

# ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

на тему:

## «Платформа ІТ-академії на основі фреймворку Spring Boot»

Студента 4 курсу, 6 групи,  
спеціальності 121 «Інженерія  
програмного забезпечення»  
освітньої програми «Інженерія  
програмного забезпечення»

підпис студента

Скрипника  
Дмитра  
Олександровича

Науковий керівник PhD,  
доцент кафедри інженерії  
програмного забезпечення та  
кібербезпеки

підпис керівника

Десятко Альона  
Миколаївна

Гарант освітньої програми  
кандидат технічних наук,  
доцент кафедри інженерії  
програмного забезпечення та  
кібербезпеки

підпис гаранта

Рзаєва Світлана  
Леонідівна

# Державний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

## Затверджую

Зав. кафедри інженерії програмного  
забезпечення та кібербезпеки

Криворучко О. В.

«14» листопада 2022 р.

## Завдання

### на випускний кваліфікаційний проєкт студентів

Скрипнику Дмитру Олександровичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Платформа ІТ-академії на основі фреймворку Spring Boot»

Затверджена наказом ректора від «6» грудня 2022 р. № 3288

2. Строк здачі студентом закінченого проєкту 5 червня 2023

3. Цільова установка та вихідні дані до проєкту

Мета проєкту - покращити та здобути нові знання у процесі розробки платформи ІТ академії на основі фреймворку Spring Boot, яка забезпечить сучасний та ефективний спосіб вивчення мови програмування.

Об'єкт дослідження - є платформи ІТ академії на основі фреймворку Spring Boot.

Предмет дослідження - аналіз, розробка та впровадження платформи ІТ академії, яка базується на сучасних технологіях.

4. Консультанти проєкту із зазначенням розділів, які консультують:

| Розділ | Консультант<br>(прізвище, ініціали) | Підпис, дата   |                  |
|--------|-------------------------------------|----------------|------------------|
|        |                                     | Завдання видав | Завдання прийняв |
|        |                                     |                |                  |
|        |                                     |                |                  |

5. Зміст випускного кваліфікаційного проєкту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ТЕХНІЧНОГО ЗАВДАННЯ

1.1. Огляд ринку інформаційних технологій та платформ навчання

1.2. Загальні теоретичні відомості проблем розробки сучасності і способи їх вирішення

1.3. Технічне завдання

1.4. Висновок до розділу 1

РОЗДІЛ 2. ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Вибір середовища розробки, мови програмування та відповідного інструментарію

2.2. Архітектура розроблюваного програмного забезпечення

2.3. Розробка графічного інтерфейсу

2.4. Висновок до розділу 2

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Розробка Back-End частини додатку

3.2. Розробка Front-End частини додатку

3.3. Процес тестування застосунку

3.4. Висновок до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

## 6. Календарний план виконання проєкту

| № пор. | Назва етапів випускного кваліфікаційного проєкту                              | Строк виконання етапів проєкту |            |
|--------|---|--------------------------------|------------|
|        |   | за планом                      | фактично   |
| 1      | 2   | 3                              | 4          |
| 1.     | <i>Вибір теми випускного кваліфікаційного проєкту</i>                         | 21.09.2022                     | 21.09.2022 |
| 2.     | <i>Розробка та затвердження завдання на проєкт</i>                            | 14.11.2022                     | 14.11.2022 |
| 3.     | <i>Вступ та перелік літературних джерел</i>                                   | 23.12.2022                     | 23.12.2022 |
| 4.     | <i>Розділ 1. Аналіз предметної області та постановка технічного завдання</i>  | 27.01.2023                     | 27.01.2023 |
| 5.     | <i>Розділ 2. Проектування програмного забезпечення</i>                        | 03.03.2023                     | 03.03.2023 |
| 6.     | <i>Розділ 3. Процес розробки програмного забезпечення</i>                     | 14.04.2023                     | 14.04.2023 |
| 7.     | <i>Висновки</i>   | 28.04.2023                     | 28.04.2023 |
| 8.     | <i>Здача випускного кваліфікаційного проєкту на кафедрі (перша перевірка)</i> | 17.05.2023                     | 17.05.2023 |
| 9.     | <i>Підготовка автореферату та презентації доповіді</i>                        | 26.05.2023                     | 26.05.2023 |
| 10.    | <i>Попередній захист випускного кваліфікаційного проєкту</i>                  | 29.05.2023 –<br>02.06.2023     |            |
| 11.    | <i>Зовнішнє рецензування випускного кваліфікаційного проєкту</i>              | 05.06.2023                     | 05.06.2023 |
| 12.    | <i>Здача прошого випускного кваліфікаційного проєкту на кафедрі</i>           | 05.06.2023                     | 05.06.2023 |
| 13.    | <i>Публічний захист випускного кваліфікаційного проєкту</i>                   | 16.06.2023                     | 16.06.2023 |

7. Дата видачі завдання «14» листопада 2022 р.

8. Науковий керівник випускного кваліфікаційного проєкту \_\_\_\_\_

Десятко А.М

(прізвище, ініціали, підпис)

9. Гарант освітньої програми \_\_\_\_\_

Рзаєва С.Л.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент \_\_\_\_\_

Скрипник Д.О.

(прізвище, ініціали, підпис)

**11. Відгук керівника випускного кваліфікаційного проєкту**

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Науковий керівник випускного кваліфікаційного проєкту

*(підпис, дата)*

Відмітка про попередній захист

*(ПІБ, підпис, дата)*

**12. Висновок про випускний кваліфікаційний проєкт**

Випускний кваліфікаційний проєкт студента

Скрипник Д. О.

*(прізвище, ініціали)*

може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми

Рзасва С.Л.

*(прізвище, ініціали, підпис)*

Завідувач кафедри

Криворучко О. В.

*(підпис, прізвище, ініціали)*

## АНОТАЦІЯ

Відповідно до мети дослідження робота присвячена розробці та впровадженню платформи ІТ академії, яка базується на сучасних технологіях. Основним завданням є створити такий продукт, який надасть можливість отримувати сучасні знання, за допомогою яких користувач стає конкурентно спроможним працівником на ринку.

В результаті порівняльного аналізу аналогічних рішень визначено теоретичні сучасні аспекти, пов'язані з програмним забезпеченням платформи академії, а також визначити проблеми та переваги роботи схожих платформ.

Розробка серверної частини виконана у середовищі IntelliJ, оскільки, як мінімум інтерфейс обраного середовища має достатню кількість інструментів для розробки програмного забезпечення. Також особливістю використання саме цього середовища, дає можливість користуватись інтерфейсв системи контролю версій Git.

Готовий програмний комплекс платформи ІТ академії було успішно протестовано та доступне будь - якому користувачу.

**Ключові слова:** фреймворк, інтерфейс, Java, Spring, Spring Boot, об'єктно-орієнтоване програмування, MVC, UML, база даних, SQL.

## ABSTRACT

In accordance with the purpose of the research, the work is devoted to the development and implementation of the IT platform of the academy, which is based on modern technologies. The main task is to create such a product, which will provide an opportunity to obtain modern knowledge, with the help of which the user becomes a competitive employee in the market.

As a result of a comparative analysis of similar solutions, the theoretical modern aspects related to the software of the academy platform were determined, as

well as the problems and advantages of the operation of similar platforms were determined.

The development of the server part was carried out in the IntelliJ environment, since at least the interface of the chosen environment has a sufficient number of tools for software development. Also, a feature of using this particular environment is that it makes it possible to use the interface of the Git version control system.

The finished software complex of the IT Academy platform has been successfully tested and is available to any user.

**Keywords:** framework, interface, Java, Spring, Spring Boot, object-oriented programming, MVC, UML, database, SQL



## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення

БД – бази даних

СУБД – система управління базами даних

ІТ – інформаційні технології

Back-End – частина коду додатку, що відповідає за логіка програми

Front-End – частина коду додатку, що відповідає за візуальну частину

JDK - Java Development Kit

JRE - Java Runtime Environment

JVM - Java Virtual Machine

ТЗ – технічне завдання

|                                  |              |                 |   |             |
|----------------------------------|--------------|-----------------|---|-------------|
| <i>ДТЕУ 121 06-23.БР</i>         |              |                 |   |             |
| <i>Зм.</i>                       | <i>Аркуш</i> | <i>№ докум.</i> | <i>Підпис</i>   | <i>Дата</i> |
| Зав. каф.                        |              | Криворучко О.В. |   | 14.04.23    |
| Керівник                         |              | Десятко А. М.   |   | 14.04.23    |
| Гарант                           |              | Рзаєва С.Л.     |   | 14.04.23    |
| Розробив                         |              | Скрипник Д. О.  |   | 14.04.23    |
| <i>Перелік умовних скорочень</i> |              |                 |   |             |
|                                  |              |                 | <i>Платформа IT-академії на основі фреймворку Spring Boot</i> |             |
|                                  |              |                 | <i>Факультет інформаційних технологій</i>                     |             |
|                                  |              |                 | <i>4 курс, 6 група</i>  |             |
|                                  |              |                 | <i>ПС</i>   |             |
|                                  |              |                 | <i>3</i>  |             |
|                                  |              |                 | <i>49</i>   |             |



## ЗМІСТ

|  |           |
|--|-----------|
| <b>ВСТУП.....</b>  | <b>5</b>  |
| <b>РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ТЕХНІЧНОГО ЗАВДАННЯ.....</b>           | <b>7</b>  |
| 1.1. Огляд ринку інформаційних технологій та платформ навчання.....                        | 7         |
| 1.2 Загальні теоретичні відомості проблем розробки сучасності і способи їх вирішення ..... | 10        |
| 1.2 Опис предметної області та мета створення системи.....                                 | 15        |
| 1.3 Технічне завдання .....  | 16        |
| 1.4 Висновки до розділу 1 .....  | 18        |
| <b>РОЗДІЛ 2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....</b>                                | <b>20</b> |
| 2.1 Вибір середовища розробки, мови програмування та відповідного інструментарію           | 20        |
| 2.2 Архітектура розроблюваного програмного забезпечення .....                              | 24        |
| 2.3 Розробка графічного інтерфейсу .....   | 30        |
| 2.4 Висновки до розділу 2 .....  | 35        |
| <b>РОЗДІЛ 3 ПРОЦЕС РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b>                              | <b>36</b> |
| 3.1 Розробка Back-End частини .....  | 36        |
| 3.2 Розробка Front-End частини.....  | 40        |
| 3.3 Процес тестування застосунку.....  | 41        |
| 3.4 Висновки до розділу 3 .....  | 44        |
| <b>ВИСНОВКИ ТА ПРОПОЗИЦІЇ .....</b>  | <b>46</b> |
| <b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>   | <b>48</b> |
| <b>ДОДАТКИ.....</b>  | <b>50</b> |

|            |              |                 |               |             |  |   |              |                |
|------------|--------------|-----------------|---------------|-------------|--|---|--------------|----------------|
|            |              |                 |               |             | <i>ДТЕУ 121 06-23.БР</i>                               |   |              |                |
| <i>Зм.</i> | <i>Аркуш</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |  | <i>Стадія</i>   | <i>Аркуш</i> | <i>Аркушів</i> |
| Зав. каф.  |              | Криворучко О.В. |               | 23.12.22    | Платформа IT-академії на основі фреймворку Spring Boot | 3   | 4            | 49             |
| Керівник   |              | Десятко А. М    |               | 23.12.22    |  | <i>Факультет інформаційних технологій<br/>4 курс, 6 група</i> |              |                |
| Гарант     |              | Рзаєва С.Л.     |               | 23.12.22    |  |   |              |                |
| Розробив   |              | Скрипник Д. О.  |               | 23.12.22    |  |   |              |                |
|            |              |                 |               |             | <i>Зміст</i>   |   |              |                |

## ВСТУП

У сучасному світі сфера ІТ має надзвичайний попит та розвивається з неймовірною швидкістю. Конкуренція на ринку програмістів завжди збільшується, а технології постійно змінюються та оновлюються, тому потрібно забезпечити користувачів платформою за допомогою якої вони зможуть почати свої перші кроки у програмуванні. В той час більш досвідчені спеціалісти матимуть змогу повторити вивченні ними технології і дізнатися нових.

*Актуальність.* Полягає безпосередньо у розробці платформи ІТ академії, яка дозволить отримати якісні та актуальні на даний момент знання на ринку інформаційних технологій, що допоможе в майбутньому стати конкурентоспроможним фахівцем та отримати першу роботу в ІТ. Також в додаток , увесь курс на платформі побудований на вивченні мови програмування Java, яка має надзвичайний попит та станом на сьогодні займає 2 місце серед усіх інших мов у комерційному використанні.

*Мета дослідження* є покращити та здобути нові знання у процесі розробки платформи ІТ академії на основі фреймворку Spring Boot, яка забезпечить сучасний та ефективний спосіб вивчення мови програмування.

*Завдання* дослідження полягають у наступному:

- аналізувати предметну область ринку інформаційних технологій;
- опрацювати актуальність технологій;
- виявити та аналізувати проблеми інших платформ;
- зробити аналіз, що необхідний роботодавцю задля наповнення курсу необхідною базою знань;

| Зм.       | Аркуш | № докум.        | Підпис | Дата     | ДТЕУ 121 06-23.БР                                      |                                    |       |         |
|-----------|-------|-----------------|--------|----------|--|------------------------------------|-------|---------|
| Зав. каф. |       | Криворучко О.В. |        | 23.12.22 | Платформа ІТ-академії на основі фреймворку Spring Boot | Стадія                             | Аркуш | Аркушів |
| Керівник  |       | Десятко А. М.   |        | 23.12.22 |  | В                                  | 5     | 49      |
| Гарант    |       | Рзаєва С.Л.     |        | 23.12.22 |  | Факультет інформаційних технологій |       |         |
| Розробив  |       | Скрипник Д. О.  |        | 23.12.22 |  | 4 курс, 6 група                    |       |         |
|           |       |                 |        |          | Вступ  |                                    |       |         |

- розробити веб застосунок - платформу ІТ академії мовою програмування Java з використанням фреймворку Spring Boot.

*Об'єктом дослідження є платформи ІТ академії на основі фреймворку Spring Boot.*

*Предметом дослідження аналіз, розробка та впровадження платформи ІТ академії, яка базується на сучасних технологіях.*

*Методи дослідження: аналіз літературних джерел, вивчення даних з практики роботи схожих платформ, розслідування потреб роботодавців, аналіз технологій та мов програмування.*

*Наукова новизна полягає у розробці платформи ІТ академії, яка забезпечить якісними та безкоштовними знаннями опираючись на аналіз ринку технологій та проблеми інших комерційних платформ.*

*Практичне значення дослідження полягає у тому, що використання платформи ІТ академії дозволить забезпечити найсучасніші та конкурентноспроможні знання на ринку працівників в ІТ сфері. Кожна людина матиме можливість здобути знання вищої категорії в короткий термін та отримати роботу мрії.*

*Для досягнення мети дослідження, запропоновано розглянути теоретичні аспекти, пов'язані з програмним забезпеченням платформи академії, а також визначити особливості, конкурентноспроможність, прибутковість та проблеми роботи схожих платформ на ринку.*

|     |       |         |        |      |  |                   |       |
|-----|-------|---------|--------|------|--|-------------------|-------|
|     |       |         |        |      |  | ДТЕУ 121 06-23.БР | Аркуш |
|     |       |         |        |      |  |                   | 6     |
| Зм. | Аркуш | № докум | Підпис | Дата |  |                   |       |

## РОЗДІЛ 1

# АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ТЕХНІЧНОГО ЗАВДАННЯ

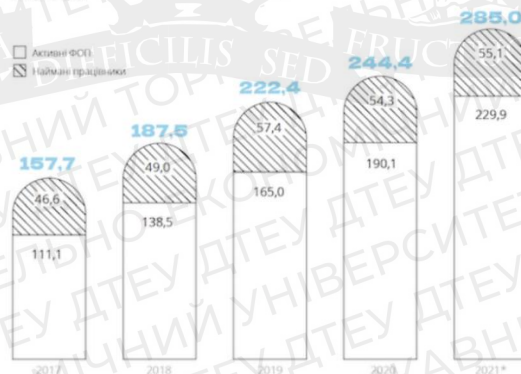
### 1.1. Огляд ринку інформаційних технологій та платформ навчання

Огляд ринку інформаційних технологій в Україні, а також аналіз онлайн ІТ академій, є важливою частиною дослідження, оскільки дозволяє визначити конкурентне середовище, допомогти побудувати технічне завдання та зрозуміти тенденції у цій сфері. Врахування цих аспектів дозволить розробити та впровадити платформу, яка задовольнятиме потреби користувачів та буде конкурентоспроможною на ринку.

Для початку потрібно проаналізувати ринок ІТ індустрії України в цілому, що допоможе розкрити основні тренди, потенціал та виклики, з якими зіштовхнулась ця галузь. Якщо брати статистичні данні за 2021 рік (за для демонстрації ринку без впливу війни) то ця галузь росла на 36% лише за рік, що значно збільшило кількість робочих місць та попит на розробників. [1]

Чисельність ІТ-талентів в Україні,

тис. осіб



Наймані працівники за даними Держстату. Активні ФОП – ті, що задекларували дохід у поточному році, дані ДПС

Рисунок 1.1. Графік зросту кількості ІТ спеціалістів

Джерело: побудовано на основі джерела [1]

| Зм.       | Аркуш | № докум.        | Підпис | Дата     | ДТЕУ 121 06-23.БР   |                                    |       |         |
|-----------|-------|-----------------|--------|----------|---|------------------------------------|-------|---------|
| Зав. каф. |       | Криворучко О.В. |        | 27.01.23 | Платформа ІТ-академії на основі фреймворку Spring Boot      | Стадія                             | Аркуш | Аркушів |
| Керівник  |       | Десятко А. М    |        | 27.01.23 |   | РІ                                 | 7     | 49      |
| Гарант    |       | Рзаєва С.Л.     |        | 27.01.23 | Аналіз предметної області та постановка технічного завдання | Факультет інформаційних технологій |       |         |
| Розробив  |       | Скрипник Д. О.  |        | 27.01.23 |   | 4 курс, 6 група                    |       |         |

Водночас ІТ-індустрія – це також єдина експортна галузь України, яка не споживає природних ресурсів та не вимагає великих капіталовкладень. Тому потрібно орієнтуватись не лише на данні по збільшенні кількості працівників, а й на об'єми експорту, так як українські розробники користуються великим попитом і за кордоном. Найбільшими експортними ринками для ІТ компаній з центрами розробки в Україні є США, Велика Британія, та Мальта.[2]

### Експорт комп'ютерних послуг з України, млн дол. США

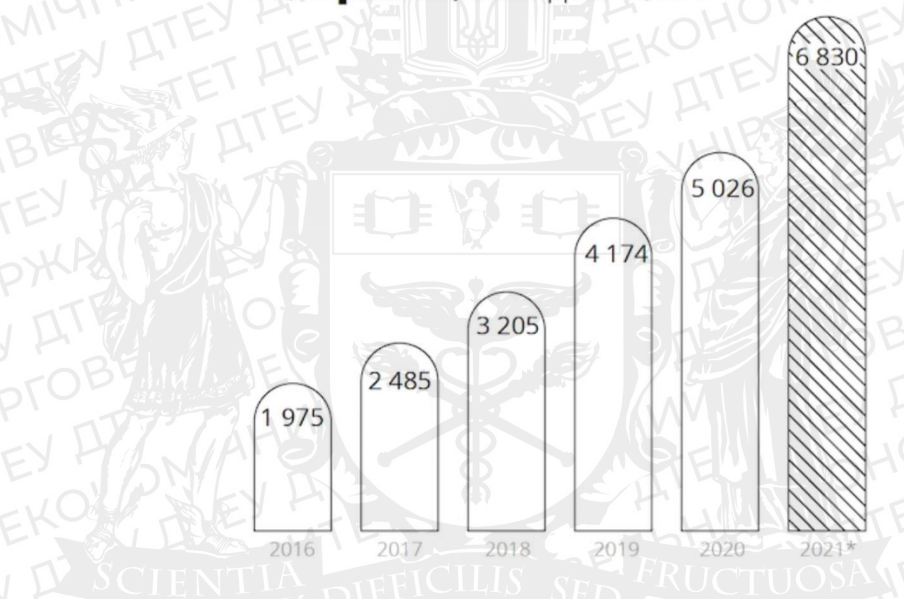


Рисунок 1.2. Графік зросту експорту ІТ послуг

Джерело: побудовано на основі джерела [1]

Для проведення аналізу ринку онлайн ІТ академій можна виокремити деякі аспекти за якими буде проводиться огляд:

- тенденції та інновації: Визначення актуальних тенденцій та інновацій у сфері;
- цінова політика: Аналіз цінової політики конкурентів та порівняння вартості навчання;

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      | ДТЕУ 121 06-23.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата |                   | 8     |

- цільова аудиторія: Вивчення цільової аудиторії різних ІТ академій.

У зв'язку з популяризацією та збільшенням попиту на розробників з'явилося чимало різних онлайн ІТ академій, що допоможе нам зробити роботу над помилками та створити власну платформу з урахуванням усіх плюсів та мінусів. На першому місці в ІТ індустрії лежать технології, які постійно змінюються і це один із головних критеріїв при виборі програми для навчання на платформі і в нашому випадку одне із кращих рішень це Java. За статистичними даними Java посідає друге місце серед інших мов у комерційному використанні, перше за сферою використання Back-end розробки, четверте місце в опенсорс-проектах та постійно підтримується та оновлюється.

Цінова політика є дуже важливою складовою при виборі курсів, адже якщо брати до уваги, що всі існуючі платформи не дають можливості отримати знання повністю безкоштовно. У залежності від академії є різні варіанти компенсації отриманих знань це або переплата за навчання, або підписання певних договорів, що зобов'язують студента після закінчення віддавати певний відсоток від заробітної плати чи пропрацювати не безкоштовній основі на компанію, що надала послугу. У нашому випадку платформа є повністю безкоштовною та не потребує ніяких вкладень, що закладає перспективу на великий попит платформи.

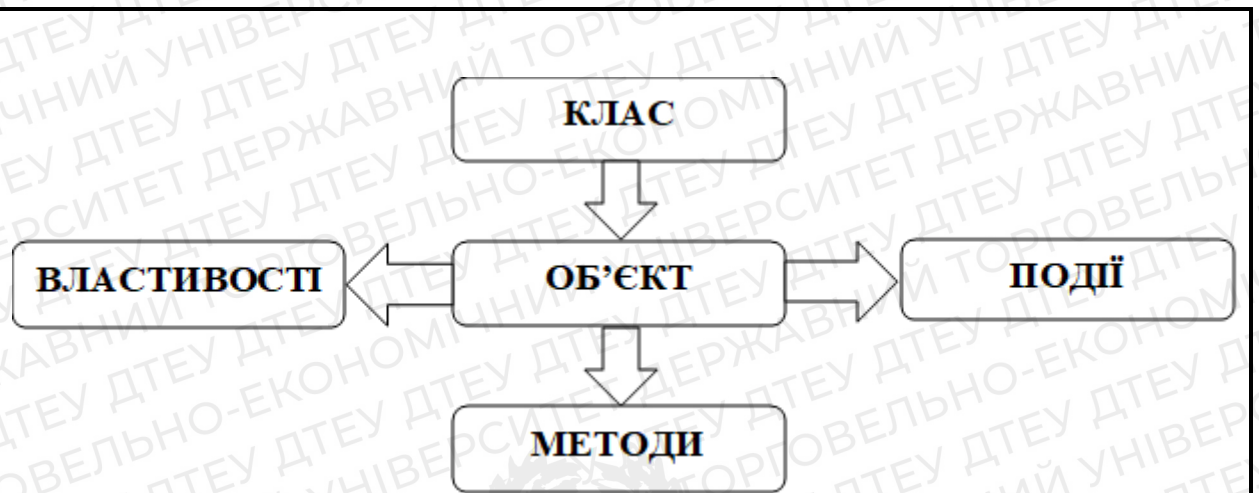
Важливо розуміти, що у ІТ сфері не має «віку» і кожен має можливість почати свій шлях розробника тому розроблена платформа підлягає під всю можливу цільову аудиторію так як побудова не з дотриманням user-friendly ітерфейс, що спрощує взаємодію з платформою для будь кого, а також як сказано вище є цілком безкоштовною, що дає можливість навчатись різним соціальними ланкам.

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      | ДТЕУ 121 06-23.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата |                   | 9     |

## 1.2 Загальні теоретичні відомості проблем розробки сучасності і способи їх вирішення

Сучасне програмне забезпечення стає все більш складним внаслідок зручного інтерфейсу, розширених можливостей та використання програмування подій. Ці фактори призводять до ускладнення процесу розробки програмного забезпечення. Виготовлення зручного у використанні програмного забезпечення стає джерелом його складності. Сьогодні, для побудови сучасної програми, потрібно більше, ніж просте об'єднання машинних інструкцій, операторів мови високого рівня або наборів процедур і модулів. Головним завданням стає розробка програми з чіткою структурою, яка легко модифікується, вільна від помилок і стійка до змін. Об'єктно-орієнтована технологія створення програмного забезпечення була розроблена як засіб подолання складності. Вона успадкувала найкращі практики структурного і модульного програмування та впровадила нові підходи до проектування програмного забезпечення. Головним завданням об'єктно-орієнтованого підходу є створення структурованої програми, що керує складними залежностями між компонентами системи. Використання об'єктно-орієнтованого підходу зменшує кількість залежностей між компонентами, дозволяє об'єктам взаємодіяти через чіткі інтерфейси, сприяє виявленню і усуненню помилок та захищає дані від несанкціонованого доступу. Інтерфейси класів гарантують модульність і переносимість компонентів програми. Однією з основних переваг об'єктно-орієнтованої технології є можливість прямого моделювання об'єктів реального світу в об'єкти програми, що спрощує проектування порівняно з проектуванням процедурного підходу.

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      | ДТЕУ 121 06-23.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата |                   | 10    |



*Рисунок 1.3. Основи об'єктно орієнтованого програмування*

*Джерело: побудовано автором в системі DrawIO*

На думку амереканського науковця у галузі теорії обчислювальних систем Алана Кея, розробника мови Smalltalk, якого вважають одним з засновників ООП (об'єктно-орієнтованого програмування), підхід полягає в використанні даних принципів:

- в основі всього лежить об'єкт;
- всі дії та розрахунки виконуються шляхом обміну даними між об'єктами, під час якої один потребує, щоб інший виконав деяку дію. Так об'єкти працюють разом, надсилаючи і отримуючи повідомлення;
- повідомлення — це запит на виконання дії, доповнений набором супроводжуючих функцій, які можуть знадобитися під час виконання дії;
- кожен об'єкт складається з об'єктів;
- кожен об'єкт є представником класу, який виражає загальні властивості об'єктів;
- у класі задається поведінка об'єкта саме тому вони належать одному класу і можуть виконувати однакові задачі [4].

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      | ДТЕУ 121 06-23.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата |                   | 11    |



Якщо бути точним щодо розуміння ООП (об'єктно-орієнтованого програмування), то це є програмування з надсиланням повідомлень об'єктам невідомого типу. Такі об'єкти можна вибирати з деякої структури даних, наприклад, масиву чи колекції. Всі об'єкти в колекції мають певні спільні характеристики (наприклад, позиція на екрані, здатність переміщуватись, активуватись та деактивуватись). Крім того існує певний перелік повідомлень, на які можуть відповідати усі ці об'єкти. З точки зору програміста вибір об'єкта з колекції не надає інформації про його тип [2].

В об'єктно-орієнтованому програмуванні кожен об'єкт представляє собою принципово динамічну сутність, тобто змінюється в залежності від часу (а також від впливу зовнішніх факторів по відношенню до нього). Інакше кажучи, об'єкт володіє тим чи іншим способом поведінки. У відношенні абстракції як об'єкта, поведінка полягає у додатку функції до аргумента [13].

Також важливою складовою концепцією об'єктно-орієнтованого програмування є інтуїтивно зрозуміле поняття наслідування. У звичайній постановці під наслідуванням розуміється змога того чи іншого об'єкта, який є похідним від якогось базового, зберігати поведінка а саме, атрибути та операції над ними, характерне для батьківського об'єкта. З точки зору мов програмування поняття спадкування означає застосовність всіх або лише деяких властивостей або методів базового, або батьківського класу для всіх класів, похідних від нього. Крім того, збереження властивостей та методів базового класу має забезпечуватися і для всіх конкретизацій (тобто конкретних об'єктів) будь-якого похідного класу [13].

В основі об'єктно-орієнтованої мови програмування лежать два основних поняття: клас та об'єкт. Об'єкт – це базове поняття в ООП (об'єктно-орієнтоване програмування), це конкретна реалізація, екземпляр класу.

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      | ДТЕУ 121 06-23.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата |                   | 12    |

Об'єкт складається з трьох частин: стан (змінні стану), методи (операції), ім'я об'єкта. Клас - це група даних і методів або функцій для роботи з цими даними, це шаблон. Об'єкти однаковими наборами змінних стану і методів, утворюють клас. Якщо об'єкти мають реалізацію з конкретного світу, то класи є абстракціями. Трохи більш складні об'єкти можуть взагалі не містити даних, а представляти процес і містити тільки функції, які реалізують цей процес. Для формування реального об'єкта необхідно мати шаблон, по прикладу якого і будується даний об'єкт [1].

Основним поняттями ООП (об'єктно-орієнтоване програмування) є інкапсуляція, поліморфізм та успадкування. Об'єкти, розташовані на робочому столі ОС Windows, відомі як поліморфні. Це означає, що вони мають певні спільні характеристики, такі, як здатність бути поміщеними в одну колекцію і здатність відповідати на однакові повідомлення. Поліморфізм безпосередньо пов'язаний з успадкуванням. За допомогою успадкування можна визначати нові об'єкти, вказуючи лише, чим вони відрізняються від уже визначених [2].

Інкапсуляція це здатність об'єктів скривати деякі способи обробки даних та самі дані від навколишнього цифрового середовища. Історія розвитку методологій програмування спонукувана боротьбою зі складністю розробки програмного забезпечення. Складність великих програмних систем, у створенні яких бере участь відразу велика кількість розробників, зменшується, якщо на верхньому рівні не видно деталей реалізації нижніх рівнів. Власне, процедурний підхід був першим кроком на цьому шляху. Під інкапсуляцією розуміється приховування інформації про внутрішній устрій об'єкта, при якому робота з об'єктом може вестися тільки через його загальнодоступний інтерфейс. Або це механізм програмування, об'єднуючий разом код і дані, якими він маніпулює, виключаючи як втручання ззовні, так

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      | ДТЕУ 121 06-23.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата |                   | 13    |

і неправильне використання даних. В об'єктно-орієнтованій мові програмування дані і код можуть бути об'єднані в абсолютно автономний чорний ящик. Усередині такого ящика знаходяться всі необхідні дані і код. Коли код і дані зв'язуються разом подібним чином, створюється об'єкт. Іншими словами, об'єкт - це елемент, що підтримує інкапсуляцію [3].

Поліморфізм - також одна з трьох основ ООП (об'єктно-орієнтоване програмування). Якщо говорити коротко, поліморфізм - це здатність об'єкта використовувати методи похідного класу, який не існує на момент створення базового. Слово «поліморфізм» можна перекласти як «багато форм». Тобто це можливість використання одного і того ж імені операції або методу до об'єктів різних класів, при цьому дії, що здійснюються з об'єктами, можуть істотно різнитися. Тому можна сказати, що в одного і того ж слова багато форм [3].

І остання основа успадкування - стосується здатності мови дозволяти будувати нові визначення класів на основі визначень існуючих класів. По суті, спадкування дозволяє розширювати поведінку базового класу, наслідуючи основну функціональність у похідному підкласі. Спадкування являє собою процес, в ході якого один об'єкт набуває властивостей іншого об'єкта. Це дуже важливий процес, оскільки він забезпечує принцип ієрархічної класифікації. Якщо вдуматися, то велика частина знань піддається систематизації завдяки ієрархічній класифікації [3].

При дотриманні усіх вище описаних принципів можна уникнути складнощів та в результаті отримати структурований та якісний продукт, що буде легкий в масштабуванні, підтримці та його освоєнню для інших розробників так як написаний за загально прийнятими правилами.

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      | ДТЕУ 121 06-23.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата |                   | 14    |

## 1.2 Опис предметної області та мета створення системи

Програмний додаток платформи ІТ академії на основі фреймворку Spring Boot – це система, розроблена для програми навчання в ІТ сфері. Її основна ціль полягає в тому щоб надавати якомога більшій кількості людей якісні знання та освіту.

В сучасному світі розвиток технологій відбувається дуже швидко. Доволі велика кількість людей хотіли б йти в ногу з часом та зробити це своєю кар'єрою і справою життя. Проте в нинішніх умовах, якщо людина хоче пов'язати своє професійне майбутнє з ІТ сферою, їй потрібно йти в університет і навчатись. Так, це хороший варіант, проте дуже багато людей віком від 30 до 60 років хотіли спробувати, дізнатись що це, і почати працювати в цій галузі. Проте коли є сім'я і діти зазвичай у людей немає так багато можливостей витратити 4 роки навчання в університеті для освоєння професії. Тому актуальність платформи ІТ академії буде дуже поширеним та прибутковим в наш час. Так само можна проаналізувати людей, які ще навчаються в школі, але хочуть вчити не тільки профільні предмети, а й щось нове і цікаве. Завдяки розробленій платформі ІТ академії, це буде можливо в будь-якому віці та в доволі обмежену кількість часу.

Всім відомо наскільки швидко розвивається ІТ сфера та як багато людей прагнуть працювати в ній. В результаті цього попит на такі платформи буде зростати щодня, саме тому це є дуже вигідною інвестицією також.

Також це ще один варіант доступу до якісної освітньої програми, де майбутніх студентів академії будуть навчати новітніми методами, процесам роботи в різних сферах. Звичайно, що людина зможе обирати напрямок, який їй буде цікаво і вчити безпосередньо його, що дозволить знизити кількість вивченого та бути висококваліфікованим професіоналом своєї справи.

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      | ДТЕУ 121 06-23.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата |                   | 15    |

Такий підхід в навчанні та готуванні нових спеціалістів допоможе зробити освіту в країні якіснішою, доступнішою. Спеціалістів більш профільними та освіченими. Тобто в результаті буде ставати вищим рівень якості працівників, що дозволить розширити ринок та створювати самим нові технології. Одним із важливих як і соціальних так і людських аспектів стане руйнування міфу про те що за 30 вже пізно щось змінювати, бо дуже багато талантів залишаються непомітними, бо їм не дають можливості для розвитку і становлення.

А тепер згадаємо пандемію у сучасний стан в Україні – повномасштабне вторгнення. В результаті таких подій процес навчання і роботи має дистанційний формат. І це такі формати стали доволі зручні, недорогі у використанні та вже перевірені. Люди вже звикли працювати вдома, тим паче доведено, що рівень працездатності працівника, який працює чи навчається вдома набагато вищий ніж у людини, яка навчається безпосередньо в приміщенні. Цей фактор дозволяє поширюватись онлайн бізнесу. Завдяки цьому люди зможу навчатись на ІТ платформі з різних куточків України, а то і світу.

### 1.3 Технічне завдання

Розробка це дуже важливий етап у житті додатку, але не менш важливим є якісно створене технічне завдання. ТЗ повинно вмішувати в собі всю інформацію про майбутній додаток від інтерфейсу до вимоги до захисту. У загальному є декілька сталих вимоги, що повинні бути описанні для кожного додатку.

Вимоги до дизайну: дизайн має бути простим та зрозумілим для користувача будь-якого рівня обізнаності, вся кольорова гама веб сторінки

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      | ДТЕУ 121 06-23.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата |                   | 16    |

повинна бути у спокійних кольорах (білий, синій, чорний) без елементів які будуть відволікати увагу від навчання.

Вимоги до технологій: розробка проводиться лише з використанням актуальних і нових інструментів чого можна досягнути використовуючи ці технології: Spring Boot, Bootstrap, MySQL. Вся архітектура побудована на фундаментальних та легко масштабованих технологіях таких як ООП, SOLID, MVC.

Вимоги до надійності: додаток має бути стабільним та безперебійним у роботі, що можна отримати за допомогою мануального тестування за для перевірки на наявність дефектів і їх усунення.

Вимоги до захисту користувача: система повинна бути захищена від втрати даних про зареєстрованого користувача і цього можна досягти за допомогою впровадження автентифікації користувачів, а також опрацювати доступність до БД за для уникнення зломів таких як SQL-ін'єкція.

Вимоги до антивірусного захисту: так як це веб додаток, доступ до якого можна отримати лише з браузера, який в свою чергу вже має велику кількість антивірусних перевірок і в такому випадку можна стверджувати про захищеність системи від вірусів.

Також можна виділити основні тези яких потрібно дотримуватись при розробці:

- Ефективність: програмний додаток повинен мати оптимальну швидкість роботи та ефективно використовувати ресурси комп'ютера
- Надійність: програмний додаток повинен бути надійним та стабільним, щоб забезпечити коректну роботу протягом тривалого часу
- Легкість використання: програмний додаток повинен бути простим та зручним у використанні. Додаток повинен мати зрозумілий та зручний інтерфейс користувача.

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      | ДТЕУ 121 06-23.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата |                   | 17    |

Основаючись на вище написаних вимогах та чітко побудованому технічному завданню можна починати працювати над розробкою додатку. У нашому випадку робота над застосунком буде йти за методологією SCRUM, що забезпечить гнучкість та модульність процесу розробки так як уся робота буде поділена на sprint, які дозволять включити постійний моніторинг та контроль якості продукту.

#### 1.4 Висновки до розділу 1

У висновку, аналіз предметної області та постановка технічного завдання для платформи ІТ академії на основі фреймворку Spring Boot, включали розгляд основних теоретичних аспектів розробки сучасних систем та способів їх вирішення. Безпосередньо опис предметної області детально розкривав основні аспекти платформи ІТ академії, а також встановлював мету її створення. Що є покращення та надання можливостей людям здобути освіти і кваліфікацію, в результаті чого працювати, незалежно від віку та можливостей.

В рамках технічного завдання були визначені конкретні вимоги до розробки системи, зокрема щодо використання фреймворку Spring Boot. Завдання передбачало створення потужної, масштабованої та ефективною платформи, зосередженої на навчанні студентів у галузі ІТ. Були визначені функціональні та технічні вимоги до системи, а також задокументовані основні засоби та інструменти, які будуть використовуватися під час розробки. Також були додані такі вимоги до: надійності, технологій, дизайну, захисту користувача та антивірусного захисту.

Результати аналізу предметної області та постановки технічного завдання підтверджують актуальність та рівень високого попиту в результаті

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      | ДТЕУ 121 06-23.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата |                   | 18    |

створення платформи ІТ академії на основі фреймворку Spring Boot. Ця платформа буде

сприяти розвитку інформаційно-технологічної галузі, надавати якісну освіту студентам. Використання фреймворку Spring Boot дає можливість створити ефективну та гнучку систему, яка зможе відповідати вимогам сучасного ринку.



|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      | ДТЕУ 121 06-23.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата |                   | 19    |



## РОЗДІЛ 2

### ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 2.1 Вибір середовища розробки, мови програмування та відповідного інструментарію

Індекс ТІОВЕ, який оцінює популярність світових мов програмування, на основі підрахунку результатів пошукових запитів, підніс Java на п'єдестал у 2018 році та віддав їй абсолютну першість серед програмістів країн СНД та ЄС. [6]

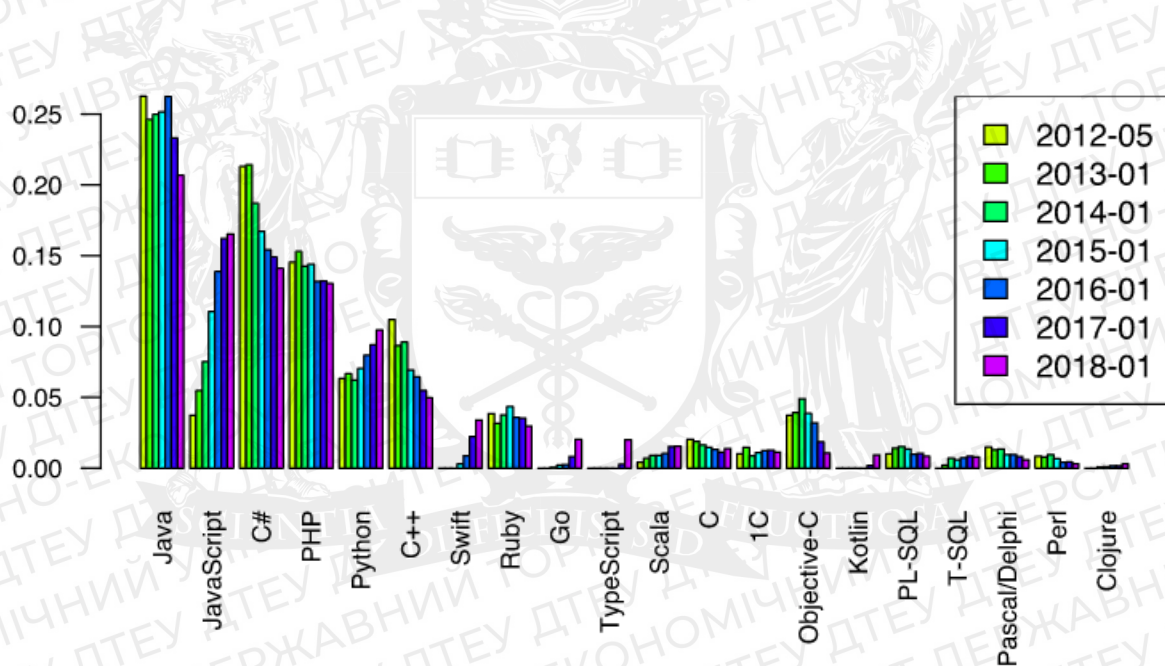


Рисунок 2.1. Динаміка популярності мов програмування

Джерело: побудовано на основі джерела[5]

Якщо брати статистику останніх років, то за шкалою популярності можна побачити зменшення позиції Java у рейтингах, але все одно залишається на найвищих позиціях серед інших мов. Якщо брати до уваги

|           |       |                 |        |          |   |  |       |         |
|-----------|-------|-----------------|--------|----------|---|--|-------|---------|
|           |       |                 |        |          | <i>ДТЕУ 121 06-23.БР</i>                                  |  |       |         |
| Зм.       | Аркуш | № докум.        | Підпис | Дата     |   |  |       |         |
| Зав. каф. |       | Криворучко О.В. |        | 03.03.23 | Платформа ІТ-академії на основі<br>фреймворку Spring Boot | Стадія   | Аркуш | Аркушів |
| Керівник  |       | Десятко А. М    |        | 03.03.23 |   | P2   | 20    | 50      |
| Гарант    |       | Рзаєва С. Л.    |        | 03.03.23 |   | Факультет інформаційних<br>технологій<br>4 курс, 6 група |       |         |
| Розробив  |       | Скрипник Д. О.  |        | 03.03.23 |   |  |       |         |
|           |       |                 |        |          | <i>Проектування програмного<br/>забезпечення</i>          |  |       |         |

саме Back-End розробку то навіть на сьогодні Java залишається топ 1 в рейтингу.[7]

Java – об’єктно-орієнтована та крос-платформна мова програмування, випущена компанією Sun Microsystems у 1995 році (з 2009 року правами володіє компанія Oracle). Спершу програми на Java створювалися для побутових електронних приладів, однак сьогодні ця мова застосовується практично в усіх комп’ютерних сферах, зокрема для написання клієнтських застосунків і серверного програмного забезпечення.[8]

Великого успіху мова програмування Java отримала за вислів «Написано одного разу, працює скрізь» і це одна з головних причин її швидкого розвитку та розповсюдження, адже за допомогою крос-платформності можна було написати код на будь якій операційній системі і потім його запустити на іншій, що в свою чергу інші мови не могли зробити. В основі крос-платформності лежить лежить JDK, JRE і JVM, більше про них:

- JDK (Java Development Kit) - це набір для розробки програмного забезпечення, необхідний для розробки java-програм. Коли Ви завантажуєте JDK, JRE також завантажується разом з ним;[9]
- JRE (Java Runtime Environment) – це програмний пакет, який надає бібліотеки класів Java, віртуальну машину Java (JVM) та інші компоненти, необхідні для запуску java-програм; [9]
- JVM (Java Virtual Machine) – це абстрактна машина, яка дозволяє Вашому комп’ютеру запускати java-програму. [9]

Також до переваг Java потрібно віднести те, що вона втілює усі принципи об’єктно орієнтованого програмування за допомогою, якого можна створити додаток будь якої складності, адже навіть у компанії NASA ця мова активно використовується і на даний момент. З часом розвитку даної мови програмування розробники намагались якомога збільшити її можливост

|     |       |         |        |      |  |                   |       |
|-----|-------|---------|--------|------|--|-------------------|-------|
|     |       |         |        |      |  | ДТЕУ 121 06-23.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата |  |                   | 21    |

і та автоматизувати певні процеси, що сприяло до виникнення фреймворків, які максимально покращили життя розробникам, найкращим прикладом такої технології це Spring.

Spring Framework — це потужний фреймворк розробки додатків Java із відкритим вихідним кодом, який дозволяє легко створювати надійні високопродуктивні додатки. Відкритий вихідний код означає те, до чого люди можуть отримати доступ, змінювати та вільно ділитися, оскільки вихідний код є у вільному доступі.

Фреймворк Spring має дуже великий функціонал, що надзвичайно спрощує процес розробки додку, з основних його можливостей можна виділити такі модулі:

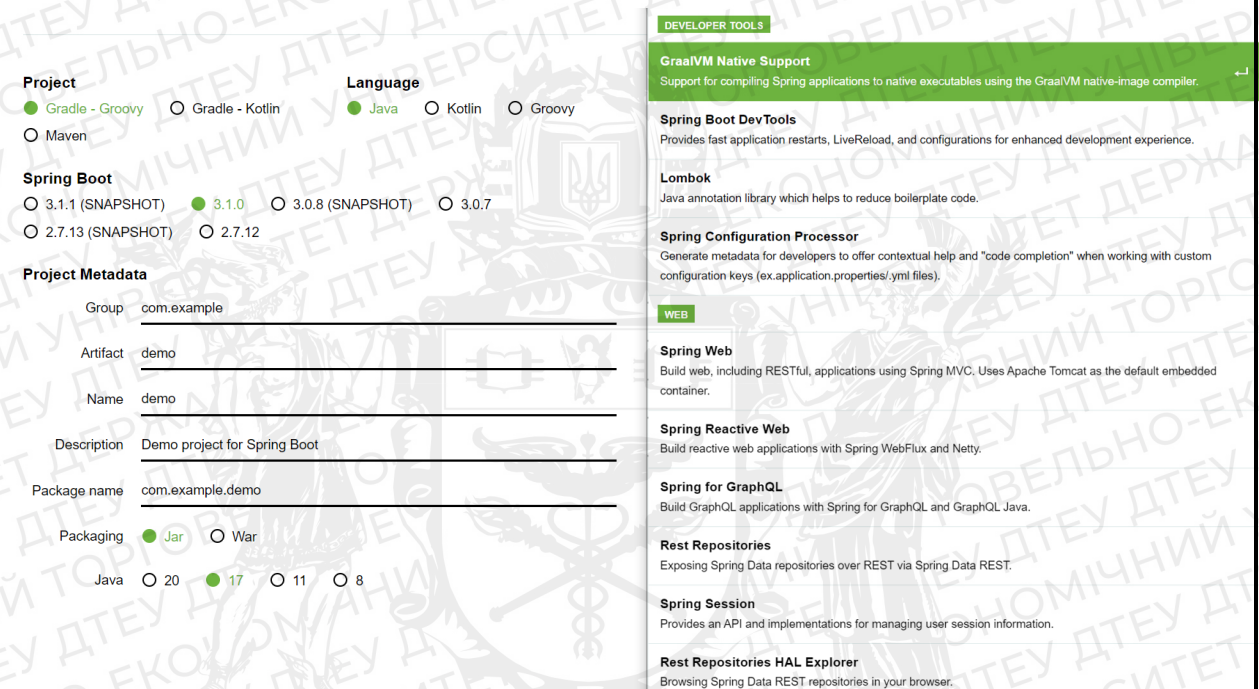
- Spring Core: основа фреймворка, яка також забезпечує IoC та AOP;
- Spring Data: модуль який допомагає працювати з БД;
- Spring MVC: модуль для побудови веб-додатків;
- Spring Security: модуль для реалізації захисту додатку за допомогою аутентифікації та авторизації;
- Spring Test: модуль, що допомагає в написанні юніт тестів для додатку;
- Spring Integration: модуль, що спрощує інтеграцію додатку з іншими;
- Spring Cloud: модуль за допомогою якого можна працювати з хмарними сервісами.[10]

Усі ці модулі можуть бути використані незалежно один від одного або в поєднанні з іншими, що дає можливість розробляти різноманітні додатки на основі Java з гнучкістю та можливістю розширення.

Ще більше інноваційним надбудовою над фреймворком Spring був Spring Boot, який пішов далі і якщо для початку розробки потрібно було налаштувати середовище власноруч то у другому випадку це вже

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      |                   | Аркуш |
|     |       |         |        |      | ДТЕУ 121 06-23.БР | 22    |
| Зм. | Аркуш | № докум | Підпис | Дата |                   |       |

відбувалось автоматично лише за допомогою середовища в якому можна обрати потрібні в розробці модулі і все інше за тебе зробить Spring Boot. Також зміни, а точніше спрощення роботи відбулося і для кожного окремого модуля, як наприклад спілкування додатка з БД, а саме автоматичне її конфігурування, логування, обробки виключень та багато іншого.



*Рисунок 2.2. Spring Initializr – додаток за допомогою якого можна створити вже проект з усіма необхідними налаштуваннями*

*Джерело: побудовано автором на платформі Spring Initializr(знімок екрану)*

Загалом можна стверджувати, що надбудова Spring Boot прагне зменшити довжину коду і спростити розробку веб-додатків з використанням усього потенціалу фреймворку на базі якого він побудований.

Існує велика кількість середовищ розробки які підтримують написання на мові програмування Java, наприклад: IntelliJ, jGRASP, JCreator, BlueJ, Eclipse, Oracle JDeveloper. Але провівши аналіз, підтримки та популярності

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      |                   | Аркуш |
|     |       |         |        |      |                   | 23    |
| Зм. | Аркуш | № докум | Підпис | Дата | ДТЕУ 121 06-23.БР |       |

кожного з IDE для написання платформи обрано IntelliJ так як дане середовище має усі необхідні функції. Також слід взяти увагу на інтерфейс обраного середовища, яке наповнено великою кількістю корисних інструментів, що також спрощує процес розробки, як наприклад роботу з системою контролю версій – GIT. Без написання команд в терміналі є можливість зробити будь-яку операцію лише за допомогою кнопок.

Останнім, але не менш важливим у розробці платформи ІТ академії є визначання яку систему управління базами даних (СУБД) обрати. Як і середовищ існування так і СУБД є велика кількість, але із усіх одним із найкращих вибором є – MySQL. Вона є однією з найпопулярніших системою управління БД саме тому і має велику спільноту користувачів, які постійно оновлюють, підтримують та документують зміни, що зменшує кількість уникнення можливих затримок при розробці так як на будь-яке питання вже є відповідь. Також до її переваг потрібно віднести: висока продуктивність, надійність, простота використання і масштабованість.

Загрегувавши все перелічене вище, отримуємо готовий набір необхідних інструментів для реалізації програмного додатку мовою програмування Java з використанням фреймворку Spring Boot. Який повністю покриває усі потреби технічного завдання. Підтримує використання усіх сучасних принципів програмування, а саме ООП, методології типу SOLID та патернів проектування, а саме MVC.

## 2.2 Архітектура розроблюваного програмного забезпечення

В основі архітектури програмного забезпечення головною ціллю є максимальне зниження складності за допомогою використання абстракції, яка дозволяє виносити основні сутності на вищий рівень тим самим розмежовуючи повноваження між ними. Окрім зниження складності, архітектура програмного

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      |                   | Аркуш |
|     |       |         |        |      |                   | 24    |
| Зм. | Аркуш | № докум | Підпис | Дата | ДТЕУ 121 06-23.БР |       |

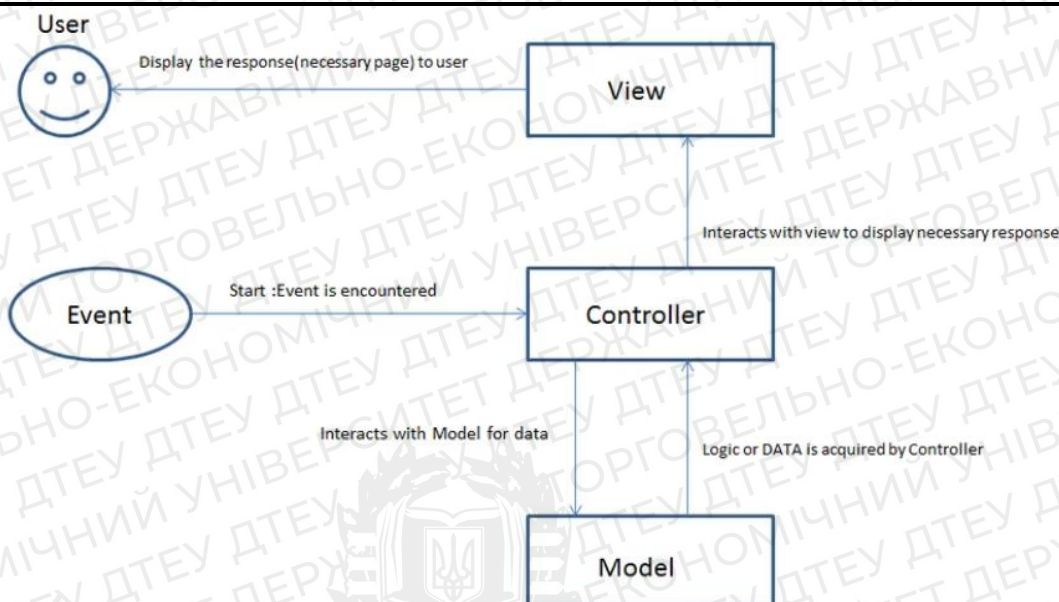
забезпечення також спрямована на досягнення інших цілей таких як розширюваність, повторне використання, покращення перформансу (збільшення ефективності роботи додатку) та надійність. Загалом можна виокремити основні концепції, які використовуються в АПЗ:

- модульність - що забезпечує легку масштабованість і спрощує кількість коду;
- розшарування – є запорукою безпеки так я розшарування на такі рівні, як доступу до БД, бізнес логіки та представницького рівня мінімізує можливість отримати доступ до конфіденційної інформації;
- клієнт-серверна архітектура – що забезпечує розподілення програми на дві частини (клієнтська та серверна).

Отже можна співставити архітектуру програмного забезпечення з фундаментом який закладається для розробки програмного додатку і обрання правильного шаблону проектування надзвичайно важливо. У розробленому програмному додатку використовується популярний патерн програмування, який забезпечить розбиття на модулі та досягнення усіх вище перелічених концепцій АПЗ - це MVC (Model View Controller).

MVC - це архітектурний шаблон, який застосовується при проектуванні та розробці програмного забезпечення. Цей шаблон розділяє систему на три компоненти: модель, вигляд та контролер. Він використовується для відокремлення даних (model) від інтерфейсу користувача (view), щоб зміни в інтерфейсі користувача мінімально впливали на обробку даних. Водночас, зміни в моделі даних можуть відбуватися без необхідності внесення змін до інтерфейсу користувача.[12]

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      |                   | Аркуш |
|     |       |         |        |      |                   | 25    |
| Зм. | Аркуш | № докум | Підпис | Дата | ДТЕУ 121 06-23.БР |       |



*Рисунок 2.3. Зображення MVC у вигляді діаграми*

*Джерело: побудовано на основі джерела[12]*

Якщо узагальнити вище описане то можна зробити такий висновок, що шаблон MVC допомагає забезпечити чистоту коду, полегшує розуміння структури програми і забезпечує більшу гнучкість та розширюваність в процесі розробки програмного забезпечення.

Для більш детального розуміння архітектури додатку необхідно побудувати UML (Unified Modeling Language) діаграми, що міститимуть в собі розуміння, як проектувати платформу ІТ академії, допоможуть моделювати поведінку застосунку, структурувати загальні вимоги та аналізувати потреби користувачів.

UML - уніфікована мова моделювання, що використовується розробниками програмного забезпечення для візуалізації процесів та роботи систем.[11]

Проаналізувавши основну поведінку користувача та можливе використання додатку було побудовано діаграму варіантів використання (Use case), яка демонструє поведінку системи при взаємодії з зовнішнім середовищем (рис. 2.4).

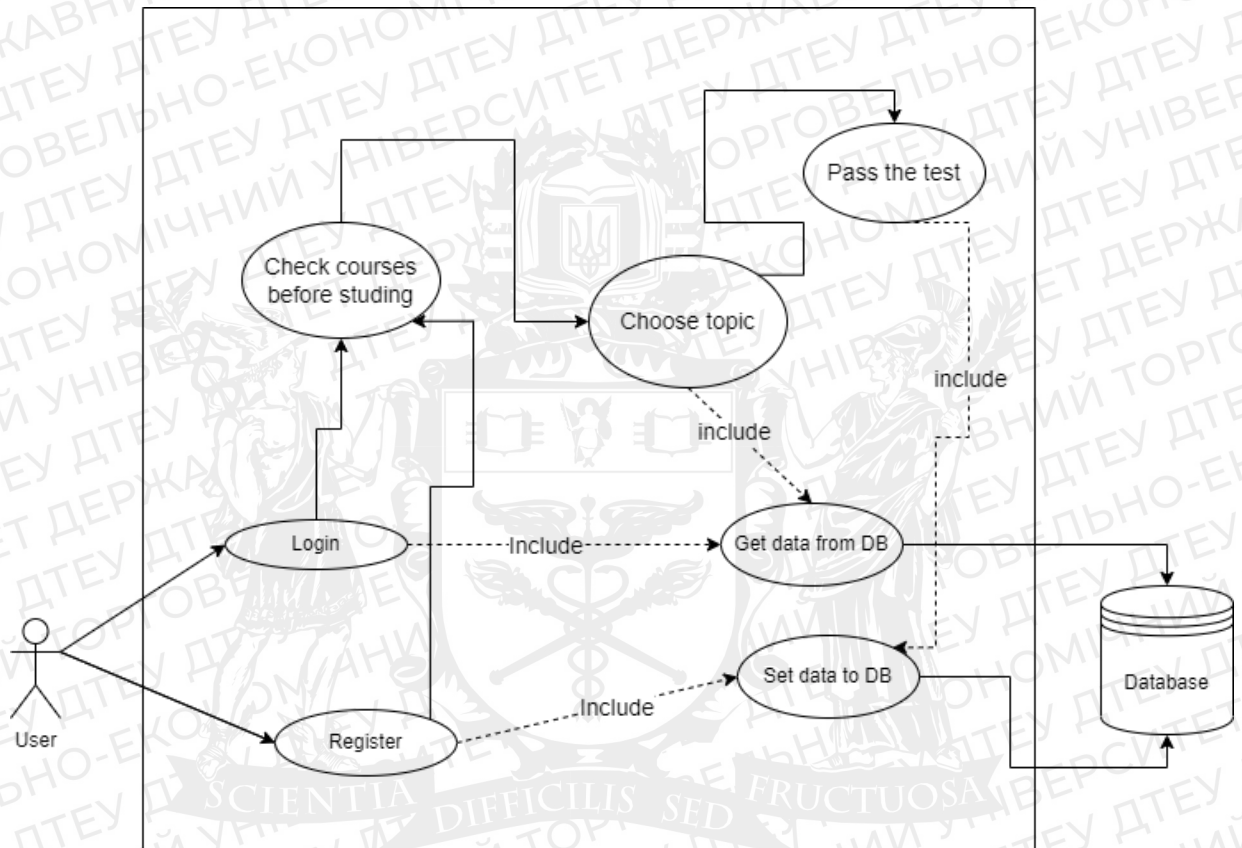


Рисунок 2.4. Діаграма прецедентів, сценарій використання

Джерело: побудовано автором в системі DrawIO

Наступним кроком буде візуалізація процесу використання, що має назву – Діаграма діяльності (Activity Diagram). За її допомогою можна продемонструвати загальну роботу крок за кроком (рис. 2.5).



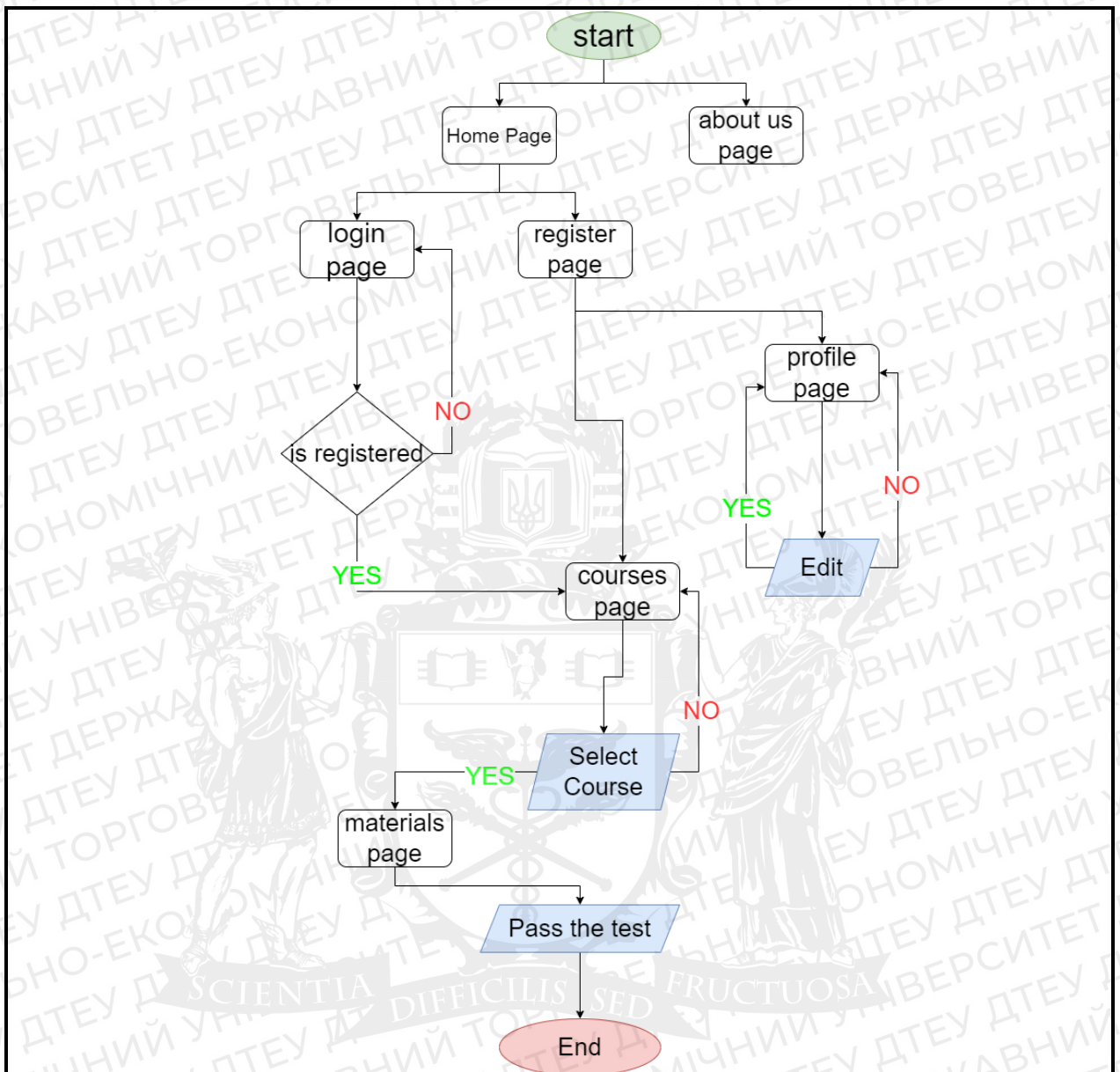


Рисунок 2.5. Діаграма діяльності

Джерело: побудовано автором в системі DrawIO

Для моделювання структури програмного коду необхідно створити діаграму класів, що дозволить візуалізувати класи, змінні, методи, ще до початку розробки і на її основі побудувати платформу з усіма дотриманнями принципів ООП (рис. 2.6).

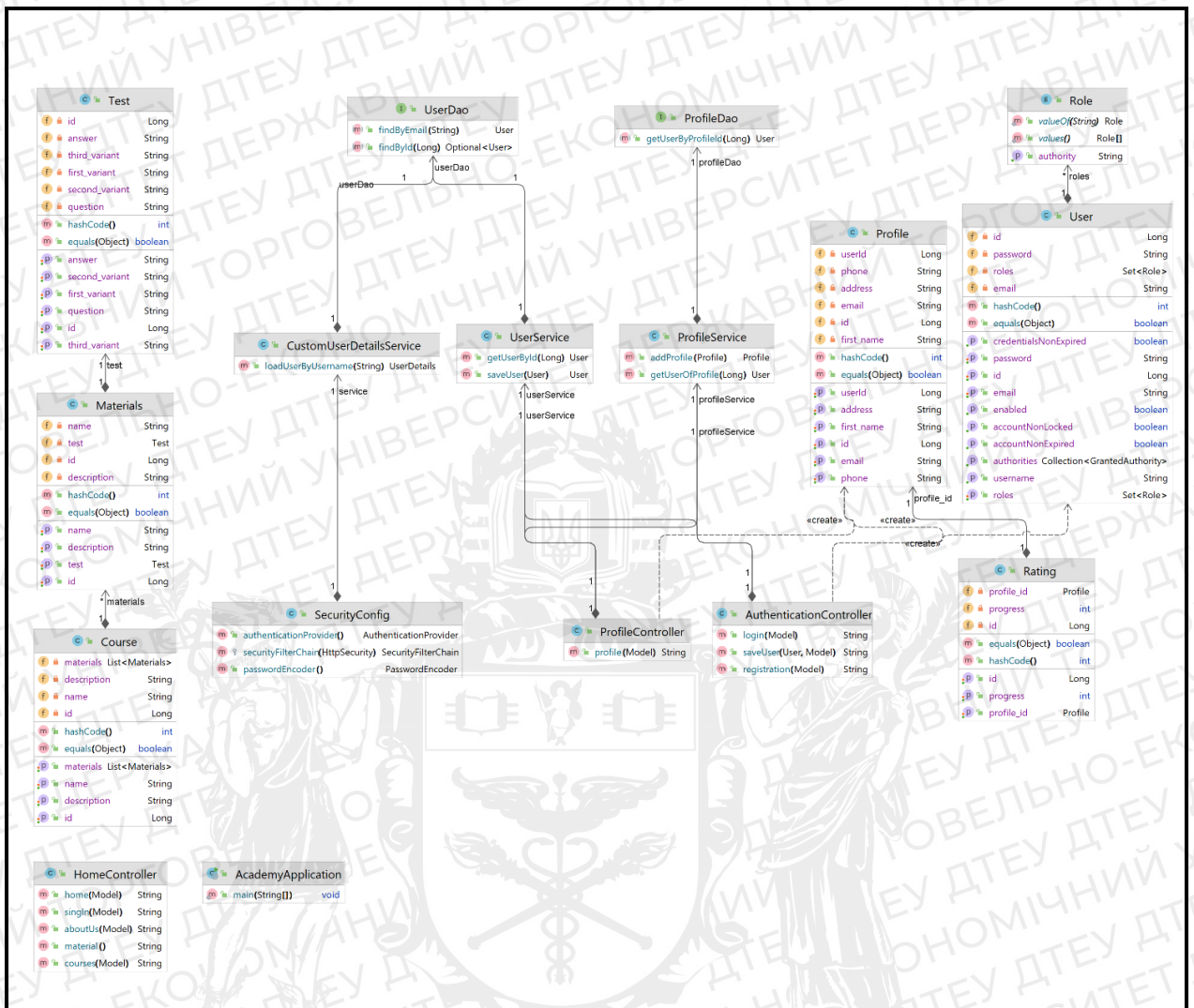


Рисунок 2.6. Діаграма класів розроблювального застосунку

Джерело: побудовано автором в системі Intelij IDEA (знімок екрану)

Усі дані які використовуються у програмному додатку зберігаються в базі даних, тому постає задача у побудові реляційної БД, яка забезпечить цілісність та можливість для розширення. Реляційна база даних представляє собою безліч пов'язаних між собою таблиць, кожна з яких містить інформацію, що стосується об'єкту, як наприклад сутність User з полями: email, password і обов'язково id. Підвищення уваги до параметру id не є випадковістю, адже це обов'язковий параметр, який в свою чергу дозволяє наблизитись до побудови нормалізованої БД, так як кожний непервинний атрибут повністю залежить від первинного ключа (друга нормальна форма).

|     |       |         |        |      |       |
|-----|-------|---------|--------|------|-------|
|     |       |         |        |      | Аркуш |
|     |       |         |        |      | 29    |
| Зм. | Аркуш | № докум | Підпис | Дата |       |

Нормалізація БД є дуже важливою складовою у процесі розробки, адже це дозволяє досягти консистентності даних, розбиття їх на менші та більш логічні структури, розширити функціональність та збільшення продуктивності роботи за рахунок меншого навантаження і все це можна отримати за рахунок дотримання декількох правил: уникнення повторювальних полів і нульового значення у рядку, наявність первинного ключа та щоб усі неключові поля в таблиці залежали від первинного ключа цієї таблиці, а не від інших неключових полів.

При побудові зв'язків для таблиць реляційної бази даних важливим етапом у проектуванні є створення логічної та фізичної моделі (рис. 2.7).

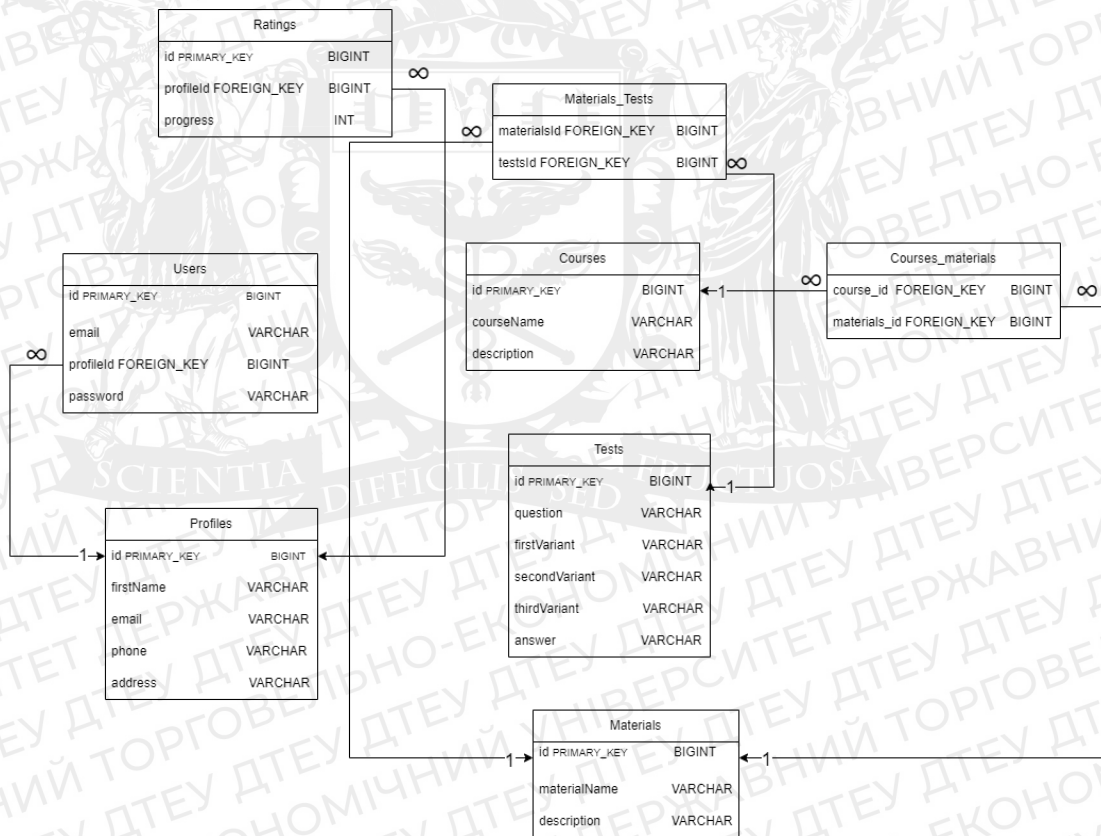


Рисунок 2.7. Поєднана діаграма фізичної та логічної моделі

Джерело: побудовано автором в системі DrawIO (знімок екрану)

### 2.3 Розробка графічного інтерфейсу

На початку роботи над макетом для платформи ІТ академії потрібно визначитись з інструментами та сервісами які будуть використовуватись для

реалізації задачі. Опираючись на напрямок в ІТ галузі – UX/UI дизайнери, робота який прораховування усього життєвого циклу користувача на сайті та реалізація макетів відносно технічного завдання було вирішено застосувати один із найвідоміших у даній сфері застосунок - Figma. Даний додаток має великий набір інструментів, які в свою чергу повністю покривають усі етапи роботи над інтерфейсом.

При першому візиті на веб-сторінку користувача повинна зустрічати основна сторінка, яка виконана у спокійних кольорах і заохочує користувача залишитись та продовжити ознайомлення з платформою. Для платформи ІТ академії були обрані світлі кольори з невеликою кількістю елементів синього кольору (картинки, кнопки). Було прийнято рішення використовувати саме поєднання білого та синього, адже білий колір є нейтральним, а синій за психологією кольорів викликає у користувачів відчуття довіри та спокою та асоціюється з стабільністю та безпекою, що в нашому випадку підкреслює рівень якості продукту.

Також новий користувач як не зареєстрований може відвідати сторінку “About us” на якій описанні основні переваги почати навчання на платформі (рис. 2.8).

|     |       |         |        |      |  |                   |       |
|-----|-------|---------|--------|------|--|-------------------|-------|
|     |       |         |        |      |  | ДТЕУ 121 06-23.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата |  |                   | 31    |

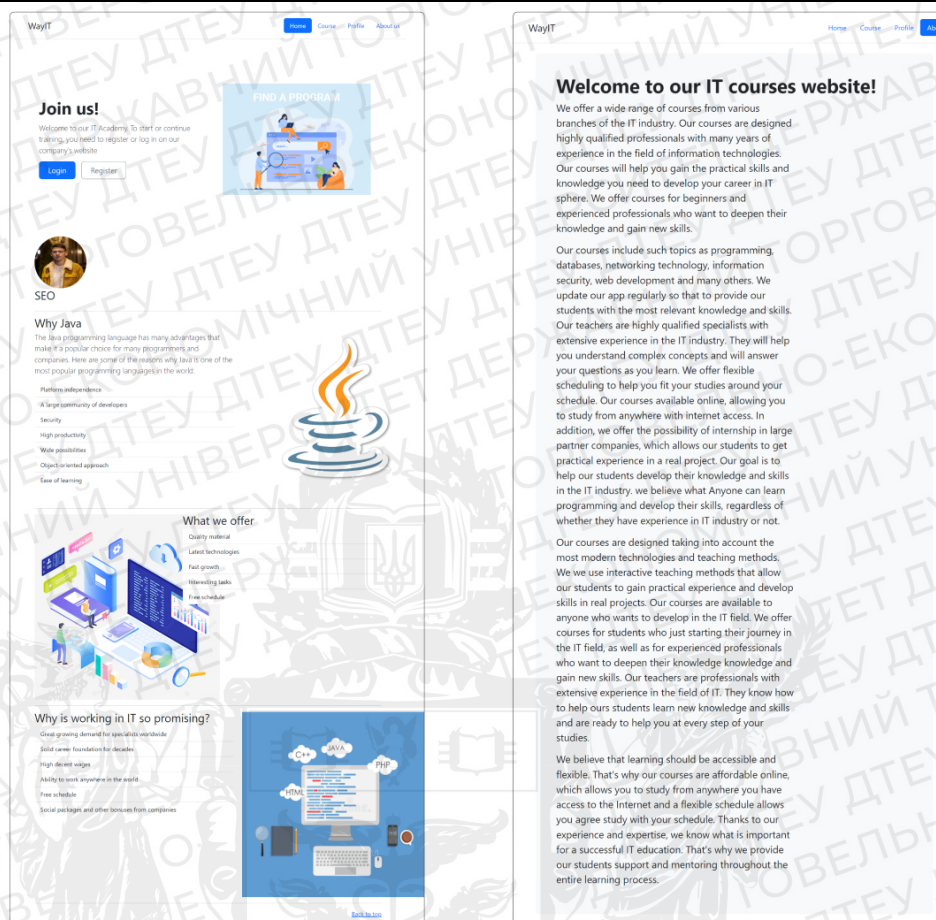


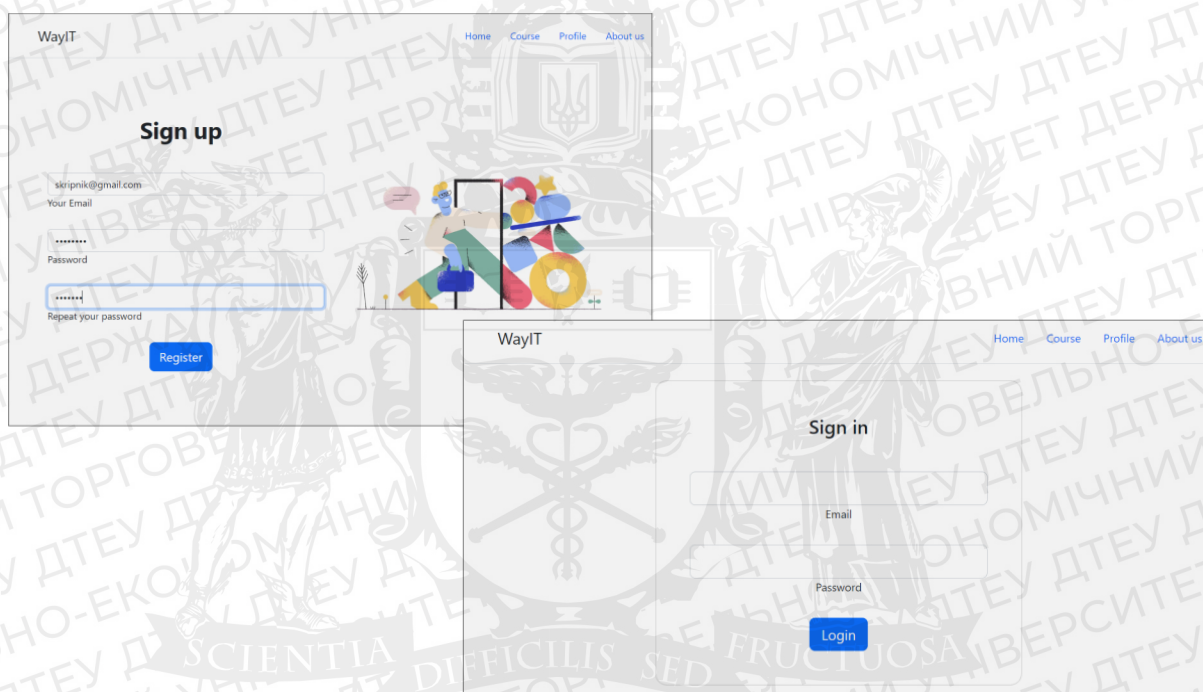
Рисунок 2.8. Домашня та інформаційна сторінка

Джерело: побудовано автором в системі Figma (знімок екрану)

Наступним кроком користувача повинна бути реєстрація або автентифікація, адже доступ до безкоштовного курсу можна отримати лише верифікованому користувачу. Для зручності на основній сторінці вже розміщені кнопки за якими можна перейти та увійти до системи. У випадку, якщо не зареєстрований користувач забажає перейти на одну з сторінок (Course, Profile) та натисне кнопку для переходу – його відправить на сторінку логін.

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      |                   | Аркуш |
|     |       |         |        |      |                   | 32    |
| Зм. | Аркуш | № докум | Підпис | Дата | ДТЕУ 121 06-23.БР |       |

У проектуванні макету для сторінок реєстрації та автінтефікації було прийнято рішення притримуватись кольорової гами та стратегії, що була зазначена раніше. Тому в результаті отримуємо максимально інтуїтивні та не ускладненні великою кількістю реклам чи інших факторів, які можуть відволікти користувача від його перебуванні на платформі, що дуже впливає на перше позитивне враження (рис. 2.9).



*Рисунок 2.9. Сторінки реєстрації та авторизації*

*Джерело: побудовано автором в системі Figma (знімок екрану)*

Після того як користувач вже є авторизованим для нього відкриваються доступ до усього функціоналу платформи ІТ академії, а саме сторінка Course, Profile та Materials (з'являється лише після обрання користувачем певного курсу).

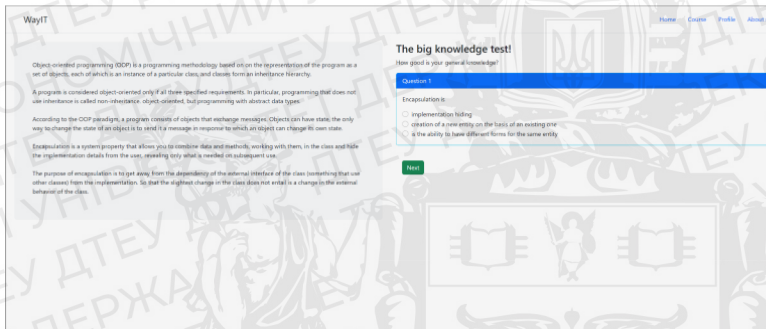
Основний доступ, який отримує користувач:

- сторінка «Course» вміщує в собі більше 20 різних тем, що можна опрацювати;

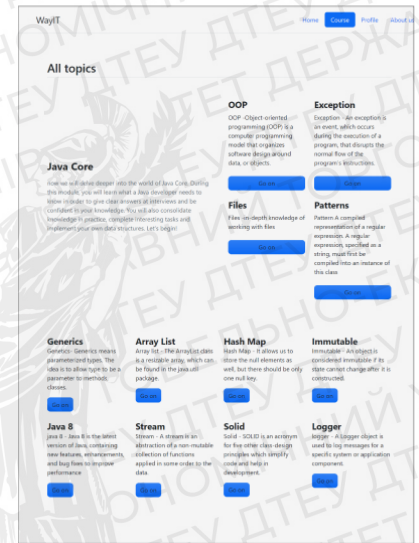
|     |       |         |        |      |  |                   |       |
|-----|-------|---------|--------|------|--|-------------------|-------|
|     |       |         |        |      |  | ДТЕУ 121 06-23.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата |  |                   | 33    |

- сторінка «Profile» дає змогу наповнити особистий кабінет інформацією про себе та вміщує статистику про загальний пройдений шлях по темах;
- сторінка «Material» стає доступний лише при обранні курсу та вміщує у собі теоретичну інформацію по обраній темі та тест для перевірки вивченого матеріалу.

### Materials page



### Course page



### Profile page

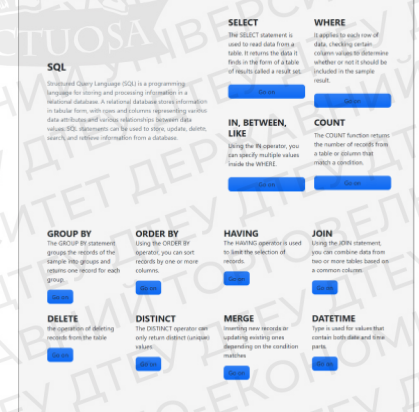
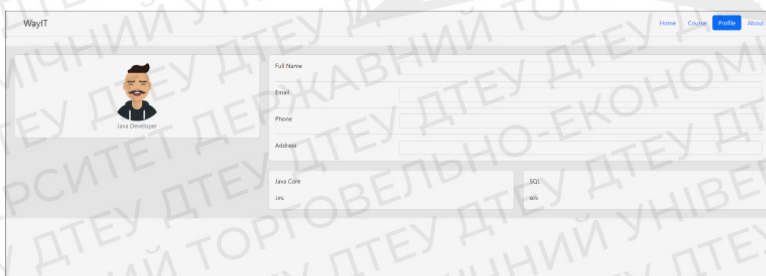


Рисунок 2.10. Сторінки до яких має доступ лише зареєстрований користувач

Джерело: побудовано автором в системі Figma (знімок екрану)

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      |                   | Аркуш |
|     |       |         |        |      |                   | 34    |
| Зм. | Аркуш | № докум | Підпис | Дата | ДТЕУ 121 06-23.БР |       |

## 2.4 Висновки до розділу 2

У процесі проектування програмного забезпечення було ідентифіковано, аналізовано та вирішено проблеми, пов'язані з вибором основних технологій, методологій, принципів, методів та інструментів. Для даного проекту був найбільш оптимальний вибір усіх інструментів розробки. Даний підхід убезпечить від складнощів підтримки продукту у майбутньому, дає можливість масштабувати та покращувати протягом довгого часу.

Побудовані UML-діаграми надають загальне уявлення про роботу додатку, його функціонування та важливі концепції, дозволяє аналізувати розробку через призму абстракції з дотриманням усіх фундаментальних практик програмування.

Ключову аспект у функціонуванні додатку відіграє саме база даних, логічна і фізична модель якої побудована з дотриманням усіх правил. Чітким визначенням усіх необхідних даних та їх типів, реалізація різних видів зв'язків між таблицями та нормалізацією їх, що гарантує вирішення проблем, які б могли викликати труднощі у майбутньому або унеможливлення ефективного та надійного зберігання даних.

У процесі роботи над зовнішнім виглядом додатку та аналізу досвіду користувача був розроблений макет платформи. Це забезпечить покращенню процесу розробки, адже маючи чіткий вигляд та опис інтерфейсу, робота на front-end частиною перестає бути абстрактною без чіткого бачення продукту в цілому.

|     |       |         |        |      |  |                   |       |
|-----|-------|---------|--------|------|--|-------------------|-------|
|     |       |         |        |      |  | ДТЕУ 121 06-23.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата |  |                   | 35    |



## РОЗДІЛ 3

### ПРОЦЕС РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 3.1 Розробка Back-End частини

Першим етапом початку розробки програмного забезпечення завжди є налаштування усіх систем та середовищ для подальшої роботи. Спираючись на технічне завдання можна виокремити основні інструменти, які будуть використовуватись під час написання платформи і потребують попередньої підготовки, а саме:

- встановлення системи управління базами даних MySQL та її налаштування;
- встановлення актуальної версії Maven, який необхідно для автоматизації великої кількості процесів, а також необхідний для налаштування фреймворку Spring Boot;
- встановлення сучасної версії JRE для можливості працювати з останньою версією фреймворку Spring Boot – 3.1.0.

Підготовка після встановлення СУБД MySQL полягає у створенні схеми у якій в свою чергу будуть зберігатись потрібні нам таблички. На етапі найменування схеми важливо вказати кодування, так як у подальшому ми будемо зберігати данні латиницею тому краще буде взяти стандартний тип кодування – utf8 і разом з ним у полі “Collation” обрати utf8\_general\_ci.

Перед тим як встановити Maven необхідно перевірити наявність встановленого JDK, що робиться за допомогою команди – «java --version» у терміналі (рис. 3.1).

|           |       |                 |        |          | <i>ДТЕУ 121 06-23.БР</i>                               |   |       |         |
|-----------|-------|-----------------|--------|----------|--|---|-------|---------|
| Зм.       | Аркуш | № докум.        | Підпис | Дата     |  |   |       |         |
| Зав. каф. |       | Криворучко О.В. |        | 14.04.23 | Платформа ІТ-академії на основі фреймворку Spring Boot | Стадія  | Аркуш | Аркушів |
| Керівник  |       | Десятко А. М    |        | 14.04.23 |  | РЗ  | 36    | 49      |
| Гарант    |       | Рзаєва С.Л.     |        | 14.04.23 |  | Факультет інформаційних технологій<br>4 курс, 6 група |       |         |
| Розробив  |       | Скрипник Д. О.  |        | 14.04.23 |  |   |       |         |
|           |       |                 |        |          | <i>Процес розробки програмного забезпечення</i>        |   |       |         |

```
C:\Users\skrip>java --version
java 11.0.14 2022-01-18 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.14+8-LTS-263)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.14+8-LTS-263, mixed mode)
```

*Рисунок 3.1. Приклад перевірки наявності необхідних інструментів*

*Джерело: побудовано автором в системі Terminal (знімок екрану)*

Якщо у результаті перевірки все необхідне встановлено, тоді наступним кроком буде завантаження Maven та додавання системних змінних для змінних середовища, а саме вказати шлях до папки bin для Maven у змінну Path, а також створити змінну «MAVEN\_HOME» зі шляхом щойно встановленого інструменту.

Встановлення сучасної версії JRE є надзвичайно важливим, адже для використання усього потенціалу Spring Boot, як у нашому випадку з останньою версією 3.0.1 необхідно враховувати, що для запуску програми потрібно мати версію JRE мінімум 17, так як в іншому випадку платформа не зможе навіть запуститися, адже випаде помилка яка свідчить про не сумісність версій.

Отже після налаштування необхідних систем також потрібно налаштувати і саме середовище. У першу чергу це додавання необхідних залежностей у файл pom.xml для отримання доступу до усіх бібліотек фреймворку. За рахунок того, що у залежностях Spring Boot реалізована вкладеність, таким чином не має потреби у їх великій кількості, адже вони в собі вже структуровано розподіленні між певними абстракціями, які ми в свою чергу і використовуємо.

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      |                   | Аркуш |
|     |       |         |        |      | ДТЕУ 121 06-23.БР | 37    |
| Зм. | Аркуш | № докум | Підпис | Дата |                   |       |

```

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.32</version>
  </dependency>
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <optional>true</optional>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
  </dependency>
</dependencies>

```

*Рисунок 3.2. Використання усіх необхідних залежностей у додатку*  
*Джерело: побудовано автором в системі IntelliJ IDEA (знімок екрану)*

Важливою складовою розробленого програмного додатку є його безпечність. У нашому випадку безпека побудована з використанням Spring Security та зміною його конфігурацій. Обов'язком кожного веб-розробника є захист особистих даних та забезпечення збереження їх від третіх осіб. Саме тому на платформі використовується система кодування паролю, що дозволяє стверджувати про захищеність користувача. У випадку використання вже влаштованого інструменту для кодування паролю з Spring Security потрібно лише створити метод з анотацією @Bean у межах класу SecurityConfig та задати довжину для об'єкту BCryptPasswordEncoder (рис. 3.3).

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      |                   | Аркуш |
|     |       |         |        |      |                   | 38    |
| Зм. | Аркуш | № докум | Підпис | Дата | ДТЕУ 121 06-23.БР |       |

```

@Bean
public PasswordEncoder passwordEncoder() {
    return new BCryptPasswordEncoder( strength: 8);
}

```

*Рисунок 3.3. Кодування паролю з допомогою BCryptPasswordEncoder  
Джерело: побудовано автором в системі Intelij IDEA (знімок екрану)*

Також з питання безпеки слід вархувувати можливість користувача мати доступ до заборонених для його перегляду частин сайту. Така фільтрація є обов'язковою частиною будь-якого захищеного додатку. У її основі лежить постійна перевірка користувача на доступ до тої чи іншої частини веб-додатку. Таким чином зацікавлена людина у отриманні заборонених для неї даних при спробі звернутись за будь-яким endpoint (кінцева точка зв'язку) проходить фільтрацію за певними параметрами. Якщо брати розроблювальну платформу ІТ академії то вона також містить постійну перевірку на можливість доступу до різних сторінок веб-додатку за параметром авторизації.

```

@Bean
protected SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
    http
        .authorizeHttpRequests((requests) -> requests
            .requestMatchers( ...patterns: "/", "/home", "/about-us", "/registration").permitAll()
            .anyRequest().authenticated()
        )
        .formLogin((form) -> form
            .loginPage("/login")
            .permitAll()
        )
        .logout(LogoutConfigurer::permitAll);

    return http.build();
}

```

*Рисунок 3.4. Перевірка автентифікації користувача  
Джерело: побудовано автором в системі Intelij IDEA (знімок екрану)*

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      |                   | Аркуш |
|     |       |         |        |      | ДТЕУ 121 06-23.БР | 39    |
| Зм. | Аркуш | № докум | Підпис | Дата |                   |       |

### 3.2 Розробка Front-End частини

Розробка інтерфейсу користувача займає важливу частину фінального результату продукту, так як ідеально прописану логіку застосунку можна оцінити лише люди які в цьому розуміються коли в свою чергу візуальна частина може поєднати між собою і можливість користування додатком звичайним користувачем, і в той ж момент оцінювати роботу на логікою.

При розробці інтерфейсу для платформи ІТ академії були використанні можливості бібліотеки Thymeleaf за допомогою якої вдалося поєднати .html код, Bootstrap бібліотеку для генерування стилів та логіку прописану мовою програмування Java на основі фреймворку Spring Boot.

Пропоную розглянути використання Thymeleaf на основі реєстрації користувача. За допомогою тегу форм в, який можна передати endpoint на якому буде відпуватися POST запит, який в свою чергу обробляється на рівні контролерів, а також за допомогою вище згаданої бібліотеки можна передати об'єкт який буде повернуто у результаті наповнення у тегах input за відповідними атрибутами даної сутності (рис. 3.6).

```
<form action="#" th:action="@{/registration}" th:object="${user}" method="POST">
  <div class="d-flex flex-row align-items-center mb-4">
    <i class="fas fa-envelope fa-lg me-3 fa-fw"></i>
    <div class="form-outline flex-fill mb-0">
      <input type="email" th:field="*{email}" id="form3Example3c" class="form-control"/>
      <label class="form-label" for="form3Example3c" style="margin-left: 0px;">Your Email</label>
      <div class="form-notch"><div class="form-notch-leading" style="width: 9px;"></div><div class="form-notch-middle" style="width: 64px;"></div><div class="form-notch-trailing"></div></div></div>
  </div>
  <div class="d-flex flex-row align-items-center mb-4">
    <i class="fas fa-lock fa-lg me-3 fa-fw"></i>
    <div class="form-outline flex-fill mb-0">
      <input type="password" th:field="*{password}" id="form3Example4c" class="form-control"/>
      <label class="form-label" for="form3Example4c" style="margin-left: 0px;">Password</label>
      <div class="form-notch"><div class="form-notch-leading" style="width: 9px;"></div><div class="form-notch-middle" style="width: 64px;"></div><div class="form-notch-trailing"></div></div></div>
  </div>
```

Рисунок 3.5. Приклад використання Thymeleaf

Джерело: побудовано автором в системі IntelliJ IDEA (знімок екрану)

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      |                   | Аркуш |
|     |       |         |        |      | ДТЕУ 121 06-23.БР | 40    |
| Зм. | Аркуш | № докум | Підпис | Дата |                   |       |

### 3.3 Процес тестування застосунку

Процес тестування платформи ІТ академії, є важливим етапом для успішної реалізації проекту. Саме під час тестування можна виявити дефекти, які можуть вплинути на різні аспекти продукту. До характеристики ми можемо віднести:

1.Функціональність ( Продукт повинен функціонувати відповідно до визначених попередньо вимог, та функцій, які є головними в процесі користування платформою. Тобто користуючись платформою ІТ академії, користувач повинен мати можливість використовувати весь функціональний багаж, не виявивши ні одного дефекту)

2.Надійність ( Ця характеристика в першу чергу говорить про те, що продукт повинен відповідати неявним, тобто нефункціональним вимогам. Тобто якісної окрім роботи функцій, важливими частинами тестування є дизайн, зручність користування та локалізація)

3.Стабільність ( При великому допустимому значенні навантаженості системи, весь її функціонал повинен працювати стабільно без багів та критичних дефектів.

4.Безпека (Дані користувача повинні бути захищеними. Особливості цієї характеристикою в першу чергу є особисті дані, такі як логін та пароль)

5.Ефективність (Одна із найважливіших частин, де головну роль грає наскільки ефективно використовуються ресурси продукту. Це наприклад технічний ресурс та бізнес ( чи буде прибутковим цей проект).

На початку процесу тестування важливо виділити основні цілі та методи тестування. До цілей відносимо:

- дослідження нереалізованих ризиків. Дуже важливо виявити дефекти на ранній стадіях розробки або тестування для запобігання майбутніх ризиків;
- вдосконалити якість продукту;

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      |                   | Аркуш |
|     |       |         |        |      |                   | 41    |
| Зм. | Аркуш | № докум | Підпис | Дата | ДТЕУ 121 06-23.БР |       |

- надання певної обгрунтованої інформації про результати процесу тестування;
- визначити чи готовий продукт до релізу за результатами тестування.

Щодо безпосередньо методів тестування. Для даного проєкту важливо використовувати наступні методи тестування: Unit testing, Integration testing, System testing, User-Acceptance testing, Testing with test cases, Black box testing, White box testing, Smoke testing, Regression testing, Usability testing.

Для реалізації наступного процесу тестування необхідно розробити тест кейси. Цей вид документації використовується дуже часто для детального описання того, як повинен працювати той чи інший модуль або функція. Для якісного та ефективного результату тестування, тест кейси складаються з таких елементів: назва тест кейсу, пріоритет, кроки для відтворення, очікуваний результат та Pass/Fail/Blocked/Untested.

Також під час етапу розробки тест кейсів, я використовував тест дизайн техніки. А саме : техніка еквівалентних класів та техніка граничних значень.

Використовуючи техніку еквівалентних класів під час тестування сторінки реєстрації, я зменшив у рази кількість тест кейсів. Розроблялось це за допомогою класів, на які ділиться діапазон допустимих та недопустимих значень та описується один випадок в одному класі. В результаті наші очікування зводяться до двох випадків : якщо тест кейс впаде, то і всі випадки в даному класі впадуть, якщо тест кейс не впаде то і всі інші обрані дані покажуть результат успішно пройденого тест кейсу.

Техніку граничних значень використовують частіше всього з технікою еквівалентних значень, бо якщо діапазон ділиться на класи, то є певні граничні значення, на яких дії системи змінюються. Найчастіше саме в таких місцях найбільша ймовірність знайти дефект. Саме тому як показано на (рис.3.6), я описав тест кейси за допомогою цих технік. Наприклад є

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      |                   | Аркуш |
|     |       |         |        |      | ДТЕУ 121 06-23.БР | 42    |
| Зм. | Аркуш | № докум | Підпис | Дата |                   |       |

діапазон, що користувач може зареєструватись, ввівши в поле пароль від 4 до 16 символів. Ми ділимо діапазон на три класи: перший – 1-3, другий 4-16, третій 17 і більше. Ввівши тільки кількість із другого класу, користувач зможе зареєструватись. Ці техніки забезпечують мінімальною кількістю створеними тест кейсами покрити більшість функціоналу платформи.

Для написання тест кейсів я використовував TestRail. Дуже зручний та розроблений саме для їх імплементації.

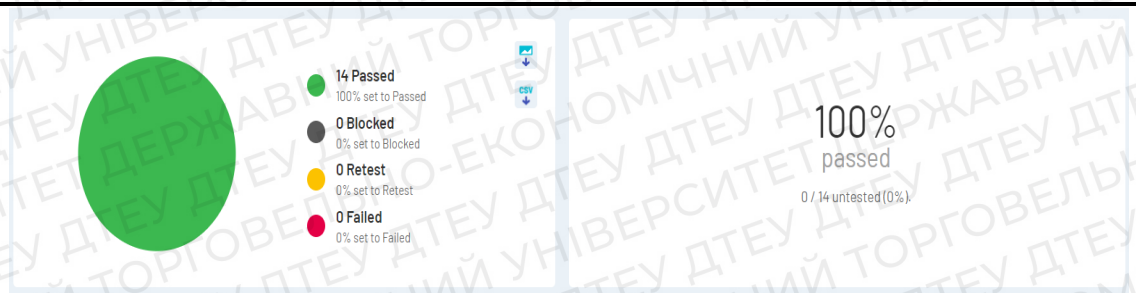
| ID  | Title  |
|-----|--|
| C1  | The password must contain at least one number and a special character  |
| C2  | The user can't create an account with non-unique email   |
| C3  | User must successfully login with already existing 'Email' and 'Password'                                    |
| C4  | User is able to register with one capital letter, 4-16 characters, one special symbol in 'Password' field    |
| C5  | User is not able to register with one capital letter, 0-3 characters, one special symbol in 'Password' field |
| C6  | User is not able to register with one capital letter, 16+ characters, one special symbol in 'Password' field |
| C7  | User is not able to register with no capital letter, 4-16 characters, one special symbol in 'Password' field |
| C8  | User is not able to register with one capital letter, 4-15 characters, no special symbol in 'Password' field |
| C9  | User is not able to register with 3 characters in 'Password' field   |
| C10 | User is able to register with 4 characters in 'Password' field   |
| C11 | User is able to register with 5 characters in 'Password' field   |
| C12 | User is able to register with 15 characters in 'Password' field  |
| C13 | User is able to register with 16 characters in 'Password' field  |
| C14 | User is not able to register with 17 characters in 'Password' field  |

*Рисунок 3.6. Тест кейси для сторінки реєстрації*

*Джерело: побудовано автором в системі TestRail (знімок екрану)*

Наступним етапом після написання тест кейсів це їхня перевірка. Зі сторони користувача, потрібно перевірити кожний кейс. В результаті чого створений Test Run для того щоб бачити точну кількість у вигляді діаграми і розуміти на скільки успішно проведено процес тестування.





*Рисунок 3.7. Результат перевірки на наявність дефектів у реєстрації*

*Джерело: побудовано автором в системі TestRail (знімок екрану)*

Отже, в результаті тестування, яке ми можемо побачити на рис 3.4 тестування пройдено успішно та якісно. 100% тест кейсів пройшли перевірку успішно, а це може означати, що платформа ІТ академії розроблена якісно та немає дефектів.

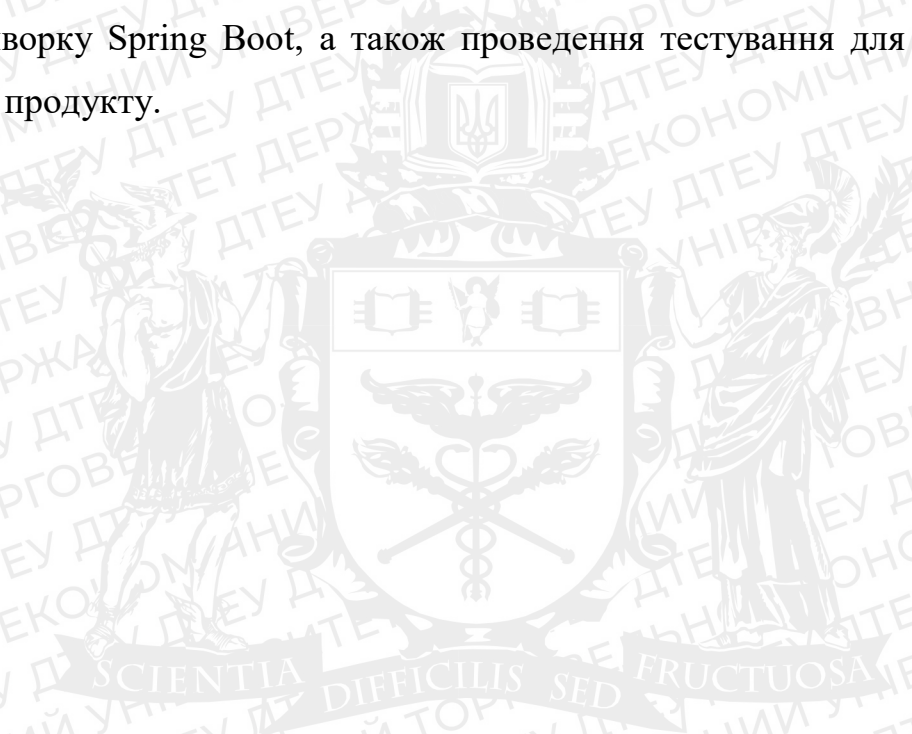
### 3.4 Висновки до розділу 3

Узагалі, висновок розділу 3 підкреслює важливість систематичного та добре організованого процесу розробки програмного забезпечення. Він включає в себе кроки від аналізу вимог і проектування до реалізації, тестування та впровадження. Використання певних технологій і фреймворків, таких як Spring Boot, сприяє прискоренню розробки та поліпшенню якості продукту.

Зважаючи на розроблений процес розробки програмного забезпечення для платформи ІТ академії, можна виокремити декілька ключових аспектів. По-перше, використання фреймворку Spring Boot для розробки як Back-End, так і Front-End частини платформи, дозволяє забезпечити швидкість розробки, ефективну роботу з базою даних та інтеграцію з іншими сервісами. Spring Boot забезпечує структурованість і модульність проекту, спрощує налаштування та розгортання, що робить його популярним в

|     |       |         |        |      |                   |       |
|-----|-------|---------|--------|------|-------------------|-------|
|     |       |         |        |      |                   | Аркуш |
|     |       |         |        |      | ДТЕУ 121 06-23.БР | 44    |
| Зм. | Аркуш | № докум | Підпис | Дата |                   |       |

розробці веб-додатків. По-друге, процес тестування є невід'ємною частиною розробки програмного забезпечення. Він допомагає виявити та виправити помилки, забезпечує якість та надійність продукту. Тестування проводилося на рівні Back-End і Front-End, що дозволяє перевірити як функціональність окремих компонентів, так і їх взаємодію. В результаті, розділ 3 надав огляд процесу розробки програмного забезпечення для платформи ІТ академії. Цей процес включав в себе розробку Back-End та Front-End частини на основі фреймворку Spring Boot, а також проведення тестування для забезпечення якості продукту.



|     |       |         |        |      |  |                   |       |
|-----|-------|---------|--------|------|--|-------------------|-------|
|     |       |         |        |      |  | ДТЕУ 121 06-23.БР | Аркуш |
| Зм. | Аркуш | № докум | Підпис | Дата |  |                   | 45    |

## ВИСНОВКИ ТА ПРОПОЗИЦІЇ

У результаті виконання випускної кваліфікаційної роботи було проведено покращення власних навичок, детально визначено та досліджено ринок ІТ галузі та її масштабування з часом, проведено аналіз основних проблем інших платформ, та визначення найкращих технологій на ринку, що дозволило написати платформу ІТ-академії.

При дослідженні проблем розробки були визначенні основні вимоги до програмного додатку, від абстрактних понять розробки до конкретного використання певних інструментів.

У ході роботи було досліджено предметну область, розробка та впровадження платформи ІТ академії, яка базується на сучасних технологіях. Та прийняті правильні конструктивні рішення, що забезпечило запобігання подовження процесу розробки, впевненість у конкурентно спроможності на ринку та побудови стратегій на майбутній розвиток платформи.

Для побудови візуальною частиною додатку – було визначено чіткі план майбутнього додатка, кольорова гамма та розроблено макет інтерфейсу у системі Figma, для того, щоб додаток був орієнтований на користувача.

Під час вибору мови програмування та середовища розробки було аналізовано фреймворк – Spring Boot, визначено ризики такі як несумісність версій порівняли з його попередником Spring, який в свою чергу також є потужним фреймворком, але функціонал, та поєднання такої кількості корисних інструментів у одній сутності повністю затьмарило Spring.

Для забезпечення якості було проведено тестування програмного забезпечення, а саме системи реєстрація, котра була покрита тест кейсами

| Зм.       | Аркуш | № докум.        | Підпис | Дата     | ДТЕУ 121 06-23.БР                                      |                                    |       |         |
|-----------|-------|-----------------|--------|----------|--|------------------------------------|-------|---------|
| Зав. каф. |       | Криворучко О.В. |        | 28.04.23 | Платформа ІТ-академії на основі фреймворку Spring Boot | Стадія                             | Аркуш | Аркушів |
| Керівник  |       | Десятко А. М    |        | 28.04.23 |  | ВП                                 | 46    | 49      |
| Гарант    |       | Рзаєва С.Л.     |        | 28.04.23 |  | Факультет інформаційних технологій |       |         |
| Розробив  |       | Скрипник Д. О.  |        | 28.04.23 |  | 4 курс, 6 група                    |       |         |
|           |       |                 |        |          | Висновки та пропозиції                                 |                                    |       |         |

та за результатами яких зясувалось, що один із найважливіших модулів працює коректно.

Пропозиції до проєкту:

- програмний додаток може бути легко масштабований у напрямку розширення кількості технологій для вивчення. Необхідно зробити аналіз ринку та додати ще одну мову програмування.
- програмний додаток може бути адаптований на різні країни, тобто бути локалізованим.
- проведення збільшення складу команди підтримки платформи, за для швидшого розвитку та досягнення нових цілей.

Завдання даної роботи можна вважати виконаним, адже в результаті розробки ми отримали повністю функціонуючий, протестований з чітким архітектурним планом програмний додаток.

|     |       |         |        |      |                   |  |  |       |
|-----|-------|---------|--------|------|-------------------|--|--|-------|
|     |       |         |        |      |                   |  |  | Аркуш |
|     |       |         |        |      |                   |  |  | 47    |
| Зм. | Аркуш | № докум | Підпис | Дата | ДТЕУ 121 06-23.БР |  |  |       |

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Чому український ІТ-сектор став таким популярним [Електронний ресурс]. Режим доступу: <http://surl.li/heuvu> (дата звертання\_22.01.2023)
2. Інфографіка: українська ІТ-галузь за 2021 рік — в цифрах [Електронний ресурс]. Режим доступу: <http://surl.li/heuxz> (дата звертання 23.01.2023)
3. Основні поняття об'єктно-орієнтованого програмування [Електронний ресурс]. Режим доступу: <http://surl.li/heuyb> (дата звертання\_25.01.2023)
4. Об'єктно-орієнтоване програмування [Електронний ресурс]. Режим доступу: <http://surl.li/hevdj> (дата звертання 27.01.2023)
5. На якій мові ви пишете для роботи [Електронний ресурс]. Режим доступу: <http://surl.li/hevdn> (дата звертання 12.02.2023)
6. Чому Java – найпопулярніша мова програмування у світі [Електронний ресурс]. Режим доступу: <http://surl.li/hevdp> (дата звертання 14.04.2023)
7. Мови програмування за сферами використання [Електронний ресурс]. Режим доступу: <http://surl.li/hevdq> (дата звертання 29.04.2023)
8. Програмування для чайників [Електронний ресурс]. Режим доступу: <http://surl.li/hevds> (дата звертання 27.02.2023)
9. Java JDK, JRE та JVM [Електронний ресурс]. Режим доступу: <http://surl.li/hevdw> (дата звертання 18.02.2023)
10. Популярні фреймворки Java: Spring та Hibernate [Електронний ресурс]. Режим доступу: <http://surl.li/hevdy> (дата звертання\_28.02.2023)
11. Коротко про UML [Електронний ресурс]. Режим доступу: <http://surl.li/hevec> (дата звертання\_01.03.2023)

|            |              |                 |               |             |  |  |              |                |
|------------|--------------|-----------------|---------------|-------------|--|--|--------------|----------------|
|            |              |                 |               |             | <i>ДТЕУ 121 06-23.БР</i>                                     |  |              |                |
| <i>Зм.</i> | <i>Аркуш</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | Платформа ІТ-академії на<br>основі фреймворку Spring<br>Boot | <i>Стадія</i>  | <i>Аркуш</i> | <i>Аркушів</i> |
| Зав. каф.  |              | Криворучко О.В. |               | 23.12.22    |  | <i>СВД</i>   | 48           | 48             |
| Керівник   |              | Десятко А. М    |               | 23.12.22    |  | Факультет інформаційних<br>технологій<br>4 курс, 6 група |              |                |
| Гарант     |              | Рзаєва С.Л.     |               | 23.12.22    |  |  |              |                |
| Розробив   |              | Скрипник Д. О.  |               | 23.12.22    |  |  |              |                |
|            |              |                 |               |             | <i>Список використаних джерел</i>                            |  |              |                |

12. MVC [Електронний ресурс]. Режим доступу: <http://surl.li/rxau> (дата звертання 01.03.2023)



|     |       |         |        |      |  |                   |       |
|-----|-------|---------|--------|------|--|-------------------|-------|
|     |       |         |        |      |  |                   | Аркуш |
|     |       |         |        |      |  | ДТЕУ 121 06-23.БР | 49    |
| Зм. | Аркуш | № докум | Підпис | Дата |  |                   |       |

## ДОДАТКИ

### ДОДАТОК А

```
<!DOCTYPE HTML>
<html>
<head>
<title>Course</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
</head>
<main>
<body>
<header class="d-flex flex-wrap justify-content-center py-3 mb-4 border-bottom">
<a href="/" class="d-flex align-items-center mb-3 mb-md-0 me-md-auto text-dark text-decoration-none">
<svg class="bi me-2" width="40" height="32">
<use xlink:href="#bootstrap"></use>
</svg>
<span class="fs-4">WayIT</span>
</a>
<ul class="nav nav-pills">
<li class="nav-item"><a href="/" class="nav-link">Home</a></li>
<li class="nav-item"><a href="/courses" class="nav-link">Course</a></li>
<li class="nav-item"><a href="/profile" class="nav-link">Profile</a></li>
<li class="nav-item"><a href="/about-us" class="nav-link active" aria-current="page">About us</a></li>
</ul>
</header>
<div class="container">
<div class="p-5 mb-4 bg-light rounded-3">
<h1 class="display-5 fw-bold">Welcome to our IT courses website!</h1>
<p class="col-md-8 fs-4">
We offer a wide range of courses from various branches of the IT industry. Our courses are designed
highly qualified professionals with many years of experience in the field of information technologies.
```

## ПРОДОВЖЕННЯ ДОДАТКУ А

Our courses will help you gain the practical skills and knowledge you need to develop your career in IT sphere. We offer courses for beginners and experienced professionals who want to deepen their knowledge and gain new skills.</p>

<p class="col-md-8 fs-4">Our courses include such topics as programming, databases, networking technology, information security, web development and many others. We update our app regularly so that to provide our students with the most relevant knowledge and skills.

Our teachers are highly qualified specialists with extensive experience in the IT industry. They will help you understand complex concepts and will answer your questions as you learn.

We offer flexible scheduling to help you fit your studies around your schedule. Our courses available online, allowing you to study from anywhere with internet access.

In addition, we offer the possibility of internship in large partner companies, which allows our students to get practical experience in a real project.

Our goal is to help our students develop their knowledge and skills in the IT industry. we believe what

Anyone can learn programming and develop their skills, regardless of whether they have experience in IT industry or not.</p>

<p class="col-md-8 fs-4">Our courses are designed taking into account the most modern technologies and teaching methods.

We use interactive teaching methods that allow our students to gain practical experience and develop skills in real projects.

Our courses are available to anyone who wants to develop in the IT field. We offer courses for students who just starting their journey in the IT field, as well as for experienced professionals who want to deepen their knowledge and gain new skills.

Our teachers are professionals with extensive experience in the field of IT. They know how to help our students learn new knowledge and skills and are ready to help you at every step of your studies.</p>

<p class="col-md-8 fs-4">We believe that learning should be accessible and flexible. That's why our courses are affordable online, which allows you to study from anywhere you have access to the Internet and a flexible schedule allows you agree



## ПРОДОВЖЕННЯ ДОДАТКУ А

study with your schedule.

Thanks to our experience and expertise, we know what is important for a successful IT education. That's why we provide our students support and mentoring throughout the entire learning process.</p>

</div>

</div>

</body>

</main>

</html>



```

<!DOCTYPE HTML>
<html>
<head>
<title>Home</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
</head>
<body>
<header class="d-flex flex-wrap justify-content-center py-3 mb-4 border-bottom">
<a href="/" class="d-flex align-items-center mb-3 mb-md-0 me-md-auto text-dark text-decoration-none">
<svg class="bi me-2" width="40" height="32">
<use xlink:href="#bootstrap"></use>
</svg>
<span class="fs-4">WayIT</span>
</a>
<ul class="nav nav-pills">
<li class="nav-item"><a href="/" class="nav-link active" aria-current="page">Home</a></li>
<li class="nav-item"><a href="/courses" class="nav-link">Course</a></li>
<li class="nav-item"><a href="/profile" class="nav-link">Profile</a></li>
<li class="nav-item"><a href="/about-us" class="nav-link">About us</a></li>
</ul>
</header>
<main>
<div class="container col-xxl-8 px-4 py-5">
<div class="row flex-lg-row-reverse align-items-center g-5 py-5">
<div class="col-10 col-sm-8 col-lg-6">

</div>
<div class="col-lg-6">
<h1 class="display-5 fw-bold lh-1 mb-3">Join us!</h1>
<p class="lead">Welcome to our IT Academy. To start or continue training, you need to register or log in

```

## ПРОДОВЖЕННЯ ДОДАТКУ Б

on our company's website</p>

```
<div class="d-grid gap-2 d-md-flex justify-content-md-start">
```

```
<button type="button" class="btn btn-primary btn-lg px-4 me-md-2"><a href="/login"
```

```
class="nav-link">Login</a>
```

```
</button>
```

```
<button type="button" class="btn btn-outline-secondary btn-lg px-4"><a href="/registration"
```

```
class="nav-link">Register</a>
```

```
</button>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="container pt-3">
```

```
<div class="row">
```

```
<div class="col-lg-4">
```

```

```

```
<h2 class="fw-normal">SEO</h2>
```

```
</div>
```

```
</div>
```

```
<hr class="featurette-divider">
```

```
<div class="row featurette">
```

```
<div class="col-md-7">
```

```
<h2 class="featurette-heading fw-normal lh-1">Why Java</h2>
```

```
<p class="lead">The Java programming language has many advantages that make it a popular choice for many  
programmers and companies. Here are some of the reasons why Java is one of the most popular  
programming languages in the world:</p>
```

```
<ul class="list-group list-group-flush">
```

```
<li class="list-group-item">Platform independence</li>
```

```
<li class="list-group-item">A large community of developers</li>
```

```
<li class="list-group-item">Security</li>
```

```
<li class="list-group-item">High productivity</li>
```

```
<li class="list-group-item">Wide possibilities</li>
```

```
<li class="list-group-item">Object-oriented approach</li>
```

## ПРОДОВЖЕННЯ ДОДАТКУ Б

```
<li class="list-group-item">Ease of learning</li>
```

```
</ul>
```

```
</div>
```

```
<div class="col-md-5 order-md-1">
```

```

```

```
</div>
```

```
</div>
```

```
<hr class="featurette-divider">
```

```
<div class="row featurette">
```

```
<div class="col-md-7 order-md-2">
```

```
<h2 class="featurette-heading fw-normal lh-1">What we offer</h2>
```

```
<ul class="list-group list-group-flush">
```

```
<li class="list-group-item">Quality material</li>
```

```
<li class="list-group-item">Latest technologies</li>
```

```
<li class="list-group-item">Fast growth</li>
```

```
<li class="list-group-item">Interesting tasks</li>
```

```
<li class="list-group-item">Free schedule</li>
```

```
</ul>
```

```
</div>
```

```
<div class="col-md-5 order-md-1">
```

```

```

```
</div>
```

```
</div>
```

```
<hr class="featurette-divider">
```

```
<div class="row featurette">
```

```
<div class="col-md-7">
```

```
<h2 class="featurette-heading fw-normal lh-1">Why is working in IT so promising?</h2>
```

```
<ul class="list-group list-group-flush">
```

```
<li class="list-group-item">Great growing demand for specialists worldwide</li>
```

```
<li class="list-group-item">Solid career foundation for decades</li>
```

## ПРОДОВЖЕННЯ ДОДАТКУ Б

```
<li class="list-group-item">High decent wages</li>
<li class="list-group-item">Ability to work anywhere in the world</li>
<li class="list-group-item">Free schedule</li>
<li class="list-group-item">Social packages and other bonuses from companies</li>
</ul>
</div>
<div class="col-md-5">
  
</div>
</div>
<hr class="featurette-divider">
</div>
<footer class="container">
  <p class="float-end"><a href="#">Back to top</a></p>
</footer>
</main>
</body>
</html>
```



```

<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<title>Course</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
</head>
<body>
<header class="d-flex flex-wrap justify-content-center py-3 mb-4 border-bottom">
<a href="/" class="d-flex align-items-center mb-3 mb-md-0 me-md-auto text-dark text-decoration-none">
<svg class="bi me-2" width="40" height="32"><use xlink:href="#bootstrap"></use></svg>
<span class="fs-4">WayIT</span>
</a>
<ul class="nav nav-pills">
<li class="nav-item"><a href="/" class="nav-link">Home</a></li>
<li class="nav-item"><a href="/courses" class="nav-link active" aria-current="page">Course</a></li>
<li class="nav-item"><a href="/profile" class="nav-link">Profile</a></li>
<li class="nav-item"><a href="/about-us" class="nav-link">About us</a></li>
</ul>
</header>
<main>
<div class="container px-4 py-5">
<h2 class="pb-2 border-bottom">All topics</h2>
<div class="row row-cols-1 row-cols-md-2 align-items-md-center g-5 py-5">
<div class="col d-flex flex-column align-items-start gap-2">
<h3 class="fw-bold">Java Core</h3>
<p class="text-muted">now we will delve deeper into the world of Java Core. During this module, you will learn what a Java developer needs to know in order to give clear answers at interviews and be confident in your knowledge. You will also consolidate knowledge in practice, complete interesting tasks and implement your own data structures. Let's begin!</p>
</div>
<div class="col">

```

## ПРОДОВЖЕННЯ ДОДАТКУ В

```
<div class="row row-cols-1 row-cols-sm-2 g-4">
  <div class="col d-flex flex-column gap-2">
    <h3 class="fw-bold mb-0 fs-4">OOP</h3>
    <p>OOP -Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or
objects.</p>
    <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-
3">
      <a href="#" class="btn">Go on</a>
    </div>
  </div>
  <div class="col d-flex flex-column gap-2">
    <h3 class="fw-bold mb-0 fs-4">Exception</h3>
    <p>Exception - An exception is an event, which occurs during the execution of a program, that disrupts the normal flow of the
program's instructions.</p>
    <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-
3">
      <a href="#" class="btn">Go on</a>
    </div>
  </div>
  <div class="col d-flex flex-column gap-2">
    <h3 class="fw-bold mb-0 fs-4">Files</h3>
    <p>Files -in-depth knowledge of working with files</p>
    <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-
3">
      <a href="#" class="btn">Go on</a>
    </div>
  </div>
  <div class="col d-flex flex-column gap-2">
    <h3 class="fw-bold mb-0 fs-4">Patterns</h3>
    <p>Pattern
      A compiled representation of a regular expression. A regular expression, specified as a string, must first be compiled into an
instance of this class</p>
    <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-
3">
```

## ПРОДОВЖЕННЯ ДОДАТКУ В

```
<a href="#" class="btn">Go on</a>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="row row-cols-1 row-cols-sm-2 row-cols-md-3 row-cols-lg-4 g-4 py-5">
  <div class="col d-flex align-items-start">
    <div>
      <h3 class="fw-bold mb-0 fs-4">Generics</h3>
      <p>Generics- Generics means parameterized types. The idea is to allow type to be a parameter to methods, classes.</p>
      <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
        <a href="#" class="btn">Go on</a>
      </div>
    </div>
  </div>
  <div class="col d-flex align-items-start">
    <div>
      <h3 class="fw-bold mb-0 fs-4">Array List</h3>
      <p>Array list - The ArrayList class is a resizable array, which can be found in the java.util package.</p>
      <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
        <a href="#" class="btn">Go on</a>
      </div>
    </div>
  </div>
  <div class="col d-flex align-items-start">
    <div>
      <h3 class="fw-bold mb-0 fs-4">Hash Map</h3>
      <p>Hash Map - It allows us to store the null elements as well, but there should be only one null key.</p>
      <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
        <a href="#" class="btn">Go on</a>
      </div>
    </div>
  </div>
</div>
```



## ПРОДОВЖЕННЯ ДОДАТКУ В

```
<div class="col d-flex align-items-start">
  <div>
    <h3 class="fw-bold mb-0 fs-4">Immutable</h3>
    <p>Immutable - An object is considered immutable if its state cannot change after it is constructed.</p>
    <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
      <a href="#" class="btn">Go on</a>
    </div>
  </div>
</div>
<div class="col d-flex align-items-start">
  <div>
    <h3 class="fw-bold mb-0 fs-4">Java 8</h3>
    <p>java 8 - Java 8 is the latest version of Java, containing new features, enhancements, and bug fixes to improve performance.</p>
    <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
      <a href="#" class="btn">Go on</a>
    </div>
  </div>
</div>
<div class="col d-flex align-items-start">
  <div>
    <h3 class="fw-bold mb-0 fs-4">Stream</h3>
    <p>Stream - A stream is an abstraction of a non-mutable collection of functions applied in some order to the data.</p>
    <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
      <a href="#" class="btn">Go on</a>
    </div>
  </div>
</div>
<div class="col d-flex align-items-start">
  <div>
    <h3 class="fw-bold mb-0 fs-4">Solid</h3>
    <p>Solid - SOLID is an acronym for five other class-design principles which simplify code and help in development.</p>
    <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
      <a href="#" class="btn">Go on</a>
    </div>
  </div>
</div>
```

## ПРОДОВЖЕННЯ ДОДАТКУ В

```
</div>
</div>
<div class="col d-flex align-items-start">
  <div>
    <h3 class="fw-bold mb-0 fs-4">Logger</h3>
    <p>logger - A Logger object is used to log messages for a specific system or application component.</p>
    <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
      <a href="#" class="btn">Go on</a>
    </div>
  </div>
</div>
</div>
</div>
</div>
</div>
<hr class="my-4">
<div class="container px-4 py-5">
  <div class="row row-cols-1 row-cols-md-2 align-items-md-center g-5 py-5">
    <div class="col d-flex flex-column align-items-start gap-2">
      <h3 class="fw-bold">SQL</h3>
      <p class="text-muted">Structured Query Language (SQL) is a programming language for storing and processing information in a relational database. A relational database stores information in tabular form, with rows and columns representing various data attributes and various relationships between data values. SQL statements can be used to store, update, delete, search, and retrieve information from a database.</p>
    </div>
    <div class="col">
      <div class="row row-cols-1 row-cols-sm-2 g-4">
        <div class="col d-flex flex-column gap-2">
          <h3 class="fw-bold mb-0 fs-4">SELECT</h3>
          <p>The SELECT statement is used to read data from a table. It returns the data it finds in the form of a table of results called a result set.</p>
        </div>
        <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
          <a href="#" class="btn">Go on</a>
        </div>
      </div>
    </div>
  </div>
</div>
```

## ПРОДОВЖЕННЯ ДОДАТКУ В

```
</div>
<div class="col d-flex flex-column gap-2">
  <h3 class="fw-bold mb-0 fs-4">WHERE</h3>
  <p>It applies to each row of data, checking certain column values to determine whether or not it should be included in the sample result.</p>
  <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
    <a href="#" class="btn">Go on</a>
  </div>
</div>
<div class="col d-flex flex-column gap-2">
  <h3 class="fw-bold mb-0 fs-4">IN, BETWEEN, LIKE</h3>
  <p>Using the IN operator, you can specify multiple values inside the WHERE.</p>
  <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
    <a href="#" class="btn">Go on</a>
  </div>
</div>
<div class="col d-flex flex-column gap-2">
  <h3 class="fw-bold mb-0 fs-4">COUNT</h3>
  <p>The COUNT function returns the number of records from a table or column that match a condition.</p>
  <div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
    <a href="#" class="btn">Go on</a>
  </div>
</div>
</div>
<div class="row row-cols-1 row-cols-sm-2 row-cols-md-3 row-cols-lg-4 g-4 py-5">
  <div class="col d-flex align-items-start">
    <div>
      <h3 class="fw-bold mb-0 fs-4">GROUP BY</h3>
```

## ПРОДОВЖЕННЯ ДОДАТКУ В

<p>The GROUP BY statement groups the records of the sample into groups and returns one record for each group.</p>

```
<div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
```

```
<a href="#" class="btn">Go on</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="col d-flex align-items-start">
```

```
<div>
```

```
<h3 class="fw-bold mb-0 fs-4">ORDER BY</h3>
```

<p>Using the ORDER BY operator, you can sort records by one or more columns.</p>

```
<div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
```

```
<a href="#" class="btn">Go on</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="col d-flex align-items-start">
```

```
<div>
```

```
<h3 class="fw-bold mb-0 fs-4">HAVING</h3>
```

<p>The HAVING operator is used to limit the selection of records.</p>

```
<div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
```

```
<a href="#" class="btn">Go on</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="col d-flex align-items-start">
```

```
<div>
```

```
<h3 class="fw-bold mb-0 fs-4">JOIN</h3>
```

<p>Using the JOIN statement, you can combine data from two or more tables based on a common column.</p>

```
<div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
```

```
<a href="#" class="btn">Go on</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="col d-flex align-items-start">
```

## ПРОДОВЖЕННЯ ДОДАТКУ В

```
<div>
<h3 class="fw-bold mb-0 fs-4">DELETE</h3>
<p>the operation of deleting records from the table</p>
<div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
<a href="#" class="btn">Go on</a>
</div>
</div>
<div class="col d-flex align-items-start">
<div>
<h3 class="fw-bold mb-0 fs-4">DISTINCT</h3>
<p>The DISTINCT operator can only return distinct (unique) values.</p>
<div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
<a href="#" class="btn">Go on</a>
</div>
</div>
</div>
<div class="col d-flex align-items-start">
<div>
<h3 class="fw-bold mb-0 fs-4">MERGE</h3>
<p>Inserting new records or updating existing ones depending on the condition matches</p>
<div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
<a href="#" class="btn">Go on</a>
</div>
</div>
</div>
<div class="col d-flex align-items-start">
<div>
<h3 class="fw-bold mb-0 fs-4">DATETIME</h3>
<p>Type is used for values that contain both date and time parts.</p>
<div class="feature-icon-small d-inline-flex align-items-center justify-content-center text-bg-primary bg-gradient fs-4 rounded-3">
<a href="#" class="btn">Go on</a>
</div>
</div>
</div>
```

```
</div>
</div>
</div>
</main>
</body>
</html>
```



```

<!DOCTYPE HTML>
<html xmlns:th="http://www.w3.org/1999/xhtml">
<head>
<title>Course</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
</head>
<main>
<body>
<header class="d-flex flex-wrap justify-content-center py-3 mb-4 border-bottom">
<a href="/" class="d-flex align-items-center mb-3 mb-md-0 me-md-auto text-dark text-decoration-none">
<svg class="bi me-2" width="40" height="32">
<use xlink:href="#bootstrap"></use>
</svg>
<span class="fs-4">WayIT</span>
</a>
<ul class="nav nav-pills">
<li class="nav-item"><a href="/" class="nav-link">Home</a></li>
<li class="nav-item"><a href="/courses" class="nav-link">Course</a></li>
<li class="nav-item"><a href="/profile" class="nav-link">Profile</a></li>
<li class="nav-item"><a href="/about-us" class="nav-link">About us</a></li>
</ul>
</header>
<div class="row d-flex justify-content-center">
<div class="col-12 col-md-8 col-lg-6 col-xl-6">
<div class="card shadow-2-strong" style="border-radius: 1rem;">
<div class="card-body p-5 text-center">
<h3 class="mb-5">Sign in</h3>
<form action="#" th:action="@{/login}" th:object="{user}" method="POST">
<div class="form-outline mb-4">
<input type="text" th:field="*{username}" id="typeEmailX-2"
class="form-control form-control-lg">

```

## ПРОДОВЖЕННЯ ДОДАТКУ Г

```
<label class="form-label" for="typeEmailX-2" style="margin-left: 0px;">Email</label>
<div class="form-notch">
  <div class="form-notch-leading" style="width: 9px;"></div>
  <div class="form-notch-middle" style="width: 40px;"></div>
  <div class="form-notch-trailing"></div>
</div>
<div>
  <div class="form-outline mb-4">
    <input type="password" th:field="*{password}" id="typePasswordX-2"
      class="form-control form-control-lg">
    <label class="form-label" for="typePasswordX-2" style="margin-left: 0px;">Password</label>
    <div class="form-notch">
      <div class="form-notch-leading" style="width: 9px;"></div>
      <div class="form-notch-middle" style="width: 64px;"></div>
      <div class="form-notch-trailing"></div>
    </div>
    <div>
      <input class="btn btn-primary btn-lg btn-block" value="Login" type="submit">
    </form>
  </div>
</div>
</div>
</div>
</div>
</body>
</main>
</html>
```



```

<!DOCTYPE HTML>
<html>
<head>
<title>Home</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
</head>
<body>
<header class="d-flex flex-wrap justify-content-center py-3 mb-4 border-bottom">
<a href="/" class="d-flex align-items-center mb-3 mb-md-0 me-md-auto text-dark text-decoration-none">
<svg class="bi me-2" width="40" height="32">
<use xlink:href="#bootstrap"></use>
</svg>
<span class="fs-4">WayIT</span>
</a>
<ul class="nav nav-pills">
<li class="nav-item"><a href="/" class="nav-link">Home</a></li>
<li class="nav-item"><a href="/courses" class="nav-link">Course</a></li>
<li class="nav-item"><a href="/profile" class="nav-link">Profile</a></li>
<li class="nav-item"><a href="/about-us" class="nav-link">About us</a></li>
</ul>
</header>
<main>
<div class="container-fluid">
<div class="row">
<div class="col">
<div class="container">
<div class="p-5 mb-4 bg-light rounded-3">
<p>
Object-oriented programming (OOP) is a programming methodology based on
the representation of the program as a set of objects, each of which is
an instance of a particular class, and classes form an inheritance hierarchy.

```

## ПРОДОВЖЕННЯ ДОДАТКУ Д

<br><br>A program is considered object-oriented only if all three

specified

requirements. In particular, programming that does not use inheritance is called non-inheritance.

object-oriented, but programming with abstract data types.

<br><br>According to the OOP paradigm, a program consists of objects that exchange messages.

Objects

can have state, the only way to change the state of an object is to send it

a message in response to which an object can change its own state.

<br><br>Encapsulation is a system property that allows you to combine data and methods,

working with

them, in the class and hide the implementation details from the user, revealing only what is needed

on subsequent use.

<br><br>The purpose of encapsulation is to get away from the dependency of the external interface of the class (something that

use other classes) from the implementation. So that the slightest change in the class does not entail

is a change in the external behavior of the class.</p>

</div>

</div>

</div>

<div class="col">

<div class="jumbotron">

<h3>The big knowledge test!</h3>

<p>How good is your general knowledge?</p>

</div>

<div class="card border-info mb-4">

<div class="d-flex justify-content-between align-items-center card-header bg-primary text-white">

<span>Question 1</span>

</div>

<div id="q1" class="collapse show" aria-labelledby="h1">

<div class="card-body">

<p>Encapsulation is</p>

<div class="form-check">

## ПРОДОВЖЕННЯ ДОДАТКУ Д

```
<input class="form-check-input" type="radio" name="q1" id="q1_r1" value="r1">
  <label class="form-check-label" for="q1_r1">
    implementation hiding
  </label>
</div>

<div class="form-check">
  <input class="form-check-input" type="radio" name="q1" id="q1_r2" value="r2">
  <label class="form-check-label" for="q1_r2">
    creation of a new entity on the basis of an existing one
  </label>
</div>

<div class="form-check">
  <input class="form-check-input" type="radio" name="q1" id="q1_r3" value="r3">
  <label class="form-check-label" for="q1_r3">
    is the ability to have different forms for the same entity
  </label>
</div>
</div>
</div>
<div class="d-flex flex-row justify-content-between align-items-center p-3 bg-white">
  <button class="btn btn-primary border-success align-items-center btn-success" type="button">Next<
    class="fa fa-angle-right ml-2"></i></button>
</div>
</div>
</div>
</div>
</div>
</main>
</body>
</html>
```

```

<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<title>Course</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
</head>
<body>
<header class="d-flex flex-wrap justify-content-center py-3 mb-4 border-bottom">
<a href="/" class="d-flex align-items-center mb-3 mb-md-0 me-md-auto text-dark text-decoration-none">
<svg class="bi me-2" width="40" height="32">
<use xlink:href="#bootstrap"></use>
</svg>
<span class="fs-4">WayIT</span>
</a>
<ul class="nav nav-pills">
<li class="nav-item"><a href="/" class="nav-link">Home</a></li>
<li class="nav-item"><a href="/courses" class="nav-link">Course</a></li>
<li class="nav-item"><a href="/profile" class="nav-link active" aria-current="page">Profile</a></li>
<li class="nav-item"><a href="/about-us" class="nav-link">About us</a></li>
</ul>
</header>
<main>
<section class="w-100 p-4" style="background-color: #eee; border-radius: .5rem .5rem 0 0;">
<form action="#" th:action="@{/profile/update}" th:object="{profile}" method="POST">
<div class="row">
<div class="col-lg-4">
<div class="card mb-4">
<div class="card-body text-center">

<p class="text-muted mb-1">Java Developer</p>

```

## ПРОДОВЖЕННЯ ДОДАТКУ Е

```
</div>
</div>
</div>
<div class="col-lg-8">
  <div class="card mb-4">
    <div class="card-body">
      <div class="row">
        <div class="col-sm-3">
          <p class="mb-0">Full Name</p>
        </div>
        <div class="col-sm-9">
          <input type="text" th:field="{first_name}" class="form-control"/>
        </div>
      </div>
      <hr>
      <div class="row">
        <div class="col-sm-3">
          <p class="mb-0">Email</p>
        </div>
        <div class="col-sm-9">
          <input type="text" th:field="{email}" class="form-control"/>
        </div>
      </div>
      <hr>
      <div class="row">
        <div class="col-sm-3">
          <p class="mb-0">Phone</p>
        </div>
        <div class="col-sm-9">
          <input type="text" th:field="{phone}" class="form-control"/>
        </div>
      </div>
      <hr>
      <div class="row">
```



```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<title>Course</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
</head>
<body>
<header class="d-flex flex-wrap justify-content-center py-3 mb-4 border-bottom">
<a href="/" class="d-flex align-items-center mb-3 mb-md-0 me-md-auto text-dark text-decoration-none">
<svg class="bi me-2" width="40" height="32"><use xlink:href="#bootstrap"></use></svg>
<span class="fs-4">WayIT</span>
</a>
<ul class="nav nav-pills">
<li class="nav-item"><a href="/" class="nav-link">Home</a></li>
<li class="nav-item"><a href="/courses" class="nav-link">Course</a></li>
<li class="nav-item"><a href="/profile" class="nav-link">Profile</a></li>
<li class="nav-item"><a href="/about-us" class="nav-link">About us</a></li>
</ul>
</header>
<main>
<div class="card-body p-md-5">
<div class="row justify-content-center">
<div class="col-md-10 col-lg-6 col-xl-5 order-2 order-lg-1">
<p class="text-center h1 fw-bold mb-5 mt-4">Sign up</p>
<form action="#" th:action="@{/registration}" th:object="${user}" method="POST">
<div class="d-flex flex-row align-items-center mb-4">
<i class="fas fa-envelope fa-lg me-3 fa-fw"></i>
<div class="form-outline flex-fill mb-0">
```

## ПРОДОВЖЕННЯ ДОДАТКУ Є

```
<input type="email" th:field="*{email}" id="form3Example3c" class="form-control"/>
<label class="form-label" for="form3Example3c" style="margin-left: 0px;">Your Email</label>
<div class="form-notch"><div class="form-notch-leading" style="width: 9px;"></div><div class="form-notch-middle"
style="width: 69.6px;"></div><div class="form-notch-trailing"></div></div>
</div>
<div class="d-flex flex-row align-items-center mb-4">
<i class="fas fa-lock fa-lg me-3 fa-fw"></i>
<div class="form-outline flex-fill mb-0">
<input type="password" th:field="*{password}" id="form3Example4c" class="form-control"/>
<label class="form-label" for="form3Example4c" style="margin-left: 0px;">Password</label>
<div class="form-notch"><div class="form-notch-leading" style="width: 9px;"></div><div class="form-notch-middle"
style="width: 64px;"></div><div class="form-notch-trailing"></div></div>
</div>
<div class="d-flex flex-row align-items-center mb-4">
<i class="fas fa-key fa-lg me-3 fa-fw"></i>
<div class="form-outline flex-fill mb-0">
<input type="password" id="form3Example4cd" class="form-control"/>
<label class="form-label" for="form3Example4cd" style="margin-left: 0px;">Repeat your password</label>
<div class="form-notch"><div class="form-notch-leading" style="width: 9px;"></div><div class="form-notch-middle"
style="width: 136px;"></div><div class="form-notch-trailing"></div></div>
</div>
<div class="d-flex justify-content-center mx-4 mb-3 mb-lg-4">
<input type="submit" value="Register" class="btn btn-primary btn-lg"/>
</div>
</form>
</div>
<div class="col-md-10 col-lg-6 col-xl-7 d-flex align-items-center order-1 order-lg-2">

```



```
</div>  
</div>  
</div>  
</main>  
</body>  
</html>
```



```

<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<title>Sign In</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">
</head>
<body>
<header class="d-flex flex-wrap justify-content-center py-3 mb-4 border-bottom">
<a href="/" class="d-flex align-items-center mb-3 mb-md-0 me-md-auto text-dark text-decoration-none">
<svg class="bi me-2" width="40" height="32"><use xlink:href="#bootstrap"></use></svg>
<span class="fs-4">WayIT</span>
</a>
<ul class="nav nav-pills">
<li class="nav-item"><a href="/" class="nav-link">Home</a></li>
<li class="nav-item"><a href="/courses" class="nav-link">Course</a></li>
<li class="nav-item"><a href="/profile" class="nav-link">Profile</a></li>
<li class="nav-item"><a href="/about-us" class="nav-link">About us</a></li>
</ul>
</header>
<div class="container">
<div class="col-md-7 col-lg-8">
<h4 class="mb-3">Billing address</h4>
<form class="needs-validation" novalidate="">
<div class="row g-3">
<div class="col-sm-6">
<label for="firstName" class="form-label">First name</label>
<input type="text" class="form-control" id="firstName" placeholder="" value="" required="">
<div class="invalid-feedback">
Valid first name is required.
</div>
</div>
</div>

```

## ПРОДОВЖЕННЯ ДОДАТКУ Ж

```
<div class="col-sm-6">
  <label for="lastName" class="form-label">Last name</label>
  <input type="text" class="form-control" id="lastName" placeholder="" value="" required="">
  <div class="invalid-feedback">
    Valid last name is required.
  </div>
</div>

<div class="col-12">
  <label for="username" class="form-label">Username</label>
  <div class="input-group has-validation">
    <span class="input-group-text">@</span>
    <input type="text" class="form-control" id="username" placeholder="Username" required="">
    <div class="invalid-feedback">
      Your username is required.
    </div>
  </div>
</div>

<div class="col-12">
  <label for="email" class="form-label">Email <span class="text-muted">(Optional)</span></label>
  <input type="email" class="form-control" id="email" placeholder="you@example.com">
  <div class="invalid-feedback">
    Please enter a valid email address for shipping updates.
  </div>
</div>

<div class="col-12">
  <label for="address" class="form-label">Address</label>
  <input type="text" class="form-control" id="address" placeholder="1234 Main St" required="">
  <div class="invalid-feedback">
    Please enter your shipping address.
  </div>
</div>
```

## ПРОДОВЖЕННЯ ДОДАТКУ Ж

```
<div class="col-md-5">
  <label for="country" class="form-label">Country</label>
  <select class="form-select" id="country" required="">
    <option value="">Choose...</option>
    <option>United States</option>
  </select>
  <div class="invalid-feedback">
    Please select a valid country.
  </div>
</div>
<div>
  <hr class="my-4">
  <button class="w-100 btn btn-primary btn-lg" type="submit">Continue</button>
</form>
</div>
<main>
</main>
</body>
</html>
```



## ДОДАТОК 3

```
spring.datasource.url=jdbc:mysql://localhost:3306/academy_db
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.username=root
spring.datasource.password=skrupa0260698
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL8Dialect
```



```

@Service
@RequiredArgsConstructor
public class UserService {
    private final UserDao userDao;
    private final PasswordEncoder passwordEncoder;

    public User saveUser(User user) {
        user.setPassword(passwordEncoder.encode(user.getPassword()));
        user.getRoles().add(Role.USER);
        userDao.save(user);
        return user;
    }

    public User getUserById(Long id) {
        Optional<User> byId = userDao.findById(id);
        return byId.orElseGet(() -> byId.orElseThrow(() -> new RuntimeException("User with " + id
            + " not found")));
    }
}

@Service
@RequiredArgsConstructor
public class TestService {
}

@Service
@RequiredArgsConstructor
public class RatingService {
}

@Service
@RequiredArgsConstructor
public class ProfileService {
    private final ProfileDao profileDao;

    public Profile addProfile(Profile profile) {
        profileDao.save(profile);
        return profile;
    }
}

@Service
@RequiredArgsConstructor
public class CustomUserDetailsService implements UserDetailsService {
    private final UserDao userDao;

    @Override
    public UserDetails loadUserByUsername(String email) throws UsernameNotFoundException {
        return userDao.findByEmail(email);
    }
}

```

## ДОДАТОК К

```
}  
  
@Entity  
@Table(name = "users")  
public class User implements UserDetails {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    @Column(name = "id")  
    private Long id;  
    @Column(unique = true)  
    private String email;  
    private String password;  
    @ElementCollection(targetClass = Role.class, fetch = FetchType.EAGER)  
    @CollectionTable(name = "user_role", joinColumns = @JoinColumn(name = "user_id"))  
    @Enumerated(EnumType.STRING)  
    private Set<Role> roles = new HashSet<>();  
    @OneToOne(cascade = CascadeType.ALL)  
    @JoinColumn(name = "profile_id", referencedColumnName = "id")  
    private Profile profile;  
  
    public Long getId() {  
        return id;  
    }  
  
    public void setId(Long id) {  
        this.id = id;  
    }  
  
    public String getEmail() {  
        return email;  
    }  
  
    public Profile getProfile() {  
        return profile;  
    }  
  
    public void setProfile(Profile profile) {  
        this.profile = profile;  
    }  
  
    public void setEmail(String email) {  
        this.email = email;  
    }  
  
    @Override  
    public Collection<? extends GrantedAuthority> getAuthorities() {  
        return roles;  
    }  
  
    public String getPassword() {  
        return password;  
    }  
}
```

}

@Override

```
public String getUsername() {  
    return email;  
}
```

@Override

```
public boolean isAccountNonExpired() {  
    return true;  
}
```

@Override

```
public boolean isAccountNonLocked() {  
    return true;  
}
```

@Override

```
public boolean isCredentialsNonExpired() {  
    return true;  
}
```

@Override

```
public boolean isEnabled() {  
    return true;  
}
```

```
public void setPassword(String password) {  
    this.password = password;  
}
```

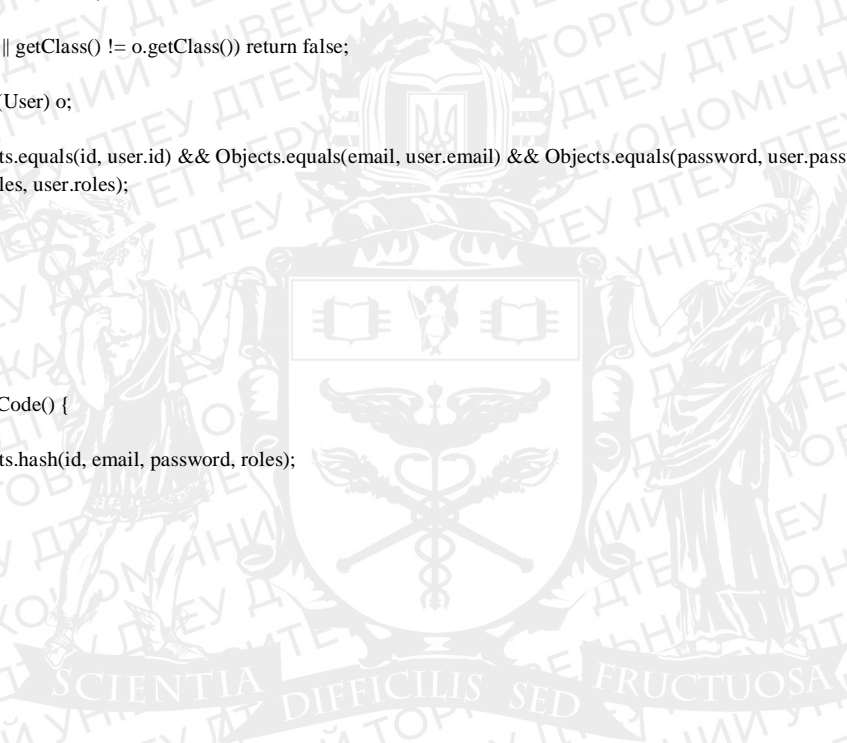
```
public Set<Role> getRoles() {  
    return roles;  
}
```





## ПРОДОВЖЕННЯ ДОДАТКУ К

```
public void setRoles(Set<Role> roles) {  
    this.roles = roles;  
}  
  
@Override  
public boolean equals(Object o) {  
    if (this == o) return true;  
    if (o == null || getClass() != o.getClass()) return false;  
    User user = (User) o;  
    return Objects.equals(id, user.id) && Objects.equals(email, user.email) && Objects.equals(password, user.password) &&  
        Objects.equals(roles, user.roles);  
}  
  
@Override  
public int hashCode() {  
    return Objects.hash(id, email, password, roles);  
}
```



```
@Entity
@Table(name = "tests")
public class Test {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String question;

    private String first_variant;
    private String second_variant;
    private String third_variant;
    private String answer;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getQuestion() {
        return question;
    }

    public void setQuestion(String question) {
        this.question = question;
    }

    public String getFirst_variant() {
        return first_variant;
    }

    public void setFirst_variant(String first_variant) {
        this.first_variant = first_variant;
    }
}
```

```
}  
  
public String getSecond_variant() {  
    return second_variant;  
}  
  
public void setSecond_variant(String second_variant) {  
    this.second_variant = second_variant;  
}  
  
public String getThird_variant() {  
    return third_variant;  
}  
  
public void setThird_variant(String third_variant) {  
    this.third_variant = third_variant;  
}  
  
public String getAnswer() {  
    return answer;  
}  
  
public void setAnswer(String answer) {  
    this.answer = answer;  
}  
  
@Override  
public boolean equals(Object o) {  
    if (this == o) return true;  
    if (o == null || getClass() != o.getClass()) return false;  
    Test test = (Test) o;  
    return Objects.equals(id, test.id) && Objects.equals(question, test.question) && Objects.equals(first_variant, test.first_variant) &&  
    Objects.equals(second_variant, test.second_variant) && Objects.equals(third_variant, test.third_variant) && Objects.equals(answer,  
    test.answer);  
}  
}
```

## ПРОДОВЖЕННЯ ДОДАТКУ Л

@Override

```
public int hashCode() {
```

```
    return Objects.hash(id, question, first_variant, second_variant, third_variant, answer);
```

```
}
```

```
}
```



```
@Entity
@Table(name = "ratings")
public class Rating {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @OneToOne
    private Profile profile;
    private int progress;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public Profile getProfile() {
        return profile;
    }

    public void setProfile(Profile profile) {
        this.profile = profile;
    }

    public int getProgress() {
        return progress;
    }

    public void setProgress(int progress) {
        this.progress = progress;
    }
}
```

```
@Override
```

```
public boolean equals(Object o) {
```

```
    if (this == o) return true;
```

```
    if (o == null || getClass() != o.getClass()) return false;
```

```
    Rating rating = (Rating) o;
```

```
    return progress == rating.progress && Objects.equals(id, rating.id) && Objects.equals(profile, rating.profile);
```

```
}
```

```
@Override
```

```
public int hashCode() {
```

```
    return Objects.hash(id, profile, progress);
```

```
}
```

```
}
```

```
@Entity
```

```
@Table(name = "profiles")
```

```
public class Profile {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.AUTO)
```

```
    @Column(name = "id")
```

```
    private Long id;
```

```
    @OneToOne(mappedBy = "profile")
```

```
    private User user;
```

```
    private String first_name;
```

```
    private String email;
```

```
    private String phone;
```

```
    private String address;
```

```
    public Long getId() {
```

```
        return id;
```

```
}
```

```
    public void setId(Long id) {
```

```
        this.id = id;
```

```
}  
  
public User getUser() {  
    return user;  
}  
  
public void setUser(User user) {  
    this.user = user;  
}  
  
public String getFirst_name() {  
    return first_name;  
}  
  
public void setFirst_name(String first_name) {  
    this.first_name = first_name;  
}  
  
public String getEmail() {  
    return email;  
}  
  
public void setEmail(String email) {  
    this.email = email;  
}  
  
public String getPhone() {  
    return phone;  
}  
  
public void setPhone(String phone) {  
    this.phone = phone;  
}
```

## ПРОДОВЖЕННЯ ДОДАТКУ М

```
public String getAddress() {  
    return address;  
}  
  
public void setAddress(String address) {  
    this.address = address;  
}  
  
@Override  
public boolean equals(Object o) {  
    if (this == o) return true;  
    if (o == null || getClass() != o.getClass()) return false;  
    Profile profile = (Profile) o;  
    return Objects.equals(id, profile.id) && Objects.equals(user, profile.user) && Objects.equals(first_name, profile.first_name) &&  
        Objects.equals(email, profile.email) && Objects.equals(phone, profile.phone) && Objects.equals(address, profile.address);  
}  
  
@Override  
public int hashCode() {  
    return Objects.hash(id, user, first_name, email, phone, address);  
}  
}
```



```
@Entity
```

```
@Table(name = "materials")
```

```
public class Materials {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private Long id;
```

```
    private String name;
```

```
    @OneToMany
```

```
    private List<Test> tests;
```

```
    private String description;
```

```
    public Long getId() {
```

```
        return id;
```

```
    }
```

```
    public void setId(Long id) {
```

```
        this.id = id;
```

```
    }
```

```
    public String getName() {
```

```
        return name;
```

```
    }
```

```
    public void setName(String name) {
```

```
        this.name = name;
```

```
    }
```

```
    public List<Test> getTests() {
```

```
        return tests;
```

```
    }
```

```
    public void setTests(List<Test> tests) {
```

```
        this.tests = tests;
```

```
    }
```



## ПРОДОВЖЕННЯ ДОДАТКУ Н

```
public String getDescription() {  
    return description;  
}  
  
public void setDescription(String description) {  
    this.description = description;  
}  
  
@Override  
public boolean equals(Object o) {  
    if (this == o) return true;  
    if (o == null || getClass() != o.getClass()) return false;  
    Materials materials = (Materials) o;  
    return Objects.equals(id, materials.id) && Objects.equals(name, materials.name) && Objects.equals(tests, materials.tests) &&  
        Objects.equals(description, materials.description);  
}  
  
@Override  
public int hashCode() {  
    return Objects.hash(id, name, tests, description);  
}  
}
```

```
@Entity
@Table(name = "courses")
public class Course {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @OneToMany
    private List<Materials> materials;
    private String name;
    private String description;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public List<Materials> getMaterials() {
        return materials;
    }

    public void setMaterials(List<Materials> materials) {
        this.materials = materials;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```



## ПРОДОВЖЕННЯ ДОДАТКУ О

```
public String getDescription() {
    return description;
}

public void setDescription(String description) {
    this.description = description;
}

@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;
    Course course = (Course) o;
    return Objects.equals(id, course.id) && Objects.equals(materials, course.materials) && Objects.equals(name, course.name) &&
        Objects.equals(description, course.description);
}

@Override
public int hashCode() {
    return Objects.hash(id, materials, name, description);
}
}
```

```
@Controller
```

```
@RequiredArgsConstructor
```

```
public class AuthenticationController {
```

```
    private final UserService userService;
```

```
    private final ProfileService profileService;
```

```
    @GetMapping("/registration")
```

```
    public String registration(Model model) {
```

```
        model.addAttribute("user", new User());
```

```
        return "register";
```

```
    }
```

```
    @PostMapping("/registration")
```

```
    public String saveUser(@ModelAttribute User user, Model model) {
```

```
        model.addAttribute("user", user);
```

```
        Profile profile = new Profile();
```

```
        profile.setEmail(user.getEmail());
```

```
        profileService.addProfile(profile);
```

```
        userService.saveUser(user);
```

```
        return "home";
```

```
    }
```

```
    @GetMapping("/login")
```

```
    public String login(Model model) {
```

```
        model.addAttribute("user", new User());
```

```
        return "login";
```

```
    }
```

```
}
```



```
@Controller
public class HomeController {

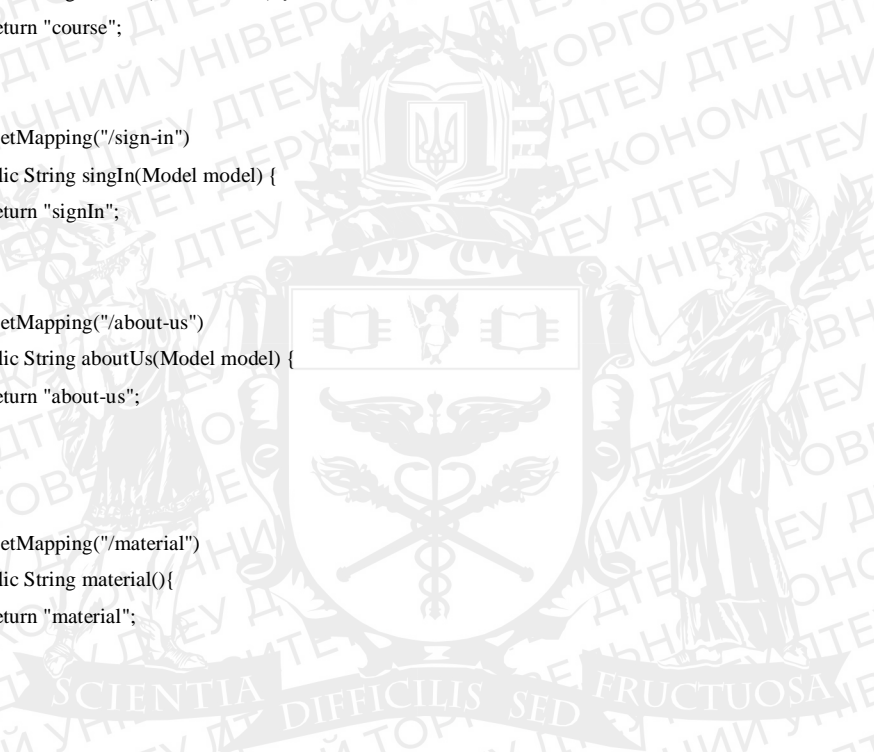
    @GetMapping("/")
    public String home(Model model) {
        return "home";
    }

    @GetMapping("/courses")
    public String courses(Model model) {
        return "course";
    }

    @GetMapping("/sign-in")
    public String signIn(Model model) {
        return "signIn";
    }

    @GetMapping("/about-us")
    public String aboutUs(Model model) {
        return "about-us";
    }

    @GetMapping("/material")
    public String material(){
        return "material";
    }
}
```



```
@Configuration
```

```
@EnableWebSecurity
```

```
@RequiredArgsConstructor
```

```
public class SecurityConfig {
```

```
    private final CustomUserDetailsService service;
```

```
    @Bean
```

```
    protected SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
```

```
        http
```

```
            .authorizeHttpRequests((requests) -> requests
```

```
                .requestMatchers("/", "/home", "/about-us", "/registration").permitAll()
```

```
                .anyRequest().authenticated()
```

```
            )
```

```
            .formLogin((form) -> form
```

```
                .loginPage("/login")
```

```
                .permitAll()
```

```
            )
```

```
            .logout(LogoutConfigurer::permitAll);
```

```
        return http.build();
```

```
    }
```

```
    @Bean
```

```
    public AuthenticationProvider authenticationProvider() {
```

```
        DaoAuthenticationProvider authenticationProvider = new DaoAuthenticationProvider();
```

```
        authenticationProvider.setUserDetailsService(service);
```

```
        authenticationProvider.setPasswordEncoder(passwordEncoder());
```

```
        return authenticationProvider;
```

```
    }
```

```
    @Bean
```

```
    public PasswordEncoder passwordEncoder() {
```

```
        return new BCryptPasswordEncoder(8);
```

## ДОДАТОК Т

```
@Controller
@RequiredArgsConstructor
public class ProfileController {
    private final UserService userService;
    private final ProfileService profileService;

    @GetMapping("/profile")
    public String profile(Model model) {
        model.addAttribute("profile", new Profile());
        // profileService.getUserOfProfile();
        // userService.getUserById()
        return "profile";
    }
}
```

