ІТ-сфері, що матиме довгострокові позитивні наслідки для індивідуальних кар'єр, компаній та технологічного прогресу загалом.

### **1.3. Аналіз існуючих платформ обміну досвідом в області ІТ розробки**

#### *1.3.1. Онлайн-сервіс Codementor*

Codementor — це онлайн-сервіс, який спеціалізується на з'єднанні користувачів, що потребують допомоги у програмуванні, з досвідченими менторами та фріланс-розробниками. Вона була заснована у 2013 році і

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 17   |

позиціонується як "цифровий рятівний круг" для тих, хто стикається з труднощами у кодуванні або потребує експертної підтримки для своїх проєктів.

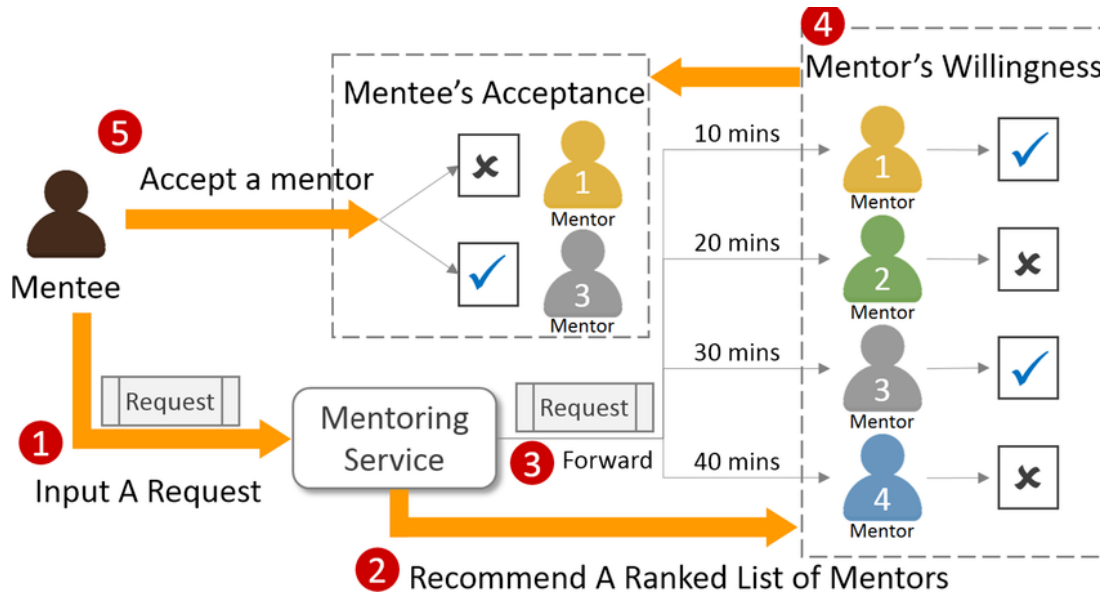


Рисунок 1.1 - Огляд загальної процедури роботи Codementor

Наведемо основні функції та можливості Codementor.

On-demand Mentorship - це основна послуга Codementor. Користувачі можуть знайти та зв'язатися з кваліфікованими менторами для отримання індивідуальних консультацій у режимі реального часу. Це може включати:

1. Дебагінг (налагодження коду): Ментори допомагають виявити та виправити помилки у коді.
2. Огляд коду (Code Review): Оцінка якості коду, пропозиції щодо його оптимізації та вдосконалення.
3. Розв'язання складних проблем: Допомога у розв'язанні технічних завдань, які неможливо вирішити за допомогою стандартних ресурсів (наприклад, Stack Overflow).
4. Персоналізоване навчання: Індивідуальні уроки та пояснення складних концепцій програмування.

Платформа надає детальні профілі менторів, включаючи їхній досвід, спеціалізацію, відгуки та рейтинги. Це дозволяє користувачам обрати найбільш відповідного фахівця для своїх потреб. Ментори встановлюють власні ставки, які можуть варіюватися від \$10-15 за 15 хвилин і вище, залежно від їхнього досвіду та кваліфікації.

Спільний доступ та інтерактивні сесії - сесії з менторами зазвичай проходять у форматі онлайн-зустрічей зі спільним доступом до екрана, що дозволяє ментору безпосередньо бачити код і надавати допомогу.

Окрім менторства, Codementor також пропонує платформу CodementorX, яка дозволяє компаніям та приватним особам наймати перевірених фріланс-розробників для виконання проектів. Це розширює функціонал платформи від суто освітнього до рекрутингового та проектного. Ментори на CodementorX проходять ретельний відбір.

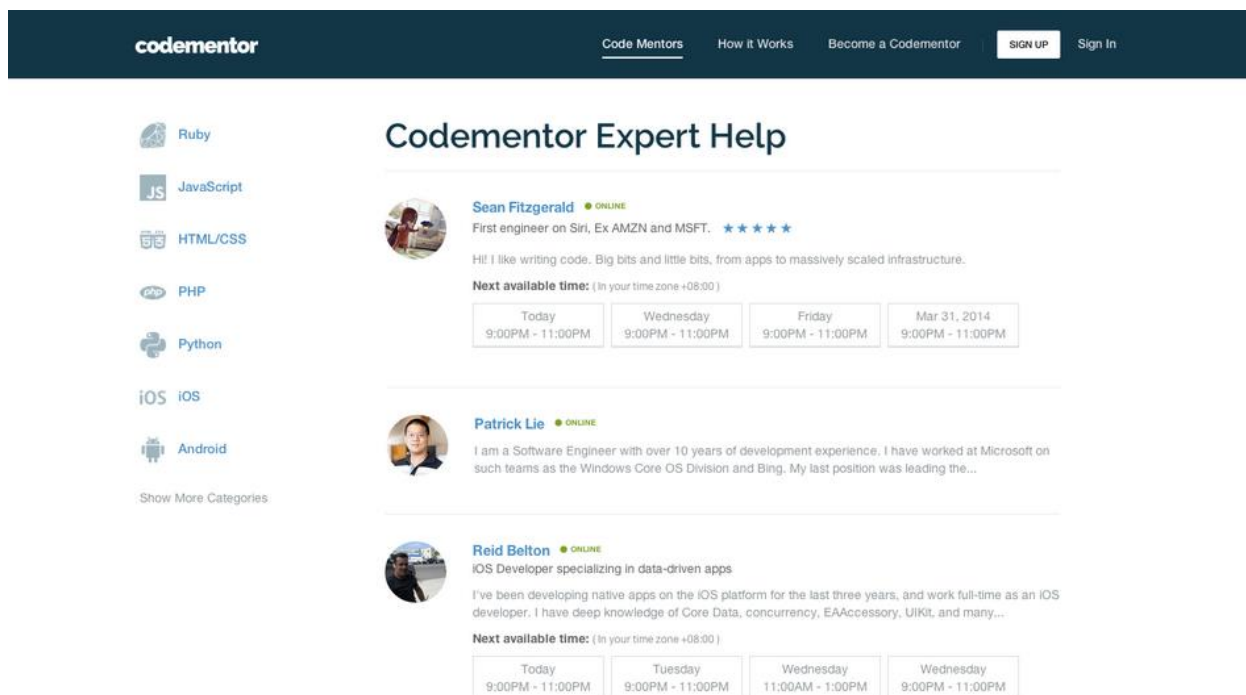


Рисунок 1.2 – Вигляд платформи Codementor

На платформі представлені ментори, які спеціалізуються на широкому спектрі мов програмування (наприклад, Python, JavaScript, Ruby, PHP),

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 19   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

фреймворків, баз даних та інших ІТ-технологій. Codementor працює за моделлю "плати за сесію" або за погодинну оплату. Користувачі оплачують час, проведений з ментором. Крім того, стягується сервісний збір, який покриває операційні витрати платформи та комісії за обробку платежів.

Переваги Codementor:

1. Персоналізована допомога.

Можливість отримати індивідуальну підтримку від досвідченого фахівця, що дозволяє вирішувати проблеми, які не можуть бути розв'язані за допомогою загальних посібників.

2. Економія часу.

Швидке отримання відповідей та рішень, що прискорює процес навчання та розробки.

3. Доступ до експертів.

Можливість консультиватися з висококваліфікованими розробниками, які мають значний досвід у своїй галузі.

Недоліки:

1. Головним недоліком Codementor, особливо для початківців та студентів, є платний характер послуг. Це створює бар'єр для доступу до знань та підтримки.

2. Взаємодія з ментором здебільшого є транзакційною (оплата за сесію), що може обмежувати формування довгострокових менторських стосунків або неформальних спільнот.

3. Хоча є можливість найму, платформа не надає безкоштовного майданчика для вільної взаємодії, обміну ідеями та менторства між початківцями та досвідченими розробниками без фінансової складової.

Таким чином, Codementor є ефективним інструментом для швидкого отримання експертної допомоги у програмуванні, але його модель оплати та фокус на індивідуальних платних сесіях створюють нішу для альтернативних

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 20   |

платформ, які пропонують безоплатне або більш доступне менторство та формування спільнот.

### *1.3.2. Платформа Chegg*

Платформа Chegg — це американська освітньо-технологічна компанія, яка пропонує широкий спектр послуг для студентів, спрямованих на підтримку їхнього навчання та академічного успіху. Заснована у 2005 році, Chegg еволюціонувала від сервісу з оренди підручників до комплексного провайдера освітніх ресурсів.

Оренда та продаж підручників (Textbook Rentals and Sales) - це була початкова і одна з найпопулярніших послуг Chegg. Компанія пропонує студентам орендувати фізичні або цифрові підручники за ціною, значно нижчою, ніж покупка нових. Це допомагає студентам економити значні кошти.

Chegg Study - це один з ключових продуктів Chegg, який пропонує:

- Покрокові рішення до завдань з підручників (Step-by-step Textbook Solutions): Доступ до мільйонів детальних рішень до завдань, що зустрічаються в популярних університетських підручниках з різних дисциплін, включаючи комп'ютерні науки, математику, інженерію, фізику, хімію тощо. Ці рішення допомагають студентам зрозуміти, як розв'язувати задачі, а не просто отримати відповідь.

- Експертна допомога з питаннями та відповідями (Expert Q&A): Можливість задавати власні питання, на які відповідають експерти Chegg (часто вчителі або викладачі). Зазвичай відповідь надходить протягом 24 годин. Кількість питань, які можна поставити, залежить від обраного тарифного плану.

- Практичні завдання та посібники (Practice Problems & Study Guides): Додаткові матеріали для самостійної практики та підготовки до іспитів.

|      |      |          |        |      |                          |      |
|------|------|----------|--------|------|--------------------------|------|
|      |      |          |        |      | БР.ІІІ – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                          | 21   |

Chegg Writing - сервіс для допомоги у написанні академічних робіт:

- Перевірка на плагіат (Plagiarism Checker): Інструмент для виявлення запозичень у тексті.

- Перевірка граматики та стилю (Grammar and Style Checker): Допомагає покращити якість написання.

- Створення посилань (Citation Generator): Автоматичне генерування посилань у різних стилях (APA, MLA, Chicago тощо).

- Експертне редагування (Expert Proofreading): Можливість отримати професійне редагування та відгук щодо академічної роботи (зазвичай, за додаткову плату або у преміум-пакетах).

Chegg Math Solver - інструмент, який допомагає вирішувати математичні задачі, надаючи покрокові рішення та пояснення.

Флеш-картки (Flashcards) - безкоштовний інструмент для створення та вивчення флеш-карток, що сприяє запам'ятовуванню ключових термінів та концепцій.

Онлайн-репетиторство (Online Tutoring). Раніше Chegg пропонував живі онлайн-репетиторські сесії. Проте, станом на 2025 рік, Chegg припинив надавати послуги живого покрокового репетиторства, натомість зосередившись на своїх підписках Chegg Study та Chegg Study Pack, які надають доступ до мільйонів рішень та можливість задавати питання експертам. Тобто фокус змістився з "живих" менторських сесій на "запитання-відповідь" та доступ до великої бази знань.

Основні послуги Chegg, такі як Chegg Study та Chegg Writing, надаються за моделлю підписки. Студенти сплачують щомісячну абонементну плату для отримання доступу до функціоналу. Існують різні тарифні плани (наприклад, Chegg Study та Chegg Study Pack), які пропонують різний набір функцій за відповідну ціну (наприклад, Chegg Study коштує близько \$15.95 на місяць, а Chegg Study Pack — близько \$19.95 на місяць, станом на 2025 рік).

|      |      |          |        |      |                          |      |
|------|------|----------|--------|------|--------------------------|------|
|      |      |          |        |      | БР.ІІІ – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                          | 22   |

Хоча Chegg і не є вузькоспеціалізованою платформою для ІТ-розробників, як Codementor, вона все ж є популярним ресурсом серед студентів комп'ютерних наук та інженерних спеціальностей завдяки своїм рішенням до завдань з підручників та Q&A сервісу. Багато студентів використовують Chegg для розуміння алгоритмів, структур даних, математичних концепцій та інших фундаментальних аспектів комп'ютерних наук.

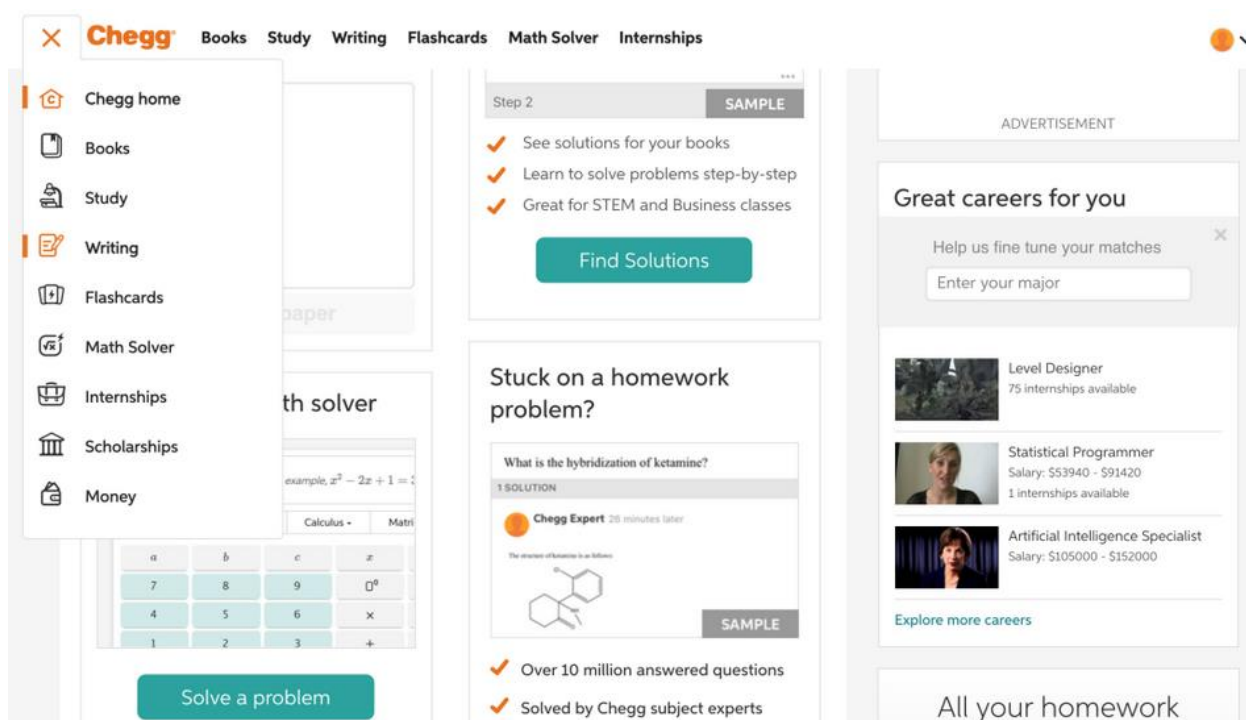


Рисунок 1.3 – Інтерфейс платформи Chegg

Основним бар'єром для багатьох студентів є необхідність платної підписки для доступу до більшості цінних ресурсів. Це обмежує доступність для тих, хто не може дозволити собі щомісячні витрати.

На відміну від ранніх версій або деяких конкурентів, Chegg більше не фокусується на живому індивідуальному репетиторстві. Це означає, що студенти не можуть отримати негайну, інтерактивну допомогу з дебагінгом або розбором складних проектів у режимі реального часу.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 23   |

Хоча Chegg допомагає з академічними завданнями, він менше орієнтований на практичні аспекти розробки програмного забезпечення, такі як спільна робота над проектами, створення портфоліо у специфічному для розробників форматі, пошук стажувань, орієнтованих саме на ІТ-сферу, або неформальний обмін досвідом у спільноті розробників.

Через доступність готових рішень, Chegg іноді використовується не за призначенням, що може призвести до академічної недоброчесності, хоча сама платформа заявляє, що її інструменти призначені для допомоги у розумінні, а не для списування.

Таким чином, Chegg є цінним академічним ресурсом, але його фокус на платній підписці та зміщення від інтерактивного репетиторства вказують на необхідність створення доступніших платформ, які сприяють формуванню спільноти, менторству та практичному розвитку ІТ-розробників без значних фінансових бар'єрів.

### *1.3.3. Онлайн-платформа MentorCruise*

MentorCruise — це онлайн-платформа, що спеціалізується на з'єднанні студентів та фахівців з досвідченими менторами з різних галузей, але з особливим акцентом на технологічний сектор. Вона була заснована у 2017 році з метою зробити менторство більш доступним, прозорим та ефективним.

На відміну від деяких інших платформ, які пропонують короткострокові "сесії на вимогу", MentorCruise більше орієнтована на довгострокові менторські стосунки. Це дозволяє менторам та менті (тим, хто навчається) будувати глибші зв'язки та працювати над довгостроковими цілями.

На платформі представлені ментори з різних сфер, включаючи розробку програмного забезпечення (Software Engineering), машинне навчання (Machine Learning), дизайн (Design), продукт-менеджмент (Product

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 24   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

Management), бізнес та інші. Багато менторів мають досвід роботи у провідних світових компаніях (наприклад, Google, Netflix, Amazon).

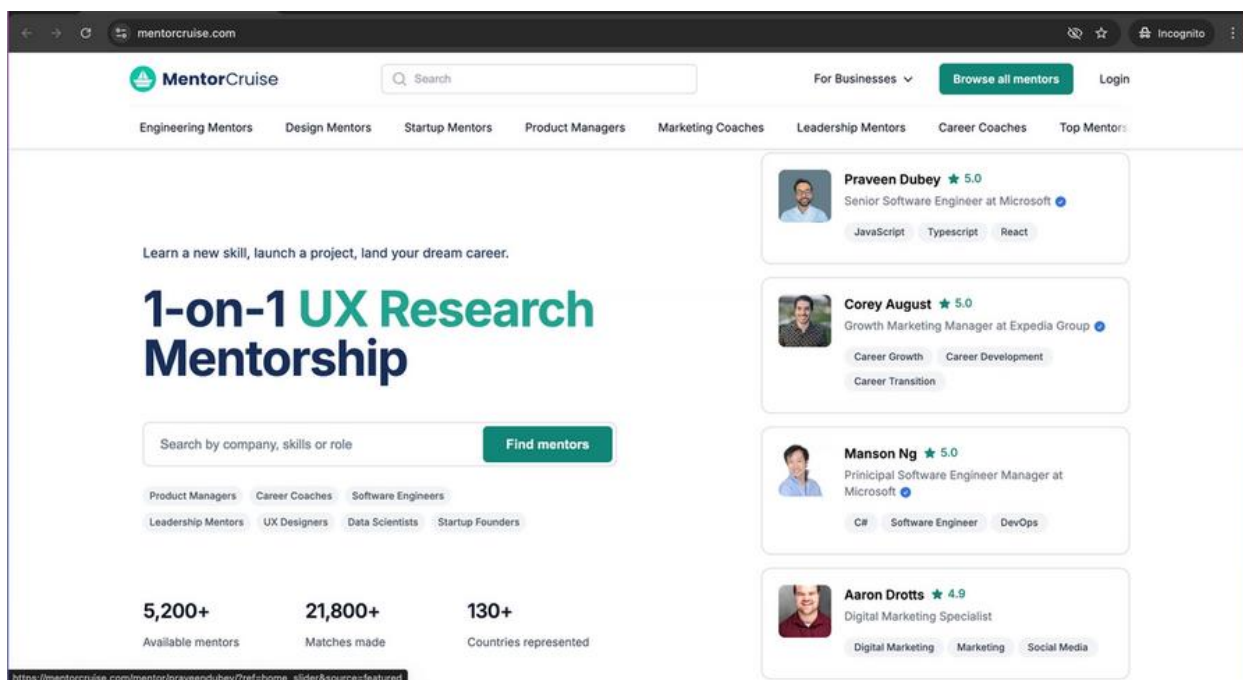


Рисунок 1.4 – Онлайн-платформа MentorCruise

Основна модель взаємодії передбачає щомісячні підписки на послуги ментора. Це включає:

- Необмежений чат: Можливість спілкуватися з ментором через вбудований чат платформи.
- Регулярні 1-на-1 дзвінки: Зазвичай кілька відео- або голосових дзвінків на місяць, кількість яких залежить від обраного тарифного плану ментора.
- Індивідуальні навчальні плани: Ментори часто допомагають менті скласти персоналізовані плани розвитку, що включають цілі, етапи та рекомендовані ресурси.
- Огляд коду, підготовка до співбесід, кар'єрні консультації: Ментори надають широкий спектр допомоги, від технічних аспектів до розвитку м'яких навичок та кар'єрного планування.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 25   |

Окрім місячних підписок, багато менторів пропонують одноразові сесії або пакети послуг, такі як:

- Вступні дзвінки (Intro Session): Короткі безкоштовні або недорогі сесії для знайомства та обговорення потреб.
- Огляд резюме (CV Review): Допомога у покращенні резюме.
- Огляд портфоліо (Portfolio Review): Оцінка та рекомендації щодо проектів у портфоліо.
- Підготовка до співбесіди (Interview Preparation): Імітація співбесід та зворотний зв'язок.

Профілі менторів містять детальну інформацію про їхній досвід, спеціалізацію, відгуки та рейтинги від попередніх менті, що допомагає користувачам зробити обґрунтований вибір. Платформа надає вбудовані функції для обміну повідомленнями та здійснення відеодзвінків, що забезпечує зручну та інтегровану взаємодію.

Ментори встановлюють власні ціни за місячні підписки або одноразові сесії. Ціни можуть значно варіюватися залежно від досвіду ментора та обсягу послуг. MentorCruise бере комісію (зазвичай 16-20%) від платежів, отриманих менторами. Зазвичай, місячні підписки починаються від \$50-100 і можуть сягати кількох сотень доларів на місяць.

Переваги MentorCruise:

1. Орієнтація на довгостроковий розвиток.

Ідеально підходить для тих, хто шукає послідовну та тривалу підтримку у своєму професійному зростанні.

2. Персоналізований підхід.

Можливість отримати індивідуальні навчальні плани та поради, адаптовані до конкретних потреб.

3. Висока якість менторів.

Багато менторів є досвідченими фахівцями з провідних компаній.

4. Зручна платформа.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 26   |

Інтуїтивний інтерфейс та інтегровані інструменти для комунікації.

Як і Codementor, MentorCruise є платною платформою. Хоча існують різні цінові категорії, для багатьох початківців або студентів це може бути значним фінансовим бар'єром. Хоча є елементи спільноти (наприклад, "Neighborhood" форум), основний акцент робиться на 1-на-1 взаємодії, що може бути менш підходящим для тих, хто шукає широку спільноту для обміну знаннями та неформального спілкування.

Загалом, MentorCruise є сильною платформою для тих, хто готовий інвестувати у своє професійне менторство та шукає системну та довгострокову підтримку від досвідчених фахівців.

#### *1.3.4. Веб-ресурс Stack Overflow*

Stack Overflow — це найпопулярніший та найбільший веб-сайт запитань та відповідей для професійних та ентузіастів-програмістів, розроблений та запущений у 2008 році Джеффом Етвудом (Jeff Atwood) та Джоелом Спольськи (Joel Spolsky). Він є частиною ширшої мережі Stack Exchange. Stack Overflow став де-факто ресурсом для мільйонів розробників по всьому світу, які шукають допомогу у вирішенні проблем кодування, розумінні концепцій або обміні знаннями.

Core-функціонал платформи полягає у можливості ставити запитання, пов'язані з програмуванням (наприклад, проблеми з кодом, помилки, архітектурні рішення, вибір технологій), та отримувати відповіді від спільноти. Користувачі можуть голосувати за запитання та відповіді ("upvote" / "downvote"), що допомагає виділяти найкорисніший та найточніший контент. Запитання та відповіді організуються за допомогою тегів (tags) (наприклад, javascript, python, reactjs, mongodb), що полегшує пошук та категоризацію інформації.

Stack Overflow використовує репутаційну систему, яка заохочує користувачів надавати якісні відповіді та задавати добре сформульовані

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 27   |

запитання. Користувачі заробляють репутацію, коли їхні запитання або відповіді отримують "upvote" або коли їхні відповіді позначаються як "прийняті" (accepted answer) автором запитання. Вищий рівень репутації відкриває додаткові привілеї, такі як можливість голосувати, коментувати, редагувати чужі дописи, закривати запитання, тощо. Це допомагає спільноті самостійно модерувати контент та підтримувати високу якість.

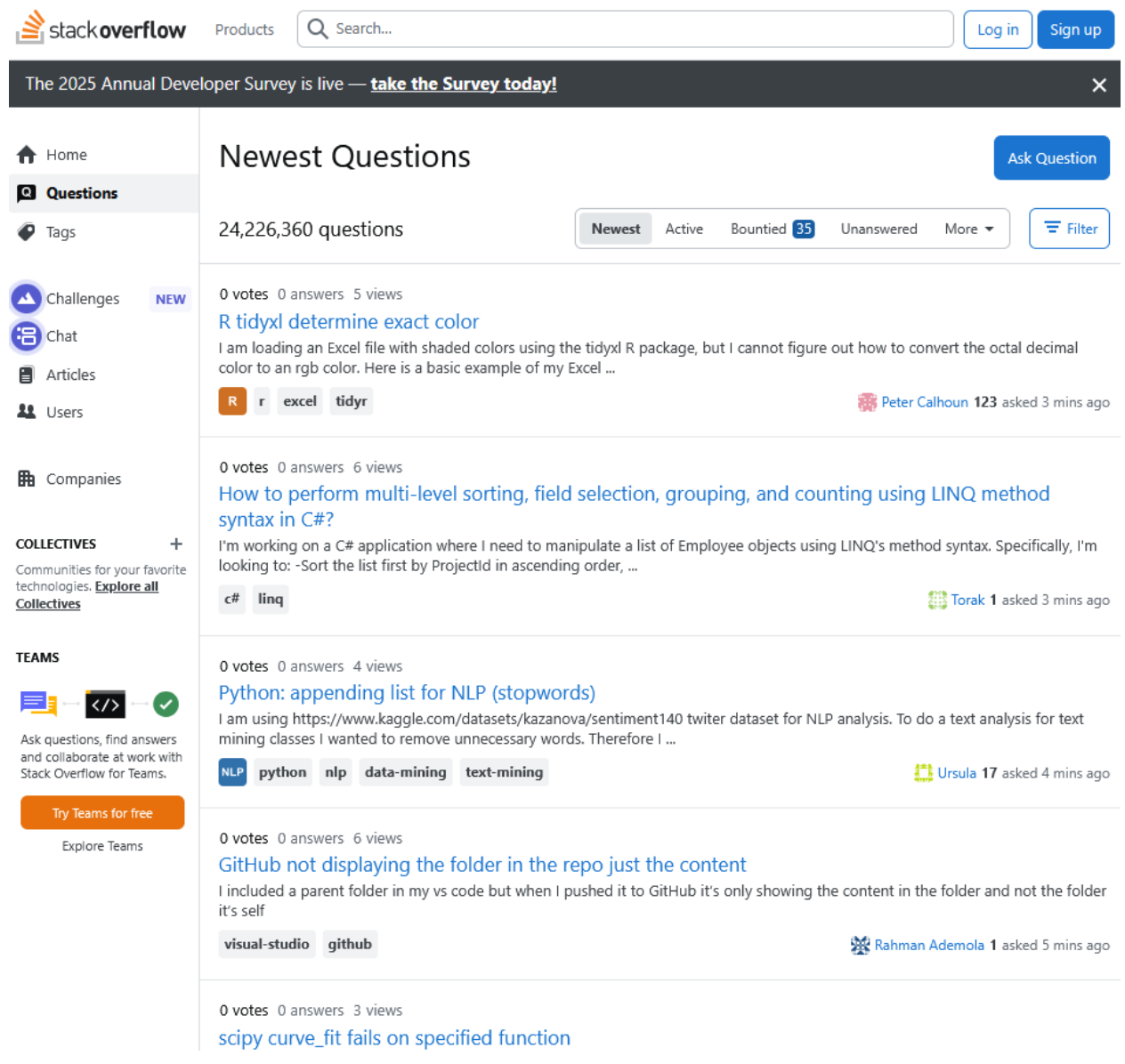


Рисунок 1.5 – Вигляд веб-ресурсу Stack Overflow

Платформа значною мірою покладається на саморегулювання та модерацію з боку активних учасників спільноти. Високореєтингові

|      |      |          |        |      |                         |  |  |  |      |
|------|------|----------|--------|------|-------------------------|--|--|--|------|
|      |      |          |        |      |                         |  |  |  | Арк. |
|      |      |          |        |      |                         |  |  |  | 28   |
| Змн. | Арк. | № докум. | Підпис | Дата | БР.ІП – 44.00.00.000 ПЗ |  |  |  |      |

користувачі мають можливість редагувати запитання та відповіді, голосувати за закриття (duplication, off-topic, unclear) або повторне відкриття запитань, а також позначати неприйнятний контент.

З часом Stack Overflow став величезною, динамічно зростаючою енциклопедією програмних знань. Ймовірність знайти відповідь на своє запитання через пошук Google, що веде на Stack Overflow, є надзвичайно високою. Stack Overflow також пропонує розділи для пошуку роботи (Stack Overflow Careers, тепер Integrated within the main platform) та інші спільнотні функції, хоча його основна цінність залишається в Q&A.

Переваги Stack Overflow:

1. Миттєвий доступ до знань.

Дуже висока ймовірність швидко знайти відповідь на майже будь-яке питання з програмування.

2. Величезна база знань.

Мільйони запитань та відповідей, що охоплюють практично всі технології.

3. Авторитетність.

Система голосування та модерації допомагає виділяти якісні та перевірені рішення.

4. Активна спільнота.

Мільйони активних користувачів, готових допомогти.

5. Безоплатність.

Основний функціонал доступний безкоштовно.

Недоліки Stack Overflow

- Це не платформа для індивідуального навчання чи довгострокового менторства. Відповіді можуть бути короткими та не завжди надавати повний контекст.

- Якість та своєчасність відповідей залежить від активності спільноти в конкретній галузі.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 29   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

- Існує певна культура формулювання запитань. Погано сформульовані або дубльовані запитання можуть бути швидко закриті або отримати негативні голоси.

- Технології розвиваються швидко, і деякі старі відповіді можуть бути вже неактуальними або не найкращими рішеннями.

- Іноді користувачі скаржаться на те, що спільнота може бути надто критичною або навіть ворожою до новачків, які задають "прості" або "дубльовані" запитання.

Stack Overflow є незамінним інструментом для кожного розробника, що надає безпрецедентний доступ до колективних знань світової спільноти програмістів. Хоча він не замінює формальне навчання або індивідуальне менторство, його роль як джерела швидких відповідей та практичних рішень є фундаментальною у щоденній роботі більшості ІТ-спеціалістів.

### *1.3.5. Онлайн платформа Amazing Design People List*

ADPList (Amazing Design People List) — це глобальна онлайн-платформа, яка спеціалізується на з'єднанні студентів та професіоналів з досвідченими менторами з різних галузей, зокрема дизайну, продуктового менеджменту, розробки програмного забезпечення, маркетингу та інших технологічних напрямків. Заснована у 2020 році Феліксом Лі (Felix Lee) під час пандемії COVID-19 з метою зробити менторство доступним для всіх, ADPList швидко здобула популярність завдяки своїй моделі безоплатного менторства.

Ключовою відмінністю ADPList є те, що ментори надають свої послуги безкоштовно. Це принципове рішення, спрямоване на демократизацію доступу до знань та досвіду, що робить платформу надзвичайно привабливою для студентів, початківців та тих, хто стикається з фінансовими обмеженнями. Ментори є волонтерами, які присвячують свій час, щоб допомогти іншим.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 30   |

Хоча назва походить від "Design People List", платформа значно розширила своє охоплення і включає менторів з:

- Дизайну (UX/UI, Product Design, Visual Design)
- Продуктового менеджменту
- Розробки програмного забезпечення (Software Engineering, Frontend, Backend, Mobile)
- Маркетингу та Бренд-менеджменту
- Даних та Аналітики
- Управління проектами
- Кар'єрного коучингу та багато іншого.

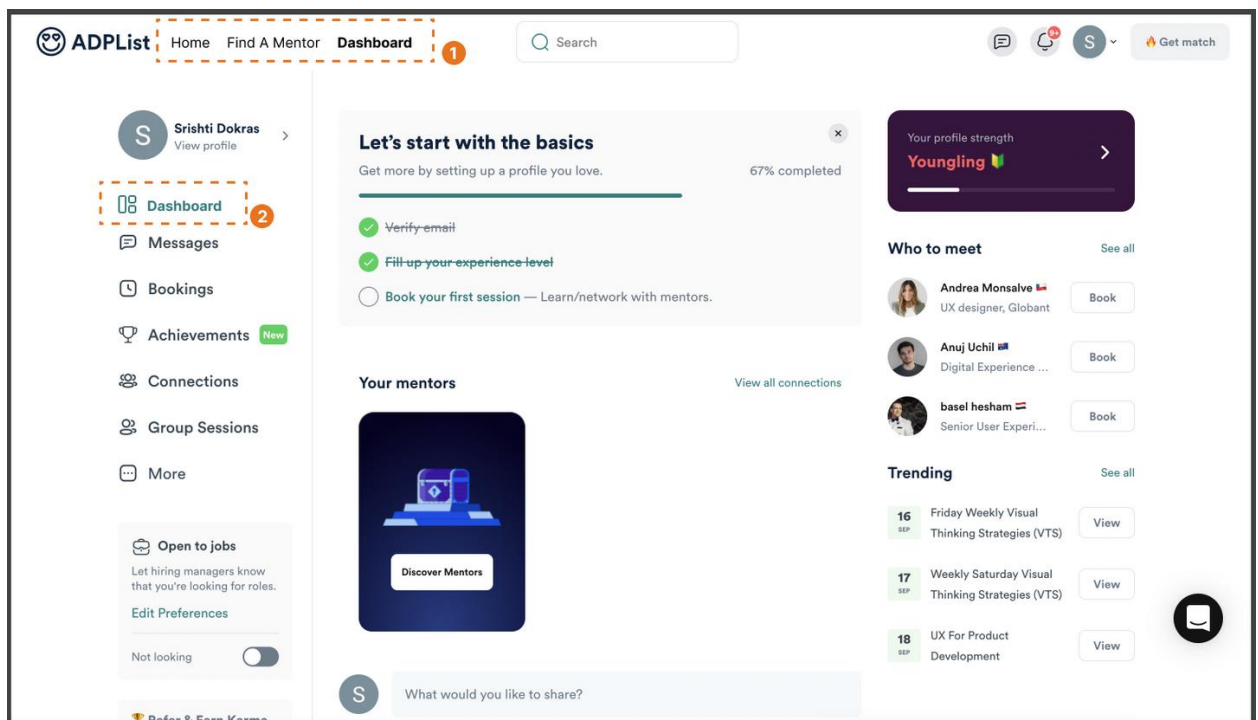


Рисунок 1.6 – Платформа ADPList

Основний формат взаємодії на ADPList — це індивідуальні консультації та менторські сесії (1:1), які зазвичай проводяться через відеозв'язок (наприклад, Google Meet, Zoom). Окрім індивідуальних сесій, ADPList організовує вебінари, панельні дискусії та інші спільнотні заходи з

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      |                         | Арк. |
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | 31   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

провідними експертами, що розширює можливості для навчання та нетворкінгу.

#### Переваги ADPList

- Безоплатність: Це ключова перевага, що робить якісне менторство доступним для широкої аудиторії.

- Глобальний доступ: Платформа з'єднує людей з різних куточків світу, надаючи доступ до різноманітного досвіду.

- Висока якість менторів: Багато менторів є досвідченими професіоналами з відомих компаній, які готові ділитися своїми знаннями.

- Зручний інтерфейс: Інтуїтивно зрозумілий процес пошуку та бронювання сесій.

- Підтримка кар'єрного росту: Платформа є цінним ресурсом для тих, хто шукає поради щодо кар'єри, розвитку навичок або зміни професії.

Кожен ментор має детальний профіль, який включає:

- Інформацію про їхню поточну та попередні компанії (часто з провідних світових tech-гігантів).

- Спеціалізації та галузі експертизи.

- Досвід та досягнення.

- Доступні теми для менторства.

- Календар доступності.

- Відгуки від попередніх користувачів.

Це дозволяє користувачам знайти найбільш відповідного ментора для своїх конкретних потреб.

Оскільки менторство безоплатне, доступність дуже популярних менторів може бути обмеженою через високий попит. Хоча більшість менторів висококваліфіковані, якість сесій може варіюватися залежно від індивідуального підходу ментора. Сесії зазвичай є одноразовими або короткостроковими, що може бути недостатньо для глибокого,

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 32   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

довгострокового менторства, хоча деякі пари можуть продовжувати взаємодію.

ADPList швидко стала одним з найважливіших ресурсів у сфері професійного розвитку, особливо в технологічній галузі. Її модель безоплатного менторства дозволила тисячам людей отримати доступ до цінних знань та порад, що сприяє розвитку талантів та побудові кар'єри без фінансових бар'єрів. Платформа успішно демонструє силу спільноти та волонтерства у глобальному масштабі.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 33   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

## РОЗДІЛ 2. МОДЕЛЮВАННЯ СТРУКТУРИ ПЛАТФОРМИ СОЦІАЛЬНОГО НЕТВОРКІНГУ ДЛЯ ВЗАЄМОДІЇ ІТ-РОЗРОБНИКІВ

### 2.1. Запропонована архітектура ресурсу соціального нетворкінгу для ІТ-розробників

Пропонована система функціонує як спеціалізований ресурс соціального нетворкінгу, розроблений для оптимізації взаємодії та обміну знаннями між фахівцями у сфері інформаційних технологій.

#### *2.1.1. Інтерфейс користувача та функціонал автентифікації*

Взаємодія з системою починається з головного екрану, який слугує точкою входу для всіх користувачів. Для неавтентифікованих користувачів передбачено можливість ознайомлення з короткими профілями зареєстрованих розробників, що дозволяє швидко оцінити потенціал спільноти. З головної сторінки надаються дві основні опції: реєстрація нового облікового запису або вхід для вже існуючих користувачів.

Після успішної реєстрації та автентифікації, розробники отримують доступ до комплексного функціоналу редагування профілю. Цей функціонал включає:

1. Персоналізація: Можливість завантаження фотографії профілю, зазначення поточного статусу, компанії та місця розташування.
2. Інтеграція з соціальними мережами: Користувачі можуть додавати посилання на свої соціальні профілі, що сприяє розширенню їхньої професійної мережі за межами платформи.
3. Професійна інформація: Можливість додавання резюме профілю, переліку навичок програмування, а також детального опису досвіду роботи, що дозволяє повноцінно презентувати професійні компетенції.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 34   |

4. Освіта: Спеціальний розділ для фіксації освітніх досягнень, включаючи інформацію про відвідування навчальних закладів або спеціалізованих курсів.

5. Динамічне редагування: Автентифіковані користувачі мають повний контроль над своїм профілем, включаючи можливість додавання та видалення інформації про досвід та освіту в будь-який час.

### *2.1.2. Взаємодія у Спільноті (розділ публікацій)*

Центральним елементом взаємодії є розділ публікацій (постів), де всі зареєстровані розробники можуть розміщувати свої коментарі. Цей розділ є ключовим для:

#### 1. Обміну знаннями та вирішення проблем.

Розробники, які стикаються з питаннями або проблемами, можуть публікувати їх у вигляді коментарів. Інші члени спільноти можуть надавати відповіді та допомогу, сприяючи колективному вирішенню технічних завдань.

#### 2. Система оцінки контенту.

Кожен коментар оснащений механізмом "подобається" та "не подобається", що дозволяє користувачам оцінювати релевантність та корисність інформації, а також швидко ідентифікувати найбільш цінні відповіді без необхідності перегляду всіх коментарів.

#### 3. Управління особистими публікаціями.

Користувачі зберігають право видаляти власні коментарі з розділу публікацій, забезпечуючи контроль над своїм контентом.

## **2.2. Специфікація вимог до системи соціального нетворкінгу**

Дана система має чітко визначені вимоги до апаратного та програмного забезпечення, що забезпечують її функціональність та стабільну роботу.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 35   |

### 2.2.1. Апаратні та програмні вимоги до серверної інфраструктури

Для розгортання та функціонування серверної частини додатку передбачаються наступні компоненти:

- Веб-сервер: Застосування платформи Heroku як PaaS (Platform as a Service) забезпечує масштабованість, автоматизоване розгортання та управління серверним середовищем для веб-додатка. Heroku є хмарним рішенням, що дозволяє уникнути необхідності прямого управління фізичними серверами.

- Сервер баз даних: Для зберігання та управління даними застосовується mLab. mLab є хмарним провайдером послуг бази даних MongoDB як сервісу (DBaaS), що спрощує розгортання, масштабування та адміністрування NoSQL бази даних.

### 2.2.2. Програмні вимоги до розробки та функціонування

Реалізація та взаємодія компонентів системи базується на наступному програмному стеку:

- Обліковий запис Heroku: Необхідний для розгортання та управління веб-додатком на платформі Heroku.

- React.js: Використовується для розробки інтерфейсу користувача (UI) на стороні клієнта. React.js є бібліотекою JavaScript для створення односторінкових додатків (SPA) та інтерактивних компонентів, що забезпечує високу продуктивність та динамічність інтерфейсу.

- Node.js: Застосовується для розробки серверної логіки (Backend). Node.js є кросплатформним середовищем виконання JavaScript, що дозволяє створювати ефективні та масштабовані серверні додатки.

- Axios: Використовується як клієнт HTTP для здійснення асинхронних запитів між фронтендом (React.js) та бекендом (Node.js). Axios спрощує взаємодію з API та обробку відповідей.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 36   |

- MongoDB: Вибрана як система управління базами даних (СУБД). MongoDB є документоорієнтованою NoSQL базою даних, що забезпечує гнучкість у зберіганні даних та ефективну роботу з великими обсягами неструктурованої інформації. Взаємодія з MongoDB здійснюється через mLab.

### 2.2.3. Вимоги до клієнтського середовища

Для доступу до функціоналу системи користувачеві необхідний лише робочий стіл (персональний комп'ютер), оснащений будь-яким сучасним веб-браузером. Це забезпечує максимальну доступність та мінімізує вимоги до кінцевого обладнання користувача, оскільки вся обробка даних та логіка додатку виконується на сервері.

## 2.3. Використані програмні інструменти та технології

Реалізація запропонованого ресурсу соціального нетворкінгу для ІТ-розробників базується на сучасному стеку веб-технологій, що забезпечує його функціональність, продуктивність та масштабованість.

### 2.3.1. Технології реалізації фронтенду

React.js: Для розробки клієнтської частини додатка використовується React.js – бібліотека JavaScript, призначена для створення інтерактивних користувацьких інтерфейсів. Вона забезпечує розробку односторінкових додатків (Single Page Applications, SPAs), де початковий завантажувальний код JavaScript виконується безпосередньо у браузері користувача. Такий підхід мінімізує кількість запитів до сервера при навігації між сторінками, що суттєво покращує швидкість відгуку інтерфейсу та загальний користувацький досвід.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 37   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

React заохочує розробку додатків шляхом поділу UI на незалежні, повторно використовувані компоненти. Кожен компонент інкапсулює свою власну логіку та вигляд. Це спрощує управління складним UI, підвищує модульність коду та його повторне використання. Компоненти можуть бути функціональними (простіші, використовують хуки для стану) або класовими (використовують `this.state` та життєві цикли).

Замість того, щоб вказувати, як змінити UI (імперативний підхід), React дозволяє розробникам описувати, яким має бути UI у певний момент часу на основі стану даних. React самостійно визначає, як ефективно оновити UI відповідно до змін стану. Цей підхід спрощує налагодження та розуміння коду, оскільки розробник фокусується на бажаному кінцевому стані, а не на послідовності маніпуляцій DOM. Однією з ключових особливостей React є використання Віртуального DOM (Virtual Document Object Model). Замість безпосередньої маніпуляції реальним DOM браузера (що є повільною операцією), React створює легку копію DOM у пам'яті. Коли стан компонента змінюється, React спочатку оновлює Віртуальний DOM. Потім він виконує "diffing" – порівнює поточний Віртуальний DOM з попереднім, виявляючи мінімальні зміни. Лише ці мінімальні зміни потім застосовуються до реального DOM, що значно підвищує продуктивність рендерингу.

Дані в React передаються у вигляді "props" (властивостей) від батьківських компонентів до дочірніх. Це робить потік даних передбачуваним і спрощує налагодження, оскільки легко відстежити, звідки походять дані. Зміни стану (state) компонента ініціюються самим компонентом або його дочірніми компонентами, але вони завжди призводять до рендерингу компонента та, за необхідності, його дочірніх елементів.

React використовує JSX, синтаксичне розширення JavaScript, яке дозволяє писати UI-структури, що виглядають як HTML, безпосередньо в JavaScript-коді. JSX є декларативним і дозволяє легко вбудовувати JavaScript-вирази в розмітку. Хоча використання JSX не є обов'язковим (можна писати

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 38   |

React-код і без нього, використовуючи `React.createElement()`, воно значно покращує читабельність та зручність розробки.

Запроваджені в React 16.8, хуки дозволяють використовувати стан та інші можливості React у функціональних компонентах без необхідності писати класові компоненти. Найбільш популярні хуки включають `useState` (для управління станом), `useEffect` (для побічних ефектів, таких як отримання даних), `useContext` (для доступу до контексту) тощо. Хуки значно спростили React-розробку та зробили функціональні компоненти більш потужними.

#### Переваги React.js

1. Завдяки Віртуальному DOM та ефективному алгоритму порівняння, React мінімізує кількість операцій з реальним DOM, що призводить до швидшого рендерингу.

2. Модульна архітектура дозволяє створювати компоненти, які можна легко використовувати в різних частинах додатка або в різних проектах.

3. Наявність великої кількості документації, навчальних ресурсів, бібліотек та готових компонентів.

4. React є бібліотекою, а не повноцінним фреймворком, що дає розробникам свободу вибору інших інструментів (роутери, менеджери стану, бібліотеки для роботи з даними).

5. Завдяки інкапсуляції, зміни в одному компоненті рідко впливають на інші частини додатка, що спрощує розробку та налагодження.

#### 2.3.2. Технології реалізації бекенду

Node.js: Серверна частина системи розроблена з використанням Node.js – кросплатформного середовища виконання JavaScript з відкритим вихідним кодом. Node.js є оптимальним рішенням для створення додатків, що вимагають постійного, двостороннього з'єднання між клієнтом і сервером (наприклад, для функціоналу чатів, стрічок новин та веб-повідомлень), завдяки своїй асинхронній, подієво-орієнтованій архітектурі.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 39   |

Axios: Для здійснення HTTP-запитів між клієнтською та серверною частинами додатка застосовується Axios. Це бібліотека HTTP-клієнта на основі промісів (Promises), що працює як у браузерному середовищі, так і в Node.js. Axios уніфікує API для роботи з XMLHttpRequest (у браузері) та http інтерфейсом Node.js, спрощуючи взаємодію з RESTful API.

Axios використовує Promises API, що дозволяє легко працювати з асинхронними операціями. Це робить код більш читабельним, спрощує обробку успішних відповідей (.then()) та помилок (.catch()) за допомогою стандартних Promise-методів.

Одна з ключових переваг Axios полягає в його здатності працювати як на стороні клієнта (у браузері), так і на стороні сервера (у Node.js). Це дозволяє розробникам використовувати один і той же код для виконання HTTP-запитів незалежно від середовища, що спрощує підтримку та розробку ізоморфних (універсальних) додатків. Axios автоматично перетворює дані, відправлені у запитах (наприклад, POST-запити), у формат JSON, якщо це необхідно. Так само, він автоматично парсить JSON-відповіді, перетворюючи їх на JavaScript-об'єкти, що усуває необхідність ручного парсингу.

Інтерцептори запитів та відповідей (Interceptors) - це потужна функція, яка дозволяє перехоплювати запити або відповіді до того, як вони будуть оброблені then() або catch(). Інтерцептори можуть бути використані для:

Додавання токенів авторизації до кожного запиту.

Логування запитів/відповідей.

Обробки помилок глобально (автоматичний вихід при 401 Unauthorized).

Модифікації даних перед відправкою або після отримання.

Axios надає функціонал для відміни HTTP-запитів, що є корисним для оптимізації продуктивності (наприклад, скасування попередніх запитів при швидкому введенні даних у поле пошуку) або для запобігання витоку пам'яті. Можливість встановлювати таймаути для запитів, що дозволяє автоматично

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 40   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

скасувати запит, якщо відповідь не надійшла протягом заданого часу, запобігаючи "зависанню" додатка.

Axios має вбудовану підтримку для захисту від XSRF-атак, автоматично додаючи токени XSRF (Cross-Site Request Forgery) до запитів.

Клієнтська сторона:

- Забезпечує крос-браузерну сумісність.
- Перехоплює запити/відповіді.
- Трансформує дані запиту/відповіді.
- Скасовує запити.

Серверна сторона (Node.js):

- Працює з Node.js http модулем.
- Перехоплює запити/відповіді.
- Трансформує дані запиту/відповіді.
- Скасовує запити.
- Автоматично дотримується перенаправлень.

Axios є надійним, гнучким та функціональним HTTP-клієнтом, що широко використовується в сучасній веб-розробці. Його Promise-орієнтований API, можливість роботи як у браузері, так і в Node.js, а також такі потужні функції, як інтерцептори та автоматична трансформація даних, роблять його відмінним вибором для керування HTTP-запитами в JavaScript-додатках.

### 2.3.3. Управління даними

MongoDB та mLab: Як система управління базами даних (СУБД) обрано MongoDB. Це документоорієнтована NoSQL база даних, що дозволяє зберігати дані у вигляді гнучких JSON-подібних документів усередині колекцій, забезпечуючи високу масштабованість та гнучкість схеми даних. Для розгортання та управління базою даних MongoDB використовується mLab – хмарний сервіс (DBaaS), що надає MongoDB як послугу.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 41   |

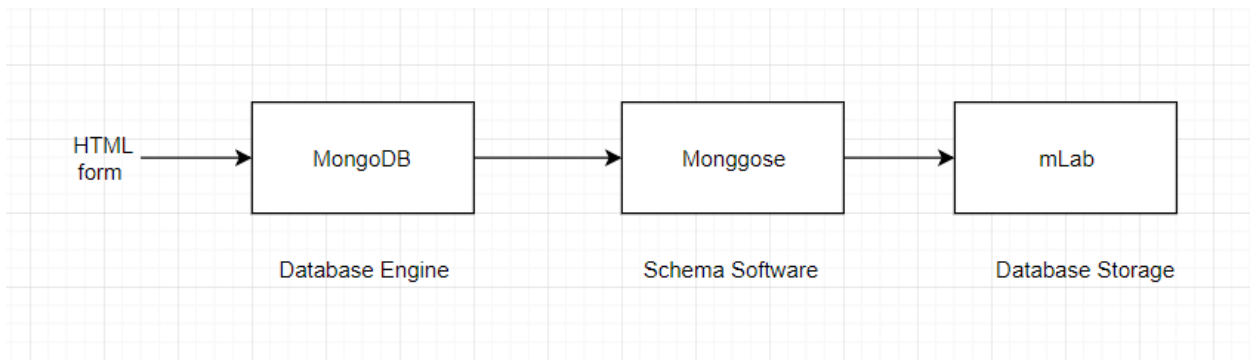


Рисунок 2.1 - Підключення MongoDB до Mlab за допомогою Mongoose для створення таблиць

mLab — це хмарний сервіс Database-as-a-Service (DBaaS), що надає повністю керовані розгортання MongoDB. Він дозволяє розробникам легко створювати, масштабувати та управляти базами даних MongoDB без необхідності турбуватися про інфраструктуру, хостинг, встановлення, оновлення чи обслуговування сервера.

Основна цінність mLab полягає в тому, що він абстрагує складність розгортання та адміністрування MongoDB. Розробникам не потрібно самостійно встановлювати та налаштовувати MongoDB на серверах, керувати реплікацією, шардингом чи резервним копіюванням. mLab бере на себе всі ці завдання, надаючи готові до використання екземпляри MongoDB. mLab працює на провідних хмарних провайдерах, таких як Amazon Web Services (AWS), Google Cloud Platform (GCP) та Microsoft Azure. Це дозволяє користувачам обирати регіон розміщення бази даних, оптимізуючи затримки та відповідність вимогам до даних.

Для забезпечення високої доступності та масштабованості у виробничих середовищах, mLab підтримує автоматичне налаштування реплікаційних наборів та sharding, що є важливими функціями MongoDB для роботи з великими обсягами даних та високими навантаженнями.

mLab надає ряд функцій безпеки, включаючи мережеву ізоляцію, аутентифікацію на рівні бази даних, SSL-шифрування для з'єднань, а також механізми аудиту та моніторингу.



При взаємодії користувача з формою входу, Passport.js обробляє надані облікові дані. Після успішної перевірки цих даних, система генерує криптографічний токен, що містить інформацію про користувача, і передає його для подальшої обробки. База даних, у свою чергу, верифікує інформацію, що міститься в токені. У разі успішної валідації користувач перенаправляється на відповідну авторизовану сторінку. Якщо ж інформація в токені не відповідає даним у базі даних, або токен є недійсним, користувач повертається на головний екран.

## 2.4. Реалізація діаграм варіантів використання системи нетворкінгу

Діаграма варіантів використання допомагає зрозуміти, що відбудеться при кожній події. За допомогою варіанту використання ми можемо дізнатися серію операцій. За допомогою варіанту використання розробник може створити огляд роботи проекту у вигляді діаграм.

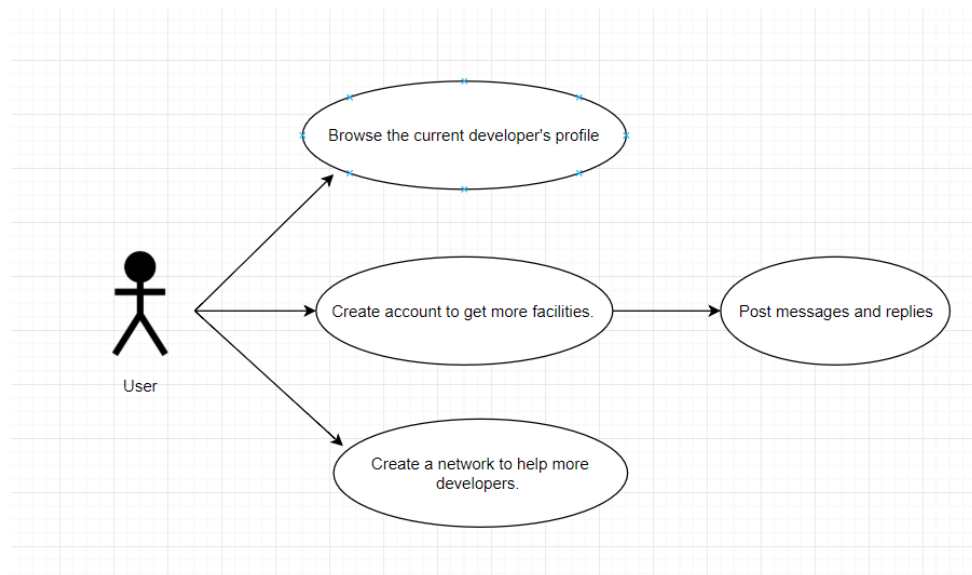


Рисунок 2.4 - Діаграма варіантів використання для нових користувачів

На цій діаграмі представлено:

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 44   |

## 1. Актор (Actor):

User - єдиний актор на діаграмі, який представляє будь-якого користувача, що взаємодіє з системою.

## 2. Варіанти Використання (Use Cases):

- Browse the current developer's profile - цей варіант використання описує можливість користувача переглядати профілі існуючих розробників у системі. З контексту можна припустити, що ця функціональність доступна навіть для неавтентифікованих користувачів, як згадувалося в описі системи.

- Create account to get more facilities - цей варіант використання відображає процес реєстрації нового користувача в системі. Зареєстровані користувачі отримують доступ до розширеного функціоналу.

- Post messages and replies - цей варіант використання доступний лише для зареєстрованих (автентифікованих) користувачів і дозволяє їм взаємодіяти у розділі публікацій, створюючи нові повідомлення та відповідаючи на існуючі. Зв'язок include або extend тут не вказаний, але логічно, що для "Post messages and replies" необхідно "Create account".

- Create a network to help more developers - цей варіант використання відображає ширшу мету системи – сприяння створенню професійної мережі серед розробників для взаємодопомоги. Це може включати функціональність портфоліо, менторства, або просто активну участь у спільноті, що дозволяє іншим знаходити та отримувати допомогу.

Ця діаграма варіантів використання наочно демонструє основні функціональні можливості системи, що доступні користувачеві. Вона показує, що користувачі можуть пасивно переглядати профілі, активно реєструватися для отримання більшого доступу до функцій, взаємодіяти шляхом публікації повідомлень, а також брати участь у побудові мережі для взаємодопомоги розробників.

Це є високорівневим уявленням про те, що система повинна робити, з точки зору зовнішнього користувача.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 45   |

## 2.5. Розробка діаграм потоку даних

Діаграма потоку даних представляє, як дані проходять через систему. Вона також надає детальний опис кожного компонента системи. Нижче наведено діаграми потоку даних для входу та реєстрації.

### 2.5.1. Діаграма потоку даних для входу та реєстрації

На рисунку 2.5 подано схему потоку системи (System Flowchart), яка відображає логіку взаємодії користувача з системою, включаючи процеси реєстрації, входу та основні функції після автентифікації.

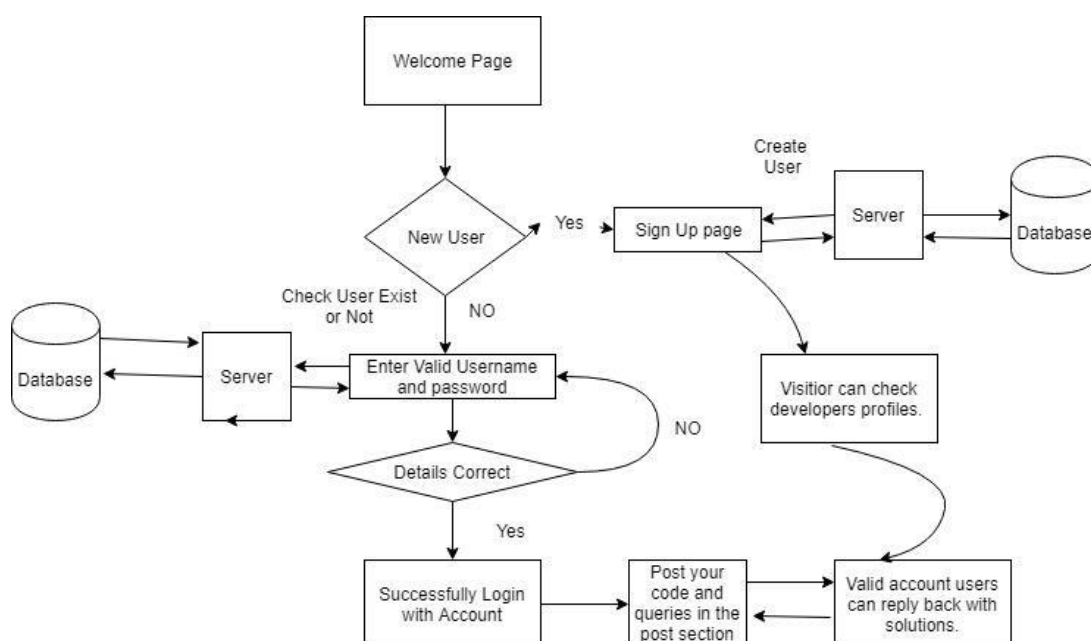


Рисунок 2.5 - Діаграма потоку даних для входу та реєстрації

Опишемо елементи та потоки на діаграмі.

1. Точки входу та початковий стан:

- Welcome Page (Привітальна сторінка): Це початкова точка входу в систему. Всі користувачі починають свою взаємодію з цієї сторінки.

2. Визначення статусу користувача:

- New User (Новий користувач) - Decision (Рішення): З "Welcome Page" система перевіряє, чи є користувач новим.

- Якщо "YES" (Так): Користувач переходить до процесу реєстрації.

- Sign Up page (Сторінка реєстрації): Користувач вводить свої дані для створення облікового запису.

- Create User (Створити користувача) - Процес: Ці дані відправляються на Server (Сервер) для обробки.

- Server <-> Database (Сервер <-> База даних): Сервер взаємодіє з Database (Базою даних) для збереження інформації про нового користувача. Після успішного створення користувача, потік повертається до "Enter Valid Username and password" для входу.

### 3. Процес Автентифікації (Вхід):

- Якщо "NO" для "New User": Користувач переходить безпосередньо до входу або якщо він щойно зареєструвався, то також переходить до входу.

- Enter Valid Username and password: Користувач вводить свої облікові дані.

- Server <-> Database: Введені дані відправляються на Server, який перевіряє їх на відповідність інформації в Database.

- Check User Exist or Not - Процес: Цей блок, розташований біля "Enter Valid Username and password", ілюструє процес перевірки існування облікового запису, що виконується сервером.

- Details Correct (Дані правильні) - Decision (Рішення): Сервер приймає рішення про валідність облікових даних.

- Якщо "NO": Користувач повертається до "Enter Valid Username and password" для повторного введення даних.

- Якщо "YES": Користувач успішно автентифікований.

### 4. Функціональність для автентифікованих користувачів:

- Successfully Login with Account: Після успішної автентифікації користувач отримує доступ до функціоналу системи.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 47   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

- Post your code and queries in the post section: Автентифіковані користувачі можуть створювати нові публікації (пости), включаючи код або запитання.

- Valid account users can reply back with solutions: Інші автентифіковані користувачі можуть відповідати на ці публікації, надаючи рішення або коментарі.

#### 5. Додаткова функціональність для неавтентифікованих користувачів:

Visitor can check developers profiles: цей потік, що відходить від "Enter Valid Username and password", вказує на те, що навіть без входу користувач має можливість переглядати профілі розробників. Це відповідає опису функціоналу "Browse the current developer's profile" з діаграми варіантів використання.

Діаграма ілюструє лінійний процес взаємодії з системою: починаючи з привітальної сторінки, користувач може або зареєструватися (з подальшим входом), або спробувати увійти. Якщо вхід успішний, він отримує доступ до функцій публікації та відповіді.

Також, відвідувачі можуть переглядати профілі розробників без автентифікації. Зв'язок "Server" та "Database" відображає взаємодію з базою даних на етапах реєстрації та входу.

#### 2.5.2. Потік даних для розділу публікацій

Діаграма (рис. 2.6) візуалізує деякі аспекти функціональності системи та її взаємодії з базою даних, а також розмежовує доступ до функцій для різних типів користувачів. Вона представляє схему функціональних блоків та взаємодії з даними, що ілюструє права доступу.

Опишемо її елементи:

1. Горизонтальні блоки, що представляють групи функціональності та права доступу:

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 48   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

- "User who already have account can access this functionality" (користувач, який вже має обліковий запис, може отримати доступ до цієї функціональності)

Цей блок об'єднує функції, доступні лише автентифікованим користувачам:

- Validate credentials: позначає процес входу в систему, де облікові дані користувача перевіряються.
- Edit profile: функціонал для зміни особистої інформації користувача, додавання досвіду, освіти тощо.
- Post feed: можливість створювати нові публікації (пости).

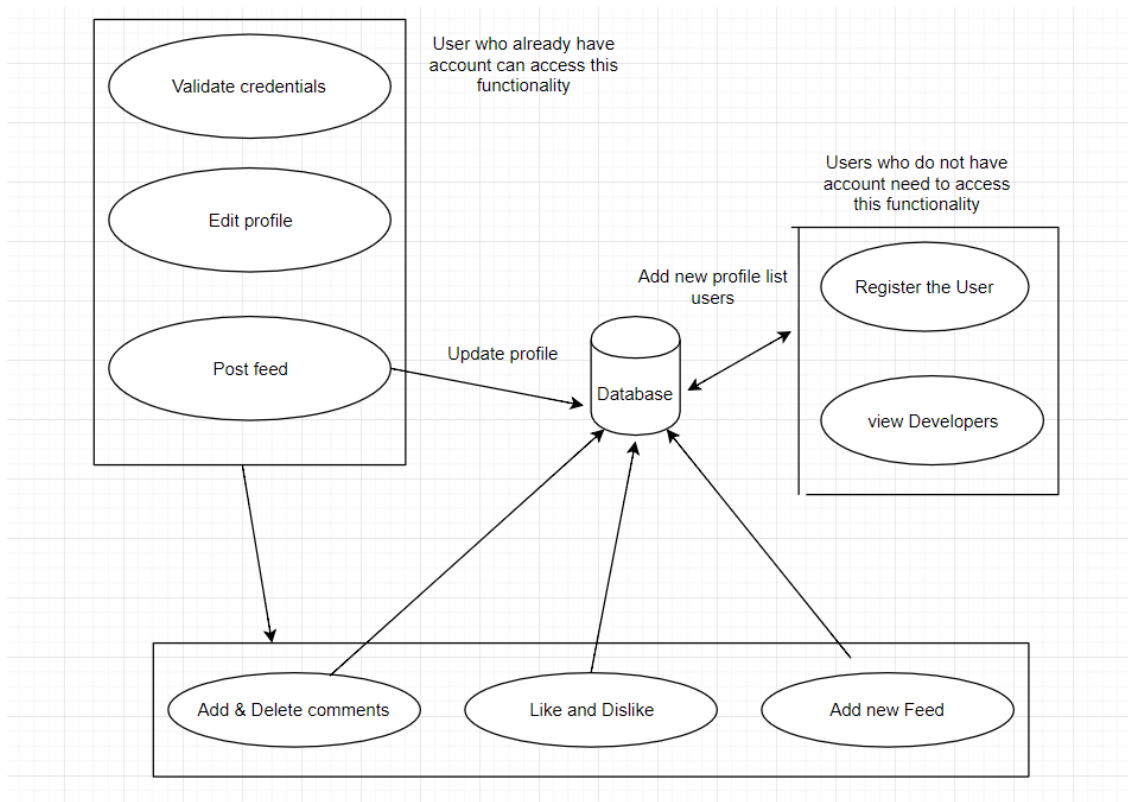


Рисунок 2.6 - Потік даних для розділу публікацій

- "Users who do not have account need to access this functionality" (користувачі, які не мають облікового запису, повинні отримати доступ до цієї функціональності)

Цей блок містить функції, доступні для неавтентифікованих користувачів (відвідувачів), або початкові функції для всіх:

- Register the User: процес створення нового облікового запису в системі.
- View Developers: Можливість переглядати профілі існуючих розробників без входу в систему.

Нижній горизонтальний блок (без заголовка, але функції, доступні для автентифікованих користувачів, пов'язані з Post feed):

Цей блок, схоже, деталізує функції, пов'язані з взаємодією навколо постів, і доступні після "Post feed" або для автентифікованих користувачів.

- Add & Delete comments: функціональність для коментування публікацій та керування власними коментарями.
- Like and Dislike: механізм оцінки контенту (коментарів, як зазначено в попередніх описах).
- Add new Feed: Хоча "Post feed" вже є у верхньому лівому блоці, "Add new Feed" означає повторне додавання нової публікації, що підкреслює циклічність процесу.

## 2. Центральний елемент: Database (База даних)

- Database: Центральне сховище інформації. Всі стрілки, що вказують на "Database" або виходять з неї, демонструють операції читання/запису даних.

- "Validate credentials" взаємодіє з базою даних для перевірки облікових даних.

- "Edit profile" взаємодіє з базою даних для оновлення інформації профілю ("Update profile").

- "Post feed" та "Add new Feed" взаємодіють з базою даних для збереження нових публікацій.

"Add new profile list users" – це є результатом реєстрації користувача, де дані нового профілю записуються в базу даних.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 50   |

- "Register the User" веде до оновлення бази даних.
- "View Developers" отримує дані з бази даних.
- "Add & Delete comments", "Like and Dislike" також взаємодіють з базою даних для зберігання та оновлення інформації про коментарі та оцінки.

Ця діаграма ефективно розмежовує функціональність системи на основі прав доступу користувачів (автентифіковані vs. неавтентифіковані). Вона також підкреслює роль бази даних як центрального сховища, з яким взаємодіють майже всі ключові операції. Діаграма дає чітке уявлення про те, які дії виконуються в системі і як вони пов'язані з зберіганням даних.

## 2.6. Розробка діаграми послідовності

На діаграмі послідовності взаємодії об'єктів розташовані в часовій послідовності. У цьому випадку об'єкти та класи, задіяні в сценарії, та послідовність повідомлень, обмінюваних між об'єктами, необхідні для реалізації функціональності сценарію. Діаграми послідовності іноді називають діаграмами подій або сценаріїв подій. Діаграма послідовності показує паралельні вертикальні лінії, різні процеси та об'єкти, які існують одночасно в цих лініях. На горизонтальних стрілках відображаються всі повідомлення, які обмінюються.

На цій діаграмі представлено чотири основні лінії життя (Lifelines), що представляють учасників взаємодії:

1. User: Актор, що ініціює дії.
2. Application: Представляє клієнтську частину веб-додатку (фронтенд, розроблений на React.js).
3. Server: Представляє бекенд-сервер додатку (на Node.js).
4. Database: Представляє сховище даних (MongoDB через mLab).

Діаграма розділена на дві основні логічні секції, що демонструють дві різні послідовності взаємодії: створення користувача та оновлення профілю.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 51   |

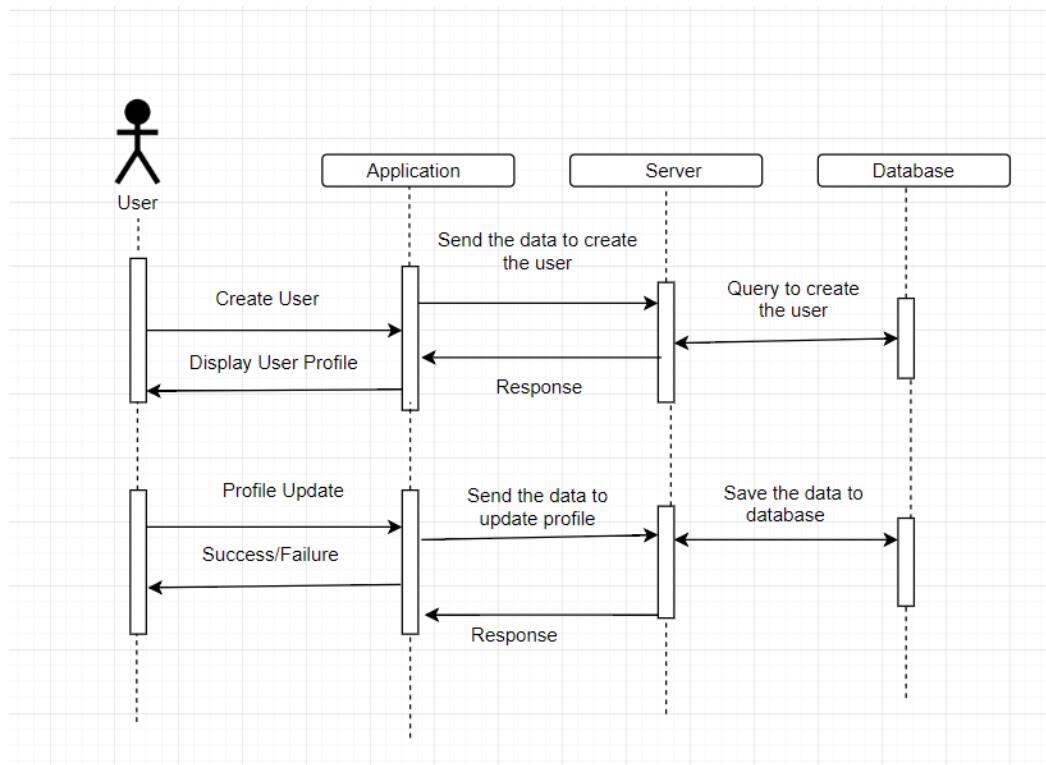


Рисунок 2.7 - Діаграма послідовності для оновлення профілю користувача

*Секція 1: Створення Користувача (Create User)*

1. User -> Application: Create User

- Користувач ініціює дію зі створення нового облікового запису (наприклад, заповнює та відправляє форму реєстрації).
- Повідомлення: "Create User".

2. Application -> Server: Send the data to create the user

- Додаток (фронтенд) отримує дані від користувача та відправляє їх на сервер для подальшої обробки.
- Повідомлення: "Send the data to create the user".

3. Server -> Database: Query to create the user

- Сервер отримує дані користувача і формує запит до бази даних для збереження інформації про нового користувача.
- Повідомлення: "Query to create the user".

#### 4. Database --> Server: Response

- База даних обробляє запит і відправляє відповідь назад на сервер, підтверджуючи або повідомляючи про статус операції створення користувача.

- Повідомлення: "Response".

#### 5. Server --> Application: Response

- Сервер отримує відповідь від бази даних і відправляє її назад додатку.

- Повідомлення: "Response".

#### 6. Application --> User: Display User Profile

- Додаток отримує відповідь від сервера та відображає профіль новоствореного користувача, підтверджуючи успішну реєстрацію.

- Повідомлення: "Display User Profile".

### *Секція 2: Оновлення Профілю (Profile Update)*

#### 1. User -> Application: Profile Update

- Користувач ініціює дію з оновлення свого профілю (наприклад, змінює дані та натискає "зберегти").

- Повідомлення: "Profile Update".

#### 2. Application -> Server: Send the data to update profile

- Додаток відправляє дані про оновлення профілю на сервер.

- Повідомлення: "Send the data to update profile".

#### 3. Server -> Database: Save the data to database

- Сервер отримує дані для оновлення профілю та відправляє їх до бази даних для збереження.

- Повідомлення: "Save the data to database".

#### 4. Database --> Server: Response

- База даних обробляє запит на оновлення та відправляє відповідь назад на сервер.

- Повідомлення: "Response".

#### 5. Server --> Application: Response

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 53   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

- Сервер отримує відповідь від бази даних і відправляє її додатку.
- Повідомлення: "Response".

#### 6. Application --> User: Success/Failure

- Додаток отримує відповідь від сервера та інформує користувача про успішне або неуспішне оновлення профілю.

- Повідомлення: "Success/Failure".

Розроблена діаграма послідовності чітко ілюструє динамічну поведінку системи при виконанні двох критично важливих операцій: створення нового користувача та оновлення існуючого профілю. Вона показує порядок викликів між фронтендом, бекендом та базою даних, а також обмін повідомленнями (даними та відповідями) між ними, що є ключовим для розуміння архітектури та логіки системи.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 54   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

## РОЗДІЛ 3. ПРОГРАМНА РОЗРОБКА РЕСУРСУ СОЦІАЛЬНОГО НЕТВОРКІНГУ ДЛЯ ВЗАЄМОДІЇ ІТ-РОЗРОБНИКІВ

### 3.1. Загальний вигляд та навігаційні елементи головної сторінки системи

Головна сторінка є початковою точкою взаємодії користувача з веб-додатком. Дизайн цієї сторінки передбачає її універсальний характер: вона доступна для перегляду всім користувачам, незалежно від їхнього статусу автентифікації (zareestrovani чи nezareestrovani).

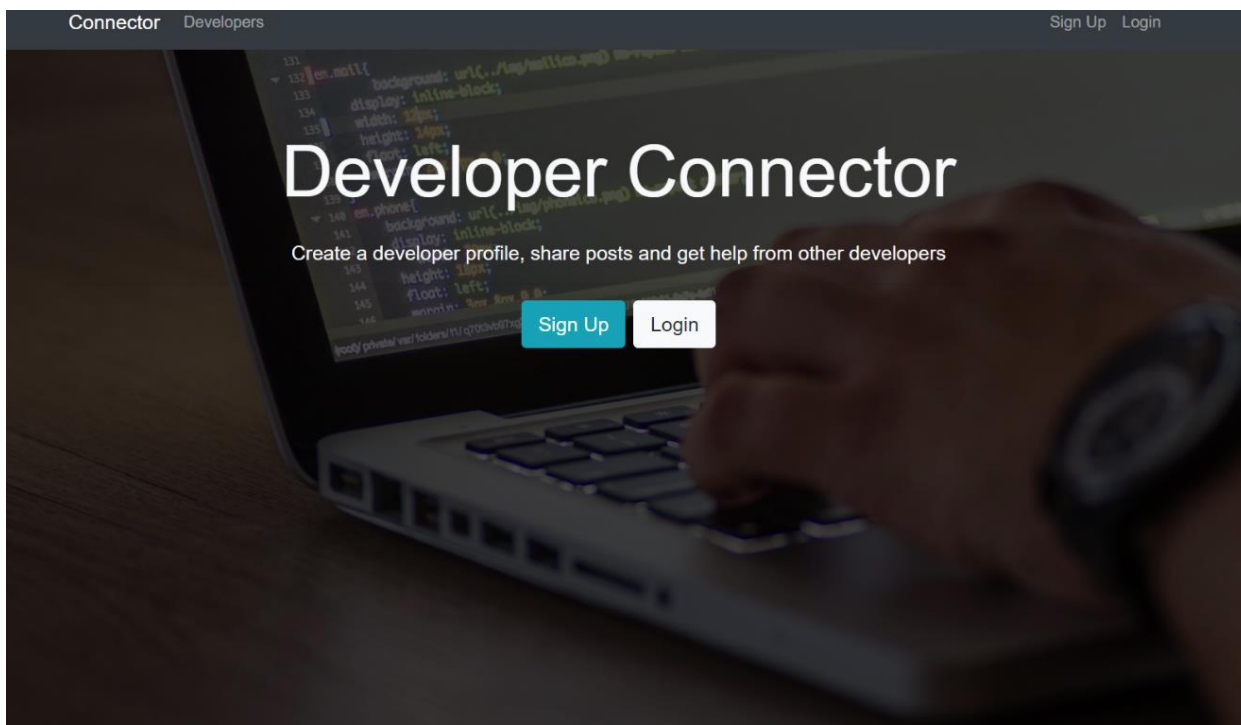


Рисунок 3.1 - Головна сторінка веб-додатку

Навігаційна структура головної сторінки організована наступним чином:

По середині розташовані ключові елементи керування доступом до системи:

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 55   |

- Кнопка "Зареєструватися" (Sign Up): Призначена для ініціації процесу створення нового облікового запису користувача.

- Кнопка "Увійти" (Login): Надає можливість автентифікації для вже зареєстрованих користувачів.

Зверху інтерфейс містить основні навігаційні елементи, що забезпечують перехід між ключовими розділами додатка:

- Кнопка "Connector": Забезпечує повернення до поточного головного екрану.

- Кнопка "Developers": При натисканні на цю кнопку система виконує запит до бази даних та відображає список усіх зареєстрованих розробників, які є користувачами платформи. Ця функція доступна для перегляду як зареєстрованим, так і незареєстрованим користувачам, сприяючи прозорості спільноти.

## **3.2. Функціональність перегляду профілів розробників**

### *3.2.1. Доступ до опцій розробників*

Опція "Розробники" є ключовим елементом інтерфейсу, що забезпечує доступ до загального списку учасників системи. Важливою особливістю цієї функції є її доступність для всіх категорій користувачів, включно з неавтентифікованими відвідувачами.

Дизайн сторінки лаконічний та інформативний. Він дозволяє користувачам швидко переглянути основні дані про розробників, їхні ролі та ключові навички без необхідності переходу на повний профіль кожного з них. Це забезпечує зручність навігації та пошуку потенційних колег або менторів за основними критеріями. Наявність фотографій та базової інформації сприяє першому контакту та прийняттю рішення про подальший перегляд детального профілю.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 56   |

# Developer Profiles

Browse and connect with developers

The screenshot displays two developer profiles in a list. The first profile is for 'San', a Junior Developer from India, with skills in React, Angular, HTML, and CSS. The second profile is for 'pr', a Senior Developer at List from India, with a skill in Java. Each profile includes a circular profile picture, a 'View Profile' button, and a 'Skill Set' section with a list of skills and checkmarks.

Рисунок 3.2 - Скриншот, який відображається, коли користувач натискає на опцію розробників

Таким чином, це єдина функціональна можливість у системі, що не вимагає попередньої реєстрації або входу. Після активації цієї опції користувачеві відображається перелік усіх розробників, які зареєстровані в "Соціальній мережі для розробників".

### 3.2.2. Деталізація профілю користувача

Після вибору конкретного розробника зі списку, система надає можливість перегляду його деталізованого профілю. Основним візуальним елементом цього профілю є фотографія користувача, яка слугує ідентифікатором та частиною персоналізації профілю. Повний профіль, як було зазначено раніше, також включатиме інформацію про статус, компанію, місце розташування, соціальні профілі, резюме, навички програмування, досвід роботи та освіти, що дозволяє отримати комплексне уявлення про професійні компетенції розробника.

[Back To Profiles](#)



### San's Bio

I am a student of Computer Science. I am looking to pursue my career in the field of programming and development.

Рисунок 3.3 - Відображення фотографії профілю користувача

## 3.3. Процеси автентифікації та реєстрації користувачів

### 3.3.1. Екран входу до системи

При натисканні користувачем на опцію "Вхід" (Login), розташовану у правому верхньому куті головної сторінки, система перенаправляє його на спеціалізовану веб-сторінку для автентифікації.

Рисунок 3.4 - Відображення екрану входу

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 58   |

Ця сторінка призначена виключно для користувачів, які вже мають зареєстрований обліковий запис у системі "Соціальна мережа для розробників". Користувачі повинні ввести свої облікові дані (наприклад, ім'я користувача та пароль) для отримання доступу до функціоналу, що вимагає автентифікації.

### 3.3.2. Екран реєстрації нових користувачів

Для нових користувачів, які бажають приєднатися до платформи, передбачена опція "Зареєструватися" (Sign Up).

The screenshot shows a web interface for signing up. At the top, there is a dark navigation bar with 'Connector Developers' on the left and 'Sign Up Login' on the right. The main content area has a heading 'Sign Up' and a sub-heading 'Create your DevConnector account'. Below these are four input fields: 'Name', 'Email', 'Password', and 'Confirm Password'. A small note below the Email field says 'This site uses Gravatar so if you want a profile image, use a Gravatar email'. At the bottom is a teal 'Submit' button.

Рисунок 3.5 - Екран реєстрації для нових користувачів

Активація цієї опції ініціює процес реєстрації, який складається з кількох етапів:

#### 1. Початкова форма реєстрації.

Користувачеві пропонується заповнити первинну форму, яка містить обов'язкові поля для створення базового облікового запису (наприклад, електронна пошта, бажане ім'я користувача та пароль).

#### 3. Форма деталізації профілю.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 59   |

Після успішного заповнення початкової форми реєстрації, користувач буде перенаправлений на другу форму. Ця форма призначена для збору більш детальної інформації, що включає:

- Особиста інформація: Дані, що стосуються особистості користувача (наприклад, ім'я, місцезнаходження).

- Професійна інформація: Деталі, що характеризують професійний досвід та навички розробника (наприклад, поточна посада, компанія, список навичок програмування, досвід роботи, освіта, посилання на соціальні профілі).

Вся введена інформація буде збережена в базі даних системи та використана для формування профілю користувача, який буде доступний для перегляду іншим учасникам мережі. Цей багатоетапний підхід забезпечує збір необхідних даних для побудови повноцінного та інформативного профілю розробника.

### **3.4. Деталізація профілю: розробка форм додавання досвіду та освіти**

Після успішного завершення початкового етапу реєстрації, користувачеві надається можливість деталізувати свій професійний профіль шляхом заповнення спеціалізованих форм. Цей етап має на меті збір структурованої інформації про досвід роботи та освіти розробника. Надання цих даних є критично важливим для формування повноцінного та репрезентативного профілю, що сприятиме ефективній взаємодії у соціальній мережі та може впливати на майбутні можливості професійного розвитку (наприклад, стажування або працевлаштування).

Користувачі мають можливість додавати кілька записів у кожній з цих категорій, що дозволяє відобразити весь спектр їхніх досягнень та професійного шляху.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 60   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

# Add Experience

Add any job or position that you have had in the past or current

\* = required fields

\* Company

\* Job Title

Location

From Date

mm/dd/yyyy

To Date

mm/dd/yyyy

Current Job

Job Description

Tell us about the the position

Submit

Рисунок 3.6 - Форма додавання досвіду

Форма містить кілька полів для введення даних, деякі з яких позначені астериском (\*) як обов'язкові поля (required fields). Це означає, що користувач не зможе відправити форму без заповнення цих полів.

Поля форми включають:

- Company (Компанія): Текстове поле для введення назви компанії, де користувач працював або працює. Це обов'язкове поле.
- Job Title (Посада): Текстове поле для введення назви посади, яку займав або займає користувач. Це також обов'язкове поле.
- Location (Місцезнаходження): Текстове поле для зазначення географічного місцезнаходження компанії або роботи. Це поле не є обов'язковим.
- From Date (Дата початку): Поле для введення дати початку роботи у форматі місяць/день/рік (mm/dd/yyyy).

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 61   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

- To Date (Дата закінчення): Поле для введення дати закінчення роботи у форматі місяць/день/рік (mm/dd/yyyy).

- Current Job (Поточна робота): Прапорець (checkbox), який користувач може відмітити, якщо ця позиція є його поточною роботою. Якщо цей прапорець активований, поле "To Date" стає необов'язковим або деактивується.

- Job Description (Опис роботи): Багаторядкове текстове поле (textarea) для введення детального опису обов'язків, досягнень та проектів на даній посаді. Супровідний текст "Tell us about the the position" (Розкажіть нам про цю посаду) пояснює призначення цього поля.

Внизу форми розташована кнопка "Submit" (Відправити), яка призначена для збереження введених даних про досвід роботи до профілю користувача.

**Add Education**

Add any school, bootcamp, etc that you have attended

\* = required fields

\* School

\* Degree or Certification

\* Field of Study

From Date

mm/dd/yyyy

To Date

mm/dd/yyyy

Current Job

Program Description

Tell us about the program that you were in

Submit

Рисунок 3.7 - Форма додавання освіти

|      |      |          |        |      |                         |            |
|------|------|----------|--------|------|-------------------------|------------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк.<br>62 |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |            |

Форма додавання освіти є схожою до попередньої і включає наступні поля:

- School (Навчальний заклад): Текстове поле для введення назви навчального закладу (школи, університету, коледжу, буткемпу тощо), який користувач відвідував. Це обов'язкове поле.

- Degree or Certification (Ступінь або Сертифікація): Текстове поле для зазначення отриманого ступеня (наприклад, бакалавр, магістр) або сертифікації (наприклад, сертифікат курсу). Це також обов'язкове поле.

- Field of Study (Галузь навчання): Текстове поле для введення спеціалізації або галузі навчання (наприклад, Комп'ютерні науки, Програмна інженерія). Це обов'язкове поле.

- From Date (Дата початку): Поле для введення дати початку навчання у форматі місяць/день/рік (mm/dd/yyyy).

- To Date (Дата закінчення): Поле для введення дати закінчення навчання у форматі місяць/день/рік (mm/dd/yyyy).

- Program Description (Опис програми): Багаторядкове текстове поле (textarea) для введення детального опису навчальної програми, ключових дисциплін або досягнень під час навчання. Супровідний текст "Tell us about the program that you were in" пояснює призначення цього поля.

### **3.5. Функціональність панелі управління та розділу публікацій**

#### *3.5.1. Панель управління профілем користувача*

Для зареєстрованих користувачів передбачена панель управління (Dashboard), яка є централізованим інтерфейсом для керування власним профілем та розміщеною інформацією. У цьому розділі користувач має можливість переглядати всі дані, які він раніше вніс до системи (наприклад, професійний досвід, освіта, навички). Крім функції перегляду, панель

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 63   |

управління надає можливість редагування та видалення будь-якої раніше доданої інформації, забезпечуючи гнучкість у керуванні вмістом профілю.

## Dashboard

Welcome [San](#)

[Edit Profile](#) [Add Experience](#) [Add Education](#)

### Experience Credentials

| Company            | Title                       | Years                  |                        |
|--------------------|-----------------------------|------------------------|------------------------|
| Reliance           | Technical Support Executive | 2013/10/31 -2014/04/20 | <a href="#">Delete</a> |
| E-centric solution | Database Administrator      | 2015/02/28 -2015/05/31 | <a href="#">Delete</a> |

### Education Credentials

| School     | Degree                  | Years                  |                        |
|------------|-------------------------|------------------------|------------------------|
| University | Bachelor of Engineering | 2009/08/31 -2013/05/31 | <a href="#">Delete</a> |
| University | Master of Science       | 2023/09/23 - Now       | <a href="#">Delete</a> |

[Delete My Account](#)

Рисунок 3.8 – Панель управління облікового запису

Додатково, панель управління включає функціонал для повного видалення облікового запису. Користувач може ініціювати цей процес шляхом натискання на кнопку "Видалити мій обліковий запис" (Delete My Account), що призведе до перманентного видалення всіх його даних із системи.

### 3.5.2. Розділ публікацій (стрічка публікацій)

Після успішної автентифікації, користувач отримує доступ до розділу "Стрічка публікацій" (Post Feed).

Цей розділ є основним майданчиком для взаємодії у спільноті розробників. Його ключові функціональні можливості включають:

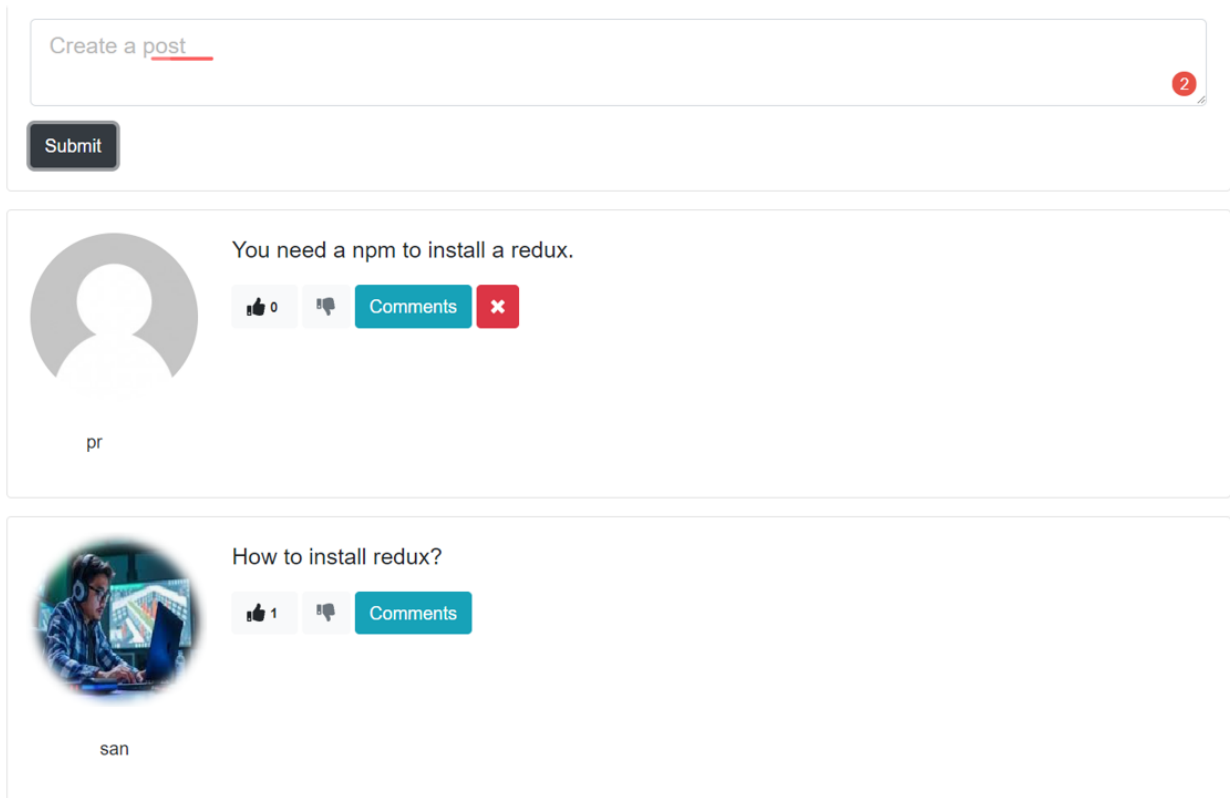


Рисунок 3.9 - Ілюстрація розділу публікацій

**Створення публікацій:** Автентифіковані користувачі мають можливість публікувати власні питання, запити або інші релевантні матеріали для спільноти.

**Коментування:** Інші користувачі можуть відповідати на розміщені публікації, надаючи рішення, коментарі або додаткову інформацію.

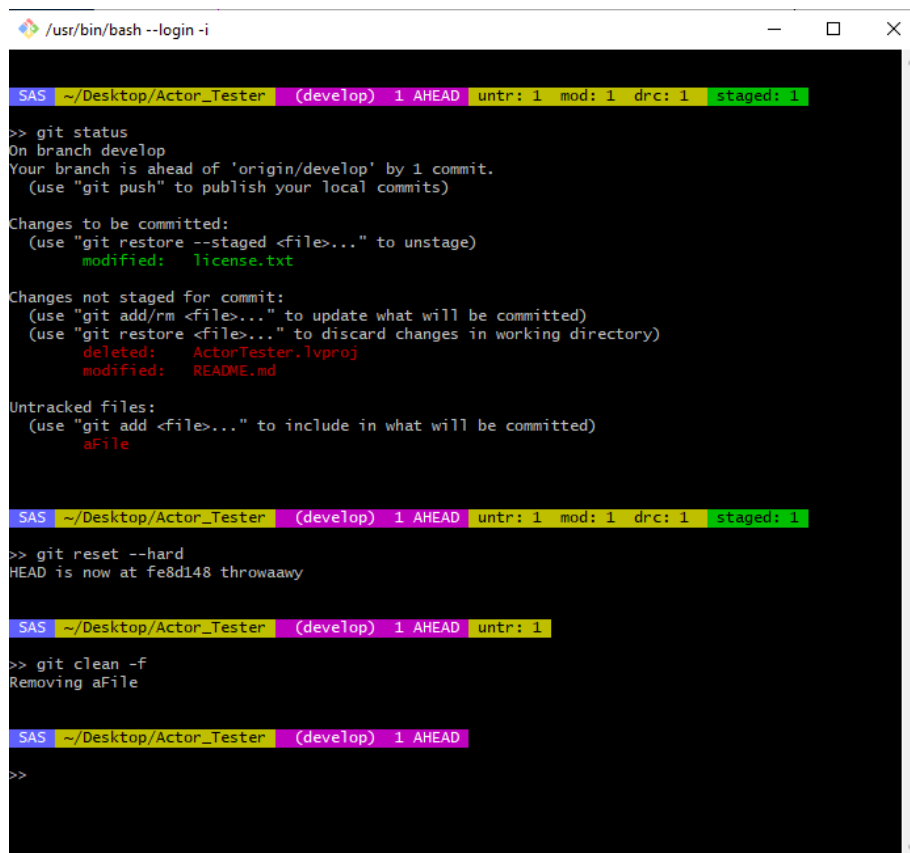
**Система оцінки контенту:** Для кожної публікації та коментаря передбачено опції "подобається" (like) та "не подобається" (dislike). Цей механізм дозволяє користувачам колективно оцінювати релевантність та якість наданих відповідей, сприяючи виявленню найбільш корисних отриманих рішень.

**Управління власними публікаціями:** Користувачі мають право видаляти власні публікації з розділу стрічки, зберігаючи контроль над контентом, який вони розмістили.

### 3.6. Тестування системи соціального нетворкінгу для ІТ-розробників

#### 3.6.1. Документація з тестування програмного забезпечення

Тестування програмного забезпечення є критично важливим етапом у життєвому циклі розробки системи, спрямованим на верифікацію та валідацію функціональності та продуктивності програмного забезпечення. Метою тестування є ідентифікація дефектів (помилки) та забезпечення відповідності системи встановленим вимогам. Процес тестування може охоплювати як ізольовані компоненти системи, так і інтегровані її частини або систему в цілому. У даному проекті для виконання певних тестових операцій використовувалося програмне середовище Git Bash. У контексті тестування програмного забезпечення існують численні підходи та методи, проте у цьому проекті застосовувалася вибіркова методологія тестування.



```

/usr/bin/bash --login -i

SAS ~/Desktop/Actor_Tester (develop) 1 AHEAD untr: 1 mod: 1 drc: 1 staged: 1
>> git status
On branch develop
Your branch is ahead of 'origin/develop' by 1 commit.
(use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   license.txt

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    ActorTester.lvproj
        modified:   README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        aFile

SAS ~/Desktop/Actor_Tester (develop) 1 AHEAD untr: 1 mod: 1 drc: 1 staged: 1
>> git reset --hard
HEAD is now at fe8d148 throwaawy

SAS ~/Desktop/Actor_Tester (develop) 1 AHEAD untr: 1
>> git clean -f
Removing aFile

SAS ~/Desktop/Actor_Tester (develop) 1 AHEAD
>>
```

Рисунок 3.10 - Середовище Git Bash

### 3.6.2. Модульне тестування (Unit Testing)

Модульне тестування є методом тестування програмного забезпечення, при якому індивідуальні компоненти або модулі вихідного коду тестуються ізольовано для визначення їх коректності. Цей підхід є фундаментальним у щоденній практиці розробників програмного забезпечення. Хоча існують різні інструменти автоматизації для верифікації функціональності модулів відповідно до заданих сценаріїв, модульне тестування не було включено до методології тестування даного проекту. Це означає, що перевірка функціональності окремих, ізольованих компонентів не проводилась.

### 3.6.3. Тестування на прийнятність користувачами (User Acceptance Testing, UAT)

Тестування на прийнятність користувачами — це етап тестування, на якому система в цілому, або її значні інтегровані частини, тестуються для перевірки відповідності бізнес-вимогам та задоволення потреб кінцевих користувачів. Цей вид тестування проводиться перед передачею системи замовнику або перед її розгортанням у промислове середовище. Метою UAT є підтвердження того, що система функціонує згідно з очікуваннями та вимогами, викладеними у специфікації. Тестування на прийнятність користувачами є загальноприйнятою та обов'язковою практикою в більшості організацій, оскільки воно відіграє ключову роль у забезпеченні успішного розгортання програмних продуктів.

### 3.6.4. Результати тестування

Тестування даного проекту здійснювалося відповідно до зазначених вище методологій. В процесі розробки програмного забезпечення були проведені ітеративні перевірки функціональності додатка з метою виявлення та усунення потенційних дефектів. За результатами проведеного тестування критичних проблем або невідповідностей у функціональності додатка

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 67   |

виявлено не було. Це свідчить про стабільну роботу системи відповідно до її специфікації на момент завершення тестування.

### **3.7. Напрямки майбутнього розвитку проекту нетворкінгу для ІТ-розробників**

Подальший розвиток системи нетворкінгу для ІТ-розробників передбачає імплементацію нових функціональних можливостей та вдосконалення існуючих з метою підвищення безпеки, розширення освітнього потенціалу та покращення кар'єрних можливостей для користувачів.

#### **1. Покращення механізмів управління паролями**

З метою посилення безпеки облікових записів користувачів та мінімізації ризиків несанкціонованого доступу, пропонується запровадити розширені політики керування паролями:

- Політика закінчення терміну дії пароля: Впровадження механізму примусової зміни пароля через певний, заздалегідь визначений період часу (наприклад, кожні 90 днів) для підвищення ротації облікових даних.

- Заборона повторного використання старих паролів: Реалізація функціоналу, що запобігатиме використанню користувачами раніше застосованих паролів з метою ускладнення підбору комбінацій.

- Вимоги до надійності пароля: Запровадження обов'язкових критеріїв "надійності" пароля, що включатимуть вимоги до мінімальної довжини, використання великих та малих літер, цифр та спеціальних символів, для забезпечення стійкості до атак методом перебору.

#### **2. Інтеграція функціоналу онлайн-навчання**

Для посилення освітньої складової платформи та сприяння обміну експертними знаннями пропонується додати функцію онлайн-навчання. Ця функція дозволить досвідченим розробникам (менторам) проводити

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 68   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

інтерактивні навчальні сесії, вебінари або індивідуальні консультації для менш досвідчених користувачів. Реалізація такого функціоналу суттєво підвищить цінність проекту як освітнього ресурсу, сприяючи прискореному професійному розвитку учасників.

### 3. Розширення можливостей пошуку вакансій

З метою підтримки кар'єрного зростання користувачів, пропонується інтеграція спеціалізованої функції для дослідження можливостей працевлаштування. Цей розділ дозволить амбітним розробникам переглядати актуальні вакансії та стажування у сфері ІТ, отримувати релевантні рекомендації та, можливо, безпосередньо відгукуватися на пропозиції. Таке розширення функціоналу перетворить платформу на комплексний ресурс для професійного розвитку, що охоплює не лише навчання та менторство, а й сприяє працевлаштуванню.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                         | 69   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

## ВИСНОВКИ

У ході виконання дипломної роботи було реалізовано повний цикл проєктування, моделювання та створення ресурсу соціального нетворкінгу, орієнтованого на взаємодію ІТ-розробників. Розробка даного ресурсу є відповіддю на зростаючий попит на спеціалізовані платформи, що поєднують функціональність професійного спілкування, обміну досвідом, менторства та участі в спільних проєктах. Високий рівень динаміки ІТ-сфери потребує інструментів для швидкої комунікації, професійного розвитку та створення ефективних спільнот.

У першому розділі було проведено детальний аналіз предметної області. Було обґрунтовано передумови створення ресурсу соціального нетворкінгу, визначено його місце серед сучасних інформаційних сервісів та окреслено функціональні й архітектурні вимоги. Проведено глибокий аналіз існуючих платформ (Codementor, Chegg, MentorCruise, Stack Overflow), що дозволило виділити ключові недоліки, такі як вузька спеціалізація, відсутність комплексної взаємодії та обмежена підтримка побудови тривалих професійних зв'язків. Це стало підґрунтям для формування концептуально нової платформи, що дозволяє ІТ-фахівцям ефективно комунікувати, навчатися та співпрацювати.

У другому розділі було здійснено моделювання структури платформи з урахуванням вимог до сучасних веб-ресурсів. Розроблено архітектуру системи, що включає модулі автентифікації, управління користувачами, публікаціями та профілями. Проведено специфікацію апаратних і програмних вимог до серверної інфраструктури, а також сформовано вимоги до клієнтської частини. Особливу увагу приділено вибору інструментів та технологій реалізації, зокрема використанню сучасних фреймворків для фронтенду (наприклад, React), бекенду (Node.js, Express) та системи управління базами даних (MongoDB). Побудовані діаграми варіантів

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 70   |

використання, потоку даних і діаграма послідовності відображають логіку взаємодії користувача з ресурсом на всіх ключових етапах.

У третьому розділі представлено реалізацію програмної частини платформи. Створено функціональні компоненти, такі як головна сторінка з навігаційними елементами, екрани входу й реєстрації, профіль користувача з можливістю додавання інформації про освіту та досвід, а також розділ публікацій, що виконує роль стрічки активності. Реалізовано функціонал персональної панелі керування для кожного користувача. Проведено модульне тестування, тестування на прийнятність користувачами (UAT), а також підготовлено документацію з тестування, що підтверджує стабільність і коректність роботи системи. За результатами тестування було виявлено та усунуено низку незначних недоліків.

Окрему увагу було приділено перспективам розвитку проєкту. У роботі запропоновано напрями подальшої еволюції ресурсу, серед яких — інтеграція функціоналу відеозв'язку, розширення можливостей менторства, додавання алгоритмів рекомендацій для формування професійних контактів, автоматизований підбір команд для спільної участі в проєктах тощо.

Таким чином, у дипломній роботі було не лише реалізовано ресурс соціального нетворкінгу для ІТ-розробників, а й обґрунтовано його актуальність, конкурентоспроможність і технічну життєздатність. Отримані результати демонструють потенціал подальшого вдосконалення платформи як ефективного інструменту для професійної комунікації, обміну знаннями та розвитку ІТ-спільнот.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 71   |

## ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Codementor Review - The Digital Merchant - <https://thedigitalmerchant.com/codementor-review/>
2. An overview of the general working procedure of Codementor | Download Scientific Diagram - [https://www.researchgate.net/figure/An-overview-of-the-general-working-procedure-of-Codementor\\_fig1\\_330071750](https://www.researchgate.net/figure/An-overview-of-the-general-working-procedure-of-Codementor_fig1_330071750)
3. Mentor-spotting: recommending expert mentors to mentees for live troubleshooting in Codementor | Knowledge and Information Systems - <https://link.springer.com/article/10.1007/s10115-018-1298-3>
4. MentorCruise Features, Pricing, and Alternatives | AI Tools - <https://aitools.inc/tools/mentorcruise>
5. Chegg Review | PCMag - [https://www.pcmag.com/reviews/chegg?test\\_uuid=02LIF0iWksilxYTJVF8uH5y&test\\_variant=A](https://www.pcmag.com/reviews/chegg?test_uuid=02LIF0iWksilxYTJVF8uH5y&test_variant=A)
6. Dabbish, L. A., & Kraut, R. E. (2006, November). Commonalities and differences in the design of social networks for software development. In Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work (pp. 581-590). ACM.
7. Vasilescu, B., van Deursen, A., & Serebrenik, A. (2014, May). Perceived career benefits of developer social media. In Proceedings of the 36th International Conference on Software Engineering (pp. 110-119). ACM.
8. Storey, M. A., & Rigby, P. C. (2016). The social aspects of software development. In Software Engineering (pp. 407-429). Springer, Berlin, Heidelberg.
9. Kraut, R. E., & Resnick, P. (Eds.). (2012). Building successful online communities: Evidence-based social design. MIT press.
10. Wang, X., Li, B., & Zhang, P. (2017). Online social network for software developers: An empirical study on Stack Overflow. Journal of Systems and Software, 133, 166-177.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | БР.ІІІ – 44.00.00.000 ІІЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 72   |

11. Gehreke, C., Schilling, J., & Kauffeld, S. (2024). Digital Tools in Peer Mentoring: Enhancing Relationships and Student Engagement in Higher Education. *Higher Education Research & Development*.
12. Mullen, J., & Hall, L. (2024). Exploring Digital Mentorship's Role in Leadership Training. *International Journal of Digital Learning*.
13. Setiani, E., et al. (2024). E-Mentoring as an Alternative to Traditional Teacher Development: Addressing Logistical Challenges. *Journal of Educational Technology & Society*.
- Jayathissa, D., et al. (2024). Investigating the Impact of Mentoring Software on Professional Development in Programming. *Computers & Education*.
14. Huang, S., et al. (2024). Technology-Enabled Teacher Development: Improving Mentorship Accessibility Through Digital Platforms. *Journal of Teacher Education*.
15. Mong, K. V., Thein, N. L., & Win, S. S. (2019, June). Developing a web application using MERN stack. In *2019 International Conference on Advanced Information Technologies (ICAIT)* (pp. 384-389). IEEE.
16. Sharma, N., & Singh, R. (2021). A Study on the MERN Stack in Web Application Development. *International Journal of Computer Applications*, 180(36), 1-5.
17. Singh, S., & Singh, N. (2020, December). MERN Stack: A Comprehensive Review on its Components and Applications. In *2020 7th International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 770-775). IEEE.
18. Choudhary, A., & Goyal, S. (2021). Building Scalable Web Applications with MERN Stack: A Review. *International Journal of Recent Trends in Engineering & Research (IJRTER)*, 7(4), 11-15.
19. Prasad, K. S., & Kumar, R. P. (2022). MERN Stack for Full-Stack Development: A Comprehensive Study. *International Journal of Advanced Trends in Computer Science and Engineering*, 11(2), 1-5.

|      |      |          |        |      |                           |            |
|------|------|----------|--------|------|---------------------------|------------|
|      |      |          |        |      | БР.ІІІ – 44.00.00.000 ІІЗ | Арк.<br>73 |
| Змн. | Арк. | № докум. | Підпис | Дата |                           |            |

- 20.IEEE. (2008). IEEE Standard for Software Test Documentation. IEEE Std 829-2008.
- 21.Beizer, B. (1990). Software testing techniques (2nd ed.). Van Nostrand Reinhold.
- 22.Ammann, P. E., & Offutt, J. (2016). Introduction to software testing. Cambridge University Press.
- 23.Binder, R. V. (1999). Testing object-oriented systems: Models, patterns, and tools. Addison-Wesley Professional.
- 24.Pandit, P., & Tahiliani, S. (2015). AgileUAT: A Framework for User Acceptance Testing based on User Stories and Acceptance Criteria. International Journal of Computer Applications, 120(10), 16-21.
- 25.Pressman, R. S. (2010). Software engineering: A practitioner's approach (7th ed.). McGraw-Hill. (Класичне джерело, яке часто посилається на різні методології тестування).
- 26.Sommerville, I. (2016). Software engineering (10th ed.). Pearson Education. (Ще одне класичне джерело з широким охопленням тестування).
- 27.Black, R. (2009). Managing the testing process: Practical tools and techniques for managing and controlling the testing process (3rd ed.). Wiley
- 28.NIST. (2017). NIST Special Publication 800-63B: Digital Identity Guidelines, Authentication and Lifecycle Management. National Institute of Standards and Technology. (Ключовий стандарт для політик паролів).
- 29.Sasse, M. A., & Whitten, A. (2010). The psychology of passwords. IEEE Security & Privacy, 8(3), 26-33.
- 30.Furnell, S. M., & Clarke, N. L. (2005). The psychological dimensions of password security. Computers & Security, 24(5), 351-360.
- 31.Sasse, M. A., & Whitten, A. (2000). The human factor in password security. The SANS Institute.

|      |      |          |        |      |                          |      |
|------|------|----------|--------|------|--------------------------|------|
|      |      |          |        |      | БР.ІІІ – 44.00.00.000 ПЗ | Арк. |
|      |      |          |        |      |                          | 74   |
| Змн. | Арк. | № докум. | Підпис | Дата |                          |      |

32. Dealing with Coding Challenges Through Digital Platforms: Assessing Their Effectiveness in Skill Development. (2025). CLEI Electronic Journal, 28(1).

33. Improving Programming Skills: The Impact of Interactive E-Learning Tools on Student Success. (2025). CLEI Electronic Journal, 28(1).

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | БР.ІП – 44.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 75   |

## **ДОДАТКИ**

## Додаток А

### Програмні коди

#### А.1. Код для розділу входу

```
import React, { Component } from 'react';
import PropTypes from 'prop-types';
import { connect } from 'react-redux';
import { loginUser } from '../../actions/authActions';
import TextFieldGroup from '../../common/TextFieldGroup';

class Login extends Component {
  constructor() {
    super();
    this.state = {
      email: '',
      password: '',
      errors: {}
    };
    this.onChange = this.onChange.bind(this);
    this.onSubmit = this.onSubmit.bind(this);
  }

  componentDidMount() {
    if (this.props.auth.isAuthenticated) {
      this.props.history.push('/dashboard');
    }
  }

  componentWillReceiveProps(nextProps) {
    if (nextProps.auth.isAuthenticated) {
      this.props.history.push('/dashboard');
    }
    if (nextProps.errors) {
      this.setState({ errors: nextProps.errors });
    }
  }

  onSubmit(e) {
    e.preventDefault();
    const userData = {
      email: this.state.email,
      password: this.state.password
    };
    this.props.loginUser(userData);
  }

  onChange(e) {
    this.setState({ [e.target.name]: e.target.value });
  }

  render() {
    const { errors } = this.state;
    return (
      <div className="login">
        <div className="container">
          <div className="row">
            <div className="col-md-8 m-auto">
              <h1 className="display-4 text-center">Увійти</h1>
            </div>
          </div>
        </div>
      </div>
    );
  }
}
```

```

        <p className="lead text-center">
            Увійдіть у свій обліковий запис DevConnector
        </p>
        <form onSubmit={this.onSubmit}>
            <TextFieldGroup
                placeholder="Електронна адреса"
                name="email"
                type="email"
                value={this.state.email}
                onChange={this.onChange}
                error={errors.email}
            />
            <TextFieldGroup
                placeholder="Пароль"
                name="password"
                type="password"
                value={this.state.password}
                onChange={this.onChange}
                error={errors.password}
            />
            <input type="submit" className="btn btn-info btn-block mt-4" />
        </form>
    </div>
</div>
</div>
</div>
    );
}
}
Login.propTypes = {
    loginUser: PropTypes.func.isRequired,
    auth: PropTypes.object.isRequired,
    errors: PropTypes.object.isRequired
};

const mapStateToProps = state => ({
    auth: state.auth,
    errors: state.errors
});

export default connect(mapStateToProps, { loginUser })(Login);

```

```

import React, { Component } from 'react';
import PropTypes from 'prop-types';
import { withRouter } from 'react-router-dom';
import { connect } from 'react-redux';
import { registerUser } from '../../actions/authActions';
import TextFieldGroup from '../common/TextFieldGroup';

class Register extends Component {
    constructor() {
        super();
        this.state = {
            name: '',
            email: '',
            password: '',
            password2: '',
            errors: {}
        };
        this.onChange = this.onChange.bind(this);
        this.onSubmit = this.onSubmit.bind(this);
    }

    componentDidMount() {
        if (this.props.auth.isAuthenticated) {
            this.props.history.push('/dashboard');
        }
    }
}

```

```

componentWillReceiveProps(nextProps) {
  if (nextProps.errors) {
    this.setState({ errors: nextProps.errors });
  }
}

onChange(e) {
  this.setState({ [e.target.name]: e.target.value });
}

onSubmit(e) {
  e.preventDefault();
  const newUser = {
    name: this.state.name,
    email: this.state.email,
    password: this.state.password,
    password2: this.state.password2
  };
  this.props.registerUser(newUser, this.props.history);
}

render() {
  const { errors } = this.state;
  return (
    <div className="register">
      <div className="container">
        <div className="row">
          <div className="col-md-8 m-auto">
            <h1 className="display-4 text-center">Зареєструватися</h1>
            <p className="lead text-center">
              Створіть свій обліковий запис DevConnector
            </p>
            <form noValidate onSubmit={this.onSubmit}>
              <TextFieldGroup
                placeholder="Ім'я"
                name="name"
                value={this.state.name}
                onChange={this.onChange}
                error={errors.name}
              />
              <TextFieldGroup
                placeholder="Електронна пошта"
                name="email"
                type="email"
                value={this.state.email}
                onChange={this.onChange}
                error={errors.email}
                info="Цей сайт використовує Gravatar, тому якщо ви хочете м
              />
              <TextFieldGroup
                placeholder="Пароль"
                name="password"
                type="password"
                value={this.state.password}
                onChange={this.onChange}
                error={errors.password}
              />
              <TextFieldGroup
                placeholder="Підтвердіть пароль"
                name="password2"
                type="password"

```

```

        value={this.state.password2}
        onChange={this.onChange}
        error={errors.password2}
      />
      <input type="submit" className="btn btn-info btn-block mt-4" />
    </form>
  </div>
</div>
</div>
</div>
  </div>
);
}
}

Register.propTypes = {
  registerUser: PropTypes.func.isRequired,
  auth: PropTypes.object.isRequired,
  errors: PropTypes.object.isRequired
};

const mapStateToProps = state => ({
  auth: state.auth,
  errors: state.errors
});

export default connect(mapStateToProps, { registerUser })(withRouter(Register));

```

```

import React, { Component } from 'react';
import PropTypes from 'prop-types';
import { connect } from 'react-redux';
import TextAreaFieldGroup from '../common/TextAreaFieldGroup';
import { addComment } from '../actions/postActions';

class CommentForm extends Component {
  constructor(props) {
    super(props);
    this.state = {
      text: '',
      errors: {}
    };
    this.onChange = this.onChange.bind(this);
    this.onSubmit = this.onSubmit.bind(this);
  }

  componentWillReceiveProps(newProps) {
    if (newProps.errors) {
      this.setState({ errors: newProps.errors });
    }
  }

  onSubmit(e) {
    e.preventDefault();
    const { user } = this.props.auth;
    const { postId } = this.props;
    const newComment = {
      text: this.state.text,
      name: user.name,
      avatar: user.avatar
    };
  }
}

```

```

import React, { Component } from 'react';
import PropTypes from 'prop-types';
import { connect } from 'react-redux';
import TextAreaFieldGroup from '../common/TextAreaFieldGroup';
import { addComment } from '../../actions/postActions';

class CommentForm extends Component {
  constructor(props) {
    super(props);
    this.state = {
      text: '',
      errors: {}
    };
    this.onChange = this.onChange.bind(this);
    this.onSubmit = this.onSubmit.bind(this);
  }

  componentWillReceiveProps(newProps) {
    if (newProps.errors) {
      this.setState({ errors: newProps.errors });
    }
  }

  onSubmit(e) {
    e.preventDefault();
    const { user } = this.props.auth;
    const newComment = {
      text: this.state.text,
      name: user.name,
      avatar: user.avatar
    };
    this.props.addComment(postId, newComment);
    this.setState({ text: '' });
  }

  onChange(e) {
    this.setState({ [e.target.name]: e.target.value });
  }

  render() {
    const { errors } = this.state;
    return (
      <div className="post-form mb-3">
        <div className="card card-info">
          <div className="card-header bg-info text-white">
            Залишити коментар...
          </div>
          <div className="card-body">
            <form onSubmit={this.onSubmit}>
              <div className="form-group">
                <TextAreaFieldGroup
                  placeholder="Відповісти на публікацію"
                  name="text"
                  value={this.state.text}
                  onChange={this.onChange}
                  error={errors.text}
                />
              </div>
            </form>
          </div>
        </div>
      </div>
    );
  }
}

```

## БІБЛІОГРАФІЧНА ДОВІДКА

**Тема дипломної роботи:** “ Реалізація ресурсу соціального нетворкінгу для взаємодії ІТ-розробників ”

Обсяг пояснювальної записки: 75 аркушів.

Дата закінчення роботи: 9 червня 2025 р.

Підпис студента \_\_\_\_\_