ДЕРЖАВНА СЛУЖБА УКРАЇНИ З НАДЗВИЧАЙНИХ СИТУАЦІЙ
ЛЬВІВСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ БЕЗПЕКИ ЖИТТЄДІЯЛЬНОСТІ
Навчально-науковий інститут цивільного захисту
Кафедра інформаційних технологій та систем електронних комунікацій

WEB-ТЕХНОЛОГІЇ ТА WEB-ПРОГРАМУВАННЯ

«ПРОЄКТУВАННЯ ТА РОЗРОБЛЕННЯ ВЕБСЕРВІСУ НА SPRING BOOT»

Методичні вказівки до виконання курсової роботи
з дисципліни «WEB-технології та WEB-програмування» (ОК 2.15)
для здобувачів першого (бакалаврського) рівня вищої освіти
спеціальності F3 «Комп’ютерні науки»

Львів – 2026

Проєктування та розроблення вебсервісу на Spring Boot : методичні вказівки до виконання курсової роботи з дисципліни «WEB-технології та WEB-програмування» для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності F3 «Комп’ютерні науки» / уклад. Р. Р. Головатий. – Львів : ЛДУ БЖД, 2026. – 21 с.

Укладач: Роман ГОЛОВАТИЙ, доцент кафедри інформаційних технологій та систем електронних комунікацій, кандидат технічних наук.

Рецензент: _____

Затверджено на засіданні кафедри інформаційних технологій та систем електронних комунікацій (протокол № ____ від «____» _____ 20___ р.).

Ухвалено методичною радою навчально-наукового інституту цивільного захисту (протокол № ____ від «____» _____ 20___ р.).

Методичні вказівки визначають вимоги до змісту, обсягу, оформлення, порядку виконання і захисту курсової роботи, містять перелік індивідуальних завдань і критерії оцінювання. Курсова робота є підсумковим етапом вивчення дисципліни: здобувач проєктує і розробляє власний вебсервіс, застосовуючи технології, опановані на лабораторних роботах № 1–8.

© Роман ГОЛОВАТИЙ, 2026
© ЛДУ БЖД, 2026

ЗМІСТ

Вступ 4
1. Мета, завдання і формати виконання курсової роботи 5
2. Вимоги до вебсервісу 6
3. Індивідуальні завдання 8
4. Етапи і терміни виконання 10
5. Структура і зміст пояснювальної записки 11
6. Вимоги до оформлення пояснювальної записки 13
7. Подання і захист курсової роботи 14
8. Критерії оцінювання 16
9. Академічна доброчесність і використання засобів ШІ 18
Список рекомендованих джерел 19
Додаток А. Титульний аркуш пояснювальної записки 20
Додаток Б. Контрольний перелік вимог до вебсервісу 21

ВСТУП

Сучасні інформаційні системи – реєстри, сервіси бронювання, інтернет-магазини, системи обліку – найчастіше реалізуються як вебсервіси: серверний застосунок зберігає дані в реляційній базі, надає користувачам вебсторінки, а іншим програмам – програмний інтерфейс (REST API), розмежовує доступ за ролями і розгортається в контейнерах. Саме такий застосунок здобувачі освіти розробляють у курсовій роботі з дисципліни «WEB-технології та WEB-програмування».

Курсова робота є підсумковим етапом вивчення дисципліни і завершальною частиною наскрізного проєкту: протягом семестру на лабораторних роботах № 1–8 кожен здобувач поступово будує застосунок для власної предметної області – від статичної сторінки і схеми бази даних (ЛР 1–2) через багатошарову архітектуру і сервлети (ЛР 3–4) до вебсервісу на Spring Boot з REST API, Spring Data JPA, Spring Security і автоматичними тестами (ЛР 5–8). Зразком слугує демонстраційний проєкт «Реєстр укриттів міста» (https://github.com/SidiromUA/shelter-registry), стан якого на кожному етапі зафіксовано git-тегами `lab-01`…`lab-08`.

Виконання курсової роботи закріплює теоретичні знання і практичні навички проєктування, розроблення, тестування та документування вебзастосунків, формує вміння обґрунтовувати технічні рішення і презентувати результати власної роботи. Якісно виконаний вебсервіс може стати частиною портфоліо здобувача, основою кваліфікаційної роботи або студентського R&D-проєкту.

1. МЕТА, ЗАВДАННЯ І ФОРМАТИ ВИКОНАННЯ КУРСОВОЇ РОБОТИ

Мета курсової роботи – спроєктувати і розробити працездатний вебсервіс для обраної предметної області з використанням технологій Java і Spring Boot, реляційної бази даних, вебсторінок і RESTful API, засобів безпеки та автоматичного тестування.

Завдання курсової роботи:

1. проаналізувати предметну область, визначити користувачів (ролі) і функціональні вимоги до вебсервісу;
2. спроєктувати модель даних, архітектуру застосунку, інтерфейс вебсторінок і REST API;
3. реалізувати вебсервіс відповідно до вимог розділу 2;
4. забезпечити безпеку, доступність і якість (валідація даних, обробка помилок, тести);
5. підготувати пояснювальну записку і документацію API;
6. презентувати і захистити результати роботи.

Курсова робота може виконуватися в одному з двох форматів.

Індивідуальний формат. Здобувач виконує роботу самостійно за індивідуальним тематичним напрямом (розділ 3). Конкретну ідею вебсервісу в межах напряму здобувач формулює сам і погоджує з керівником. Рекомендується продовжувати проєкт, розпочатий на лабораторних роботах, – тоді курсова робота розвиває та доповнює вже створений код.

Командний формат. Команда з 2–4 здобувачів спільно розв’язує комплексне завдання в межах студентського R&D-проєкту або наукової роботи кафедри (наприклад, складові проєктів «Розумний університет»: замовлення довідок, електронний журнал, розклад, чат-бот «студент – деканат»; «Портал вступника»; інформаційні системи цивільного захисту). Склад команди, тема, розподіл обов’язків і очікуваний результат погоджуються з керівником до кінця вивчення теми 1.3. Кожен учасник відповідає за окремий, чітко визначений модуль (набір сутностей, сторінок, ендпоінтів і тестів), що підтверджується історією комітів у спільному репозиторії. Пояснювальну записку кожен учасник оформлює індивідуально; спільними можуть бути лише опис загальної архітектури і моделі даних. Вимоги розділу 2 застосовуються до кожного учасника в межах його модуля з урахуванням спільної частини.

2. ВИМОГИ ДО ВЕБСЕРВІСУ

Вебсервіс курсової роботи має бути повноцінним застосунком, а не набором статичних сторінок. Мінімальні обов'язкові вимоги наведено в табл. 2.1; їх виконання є умовою допуску до захисту. Відповідність вимогам здобувач перевіряє за контрольним переліком (додаток Б).

Аспект	**Обов'язкові вимоги**	**ЛР**
Технології	Java 21, Spring Boot 4 (або актуальна підтримувана версія 3.x), Maven або Gradle; PostgreSQL; Git-репозиторій на GitHub з осмисленою історією комітів	1, 2, 5
Модель даних	не менше **5 сутностей** предметної області, щонайменше два зв'язки «один до багатьох» і один «багато до багатьох» (або сутність-зв'язок); обмеження цілісності в БД (PK, FK, NOT NULL, UNIQUE, CHECK)	2, 8
Доступ до даних	Spring Data JPA (сутності, репозиторії, похідні запити і JPQL); схема і тестові дані – міграціями Flyway; пагінація хоча б одного списку; відсутність проблеми N+1 на основних сторінках	8
Архітектура	шари контролер – сервіс – репозиторій; бізнес-правила в сервісах (не в контролерах і шаблонах); вхідні і вихідні DTO; впровадження залежностей через конструктор	3, 5
Вебсторінки	не менше **10 сторінок** (шаблонів Thymeleaf) з єдиним макетом через фрагменти: списки з фільтрами або пошуком, картки об'єктів, форми створення і редагування з валідацією, сторінки входу і помилок; шаблон Post/Redirect/Get	4, 7
REST API	CRUD щонайменше для двох основних ресурсів і один вкладений ресурс; правильні методи HTTP і коди стану (200, 201 + Location, 204, 400, 401/403, 404, 409); формат JSON; документація OpenAPI (Swagger UI)	7
Валідація і помилки	Bean Validation на вхідних DTO, бізнес-правила в сервісах; централізована обробка помилок (Problem Details для API, сторінки помилок для вебчастини)	7
Безпека	Spring Security: вхід через форму для сторінок, HTTP Basic або токен для API; щонайменше **2 ролі** з різними правами; паролі – хешами BCrypt; захист CSRF для форм; дані не виводяться без екранування	4, 8
Конфігурація	налаштування в application.yml; профілі dev і prod; секрети лише зі змінних оточення; журналювання через SLF4J; Actuator (health)	6

Тестування	не менше **10 автоматичних тестів**: модульні тести сервісів (JUnit 5, Mockito), тести репозиторіїв (@DataJpaTest), тести API з перевіркою безпеки (@WebMvcTest); команда `mvn test` завершується успішно	8
Розгортання	Dockerfile (багатоетапне збирання) і docker-compose.yml для запуску застосунку разом із PostgreSQL; README з інструкцією запуску	6
Доступність	семантична розмітка, підписи полів форм, текстові альтернативи зображень, достатній контраст, навігація з клавіатури; оцінка Lighthouse Accessibility не менше 90 для основних сторінок	3

Таблиця 2.1 – Обов'язкові вимоги до вебсервісу

Функціональна повнота. Вебсервіс має реалізовувати завершений сценарій предметної області, а не лише операції CRUD: наприклад, бронювання з перевіркою зайнятості, зміна стану замовлення за правилами, облік залишків, розрахунок вартості, звіт за період. У роботі має бути щонайменше **три бізнес-правила**, що перевіряються в сервісному шарі і покриті модульними тестами (у демо-проєкті: унікальність адреси укриття, зміна стану укриття за результатом перевірки, пошук укриттів із простроченою перевіркою).

Інтерфейс користувача. Дизайн сторінок здобувач обирає самостійно; дозволяється використовувати CSS-фреймворки (Bootstrap, Pico CSS тощо) та JavaScript для покращення взаємодії, але основна функціональність має працювати на сервері. Сторінки мають коректно відображатися на екранах ноутбука і смартфона.

Додаткові можливості, які враховуються під час оцінювання, але не є обов'язковими: завантаження файлів (зображень), надсилання електронних листів, експорт звітів (CSV, PDF), інтерактивні елементи на JavaScript із запитами до власного REST API, інтеграція із зовнішнім API, локалізація інтерфейсу, Testcontainers, автоматичне збирання і тестування в GitHub Actions, розгортання на хмарній платформі.

Запозичення готових шаблонів сайтів і чужого коду без зазначення джерела не допускається (розділ 9). Код, згенерований із використанням засобів штучного інтелекту, здобувач має розуміти і вміти пояснити під час захисту.

3. ІНДИВІДУАЛЬНІ ЗАВДАННЯ

Індивідуальне завдання – це тематичний напрям, у межах якого здобувач самостійно формулює ідею вебсервісу з унікальною логікою, дотримуючись вимог розділу 2. Номер напряму відповідає порядковому номеру здобувача в академічному журналі групи (за номера понад 25 – номер мінус 25). За погодженням із керівником здобувач може обрати інший напрям, зокрема тему, пов’язану з науковою роботою, навчальною практикою чи реальним замовником. Тему (ідею, ролі, основні сутності) необхідно погодити з керівником до кінця вивчення теми 1.3 (табл. 4.1); з того часу вона є предметною областю лабораторних робіт № 2–8.

№	Тематичний напрям	№	Тематичний напрям
1	Готель	14	Станція технічного обслуговування
2	Туристична фірма	15	Огляд техніки
3	Кулінарія	16	Служба таксі
4	Бібліотека	17	Приватний навчальний заклад
5	Волонтерський сайт допомоги Силам оборони	18	ІТ-компанія
6	Нерухомість	19	Криптовалюти
7	Перукарня	20	Банк
8	Магазин військової амуніції	21	Страхова компанія
9	Кафе	22	Тренажерний зал
10	Новинний сайт	23	Виробництво безпілотників
11	ОСББ	24	Пекарня
12	Футбольний клуб	25	Автотранспортне підприємство
13	Приватна стоматологічна клініка		

Таблиця 3.1 – Тематичні напрями індивідуальних завдань

Приклад формулювання ідеї. Напрям 22 «Тренажерний зал». Ідея: вебсервіс клубу з онлайн-записом на групові заняття і обліком абонементів.

- *Ролі:* гість (розклад, послуги, тренери), клієнт (запис на заняття, власні абонементи, історія відвідувань), тренер (свої заняття, відмітка відвідування), адміністратор (послуги, тренери, розклад, абонементи, звіти).
- *Сутності:* Послуга, Тренер, Заняття (дата, час, зал, кількість місць), Клієнт, Абонемент (тип, термін дії, залишок відвідувань), Запис на

заняття (зв’язок клієнта і заняття), Відвідування.

- *Бізнес-правила:* не можна записатися на заняття без чинного абонемента; кількість записів не перевищує кількості місць; скасувати запис можна не пізніше ніж за 2 години; відвідування зменшує залишок абонемента.
- *Сторінки:* головна, розклад із фільтром за послугою і тренером, картка заняття, профіль тренера, «Мої записи», «Мої абонементи», форма запису, панель тренера, адміністрування розкладу і абонементів, звіт про завантаженість залів, вхід, помилки.
- *REST API:* `/api/classes` (розклад із пагінацією), `/api/classes/{id}/bookings` (запис і скасування, 409 у разі відсутності місць), `/api/memberships`.

Ідеї вебсервісів у межах одного напряму в різних здобувачів не повинні збігатися. Унікальність і практична цінність ідеї, а також креативність її реалізації враховуються під час захисту.

4. ЕТАПИ І ТЕРМІНИ ВИКОНАННЯ

Курсова робота виконується паралельно з лабораторними роботами, тому кожен етап спирається на щойно опановану тему (табл. 4.1). Результат кожного етапу фіксується комітом і git-тегом у репозиторії; керівник може переглядати поступ на консультаціях.

№	Етап	Результат	Термін
1	Вибір і погодження теми	ідея, ролі, перелік сутностей і бізнес-правил погоджено з керівником	тема 1.3
2	Модель даних	ER-діаграма, схема БД з обмеженнями, тестові дані (ЛР 2)	тема 1.4
3	Архітектура і прототип	структура пакетів, мокапи основних сторінок, проєкт Spring Boot (ЛР 5)	тема 2.1
4	Конфігурація і розгортання	профілі, Docker Compose (ЛР 6)	тема 2.2
5	Функціональність	сторінки, REST API з документацією, валідація, обробка помилок (ЛР 7)	тема 2.3
6	Дані, безпека, тести	JPA, Flyway, Spring Security, автоматичні тести (ЛР 8)	тема 2.4
7	Пояснювальна записка	записка і посилання на репозиторій подано у «Віртуальний університет»	за 2 тижні до екзамену
8	Захист	презентація і демонстрація вебсервісу комісії	за розкладом кафедри

Таблиця 4.1 – Етапи виконання курсової роботи

Консультації з курсової роботи проводяться згідно з графіком консультацій кафедри. Здобувач, який не встиг виконати етап, обговорює з керівником план надолуження; систематичне невиконання етапів може стати підставою для недопуску роботи до захисту в строк.

5. СТРУКТУРА І ЗМІСТ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Пояснювальна записка обсягом **25–40 сторінок** основної частини (без додатків) містить такі структурні елементи:

1. Титульний аркуш (додаток А).
2. Анотація (до 1 с.) – тема, мета, коротка характеристика вебсервісу і використаних технологій, кількість сторінок, рисунків, таблиць, лістингів, джерел і додатків; ключові слова.
3. Зміст.
4. Вступ (1–2 с.) – актуальність, проблема, яку розв’язує вебсервіс, мета і завдання роботи, загальна ідея вебсервісу.
5. Розділ 1. Аналіз предметної області і вимоги до вебсервісу.
6. Розділ 2. Проєктування вебсервісу.
7. Розділ 3. Реалізація вебсервісу.
8. Розділ 4. Тестування і розгортання.
9. Висновки (1 с.).
10. Список використаних джерел.
11. Додатки (за необхідності).

Розділ 1. Аналіз предметної області і вимоги до вебсервісу (4–6 с.). Опис предметної області і процесів, які автоматизує вебсервіс; короткий огляд 2–3 наявних аналогів із порівнянням; ролі користувачів і їхні можливості (таблиця «роль – дії»); функціональні вимоги у вигляді сценаріїв використання (наприклад, «Клієнт записується на заняття»); бізнес-правила; нефункціональні вимоги (безпека, доступність, продуктивність).

Розділ 2. Проєктування вебсервісу (6–10 с.). Обґрунтування вибору технологій; архітектура застосунку (шари, пакети, основні класи – схема); модель даних (ER-діаграма, опис таблиць, зв’язків і обмежень); мокапи основних сторінок із поясненням навігації (спосіб створення мокапів – на вибір здобувача: Figma, Penpot, draw.io, ескізи); проєкт REST API (таблиця «метод – шлях – опис – коди відповіді – доступ»); модель безпеки (ролі і правила доступу).

Розділ 3. Реалізація вебсервісу (10–16 с.). Опис реалізації за шарами з лістингами ключових фрагментів коду: сутності і репозиторії (похідні запити, JPQL, пагінація), міграції, сервіси з бізнес-правилами і транзакціями, контролери сторінок і шаблони, REST-контролери, валідація і обробка помилок,

конфігурація безпеки. До кожного лістингу – пояснення: що робить фрагмент, які рішення прийнято і чому. Знімки екрана основних сторінок і Swagger UI. Код у записці наводять вибірково – повний код міститься в репозиторії; документаційні коментарі (Javadoc) до публічних методів сервісів обов'язкові.

Розділ 4. Тестування і розгортання (3–5 с.). Перелік автоматичних тестів (таблиця «клас – що перевіряє – вид тесту») з лістингами 2–3 характерних тестів і результатом `mvn test`; результати ручного тестування основних сценаріїв і перевірки API в Postman; результати перевірки доступності (Lighthouse); опис конфігурації профілів, Dockerfile, docker-compose.yml та інструкція запуску.

Висновки містять основні результати роботи, відповідність вебсервісу поставленим завданням, обмеження реалізації і напрями подальшого розвитку.

Список використаних джерел містить лише ті джерела, на які є посилання в тексті, зокрема документацію фреймворків, бібліотек і використані електронні ресурси. Якщо під час виконання роботи використовувалися засоби штучного інтелекту, у вступі або в окремому підрозділі зазначають інструмент і мету його застосування (розділ 9).

Додатки можуть містити повні тексти окремих класів, додаткові знімки екрана, звіти Lighthouse, опис додаткових можливостей тощо.

6. ВИМОГИ ДО ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Пояснювальна записка оформлюється в електронному вигляді у форматі PDF на сторінках формату А4. Робоча мова – українська (англійською можуть бути назви технологій, ідентифікатори і коментарі в коді).

Параметр	**Вимога**
Поля сторінки	20 мм з усіх боків
Основний текст	Times New Roman, 14 пт, міжрядковий інтервал 1,15, вирівнювання за шириною, абзацний відступ 1,25 см
Програмний код (лістинги)	Consolas або Courier New, 12 пт, інтервал 1,0, без вирівнювання за шириною; код подається текстом, а не зображенням
Нумерація сторінок	наскрізна; титульний аркуш – перша сторінка, номер на ньому не ставиться
Розділи	кожен розділ – з нової сторінки; заголовок: номер і назва (наприклад, «2 ПРОЄКТУВАННЯ ВЕБСЕРВІСУ»); підрозділи – 2.1, 2.2; вступ, висновки і список джерел не нумеруються
Рисунки	нумерація в межах розділу або наскрізна; підпис під рисунком: «Рисунок 2.1 – Мокап сторінки „Розклад занять“»; на кожен рисунок – посилання в тексті
Таблиці	назва над таблицею: «Таблиця 2.2 – Ендпоінти REST API»
Лістинги	назва під лістингом: «Лістинг 3.4 – Сервіс запису на заняття»
Посилання на джерела	номер джерела у квадратних дужках [3]; список – за ДСТУ 8302:2015 у порядку появи посилань
Додатки	позначаються великими літерами (Додаток А, Додаток Б…) і наводяться у змісті

Таблиця 6.1 – Вимоги до оформлення

Текст має бути вичитаний, без граматичних і стилістичних помилок, з єдиною термінологією. Знімки екрана мають бути чіткими, без зайвих елементів інтерфейсу ОС, з читабельним текстом.

7. ПОДАННЯ І ЗАХИСТ КУРСОВОЇ РОБОТИ

Подання. Не пізніше ніж за два тижні до екзамену здобувач подає роботу в завданні «КУРСОВА РОБОТА (для перевірки)» розділу «Підсумковий контроль» «Віртуального університету»: файлом – пояснювальну записку у форматі PDF; у полі «Текстова відповідь» або в записці – посилання на публічний GitHub-репозиторій із тегом `kursova` на версії, поданій на перевірку. Репозиторій має містити README з описом вебсервісу, інструкцією запуску (зокрема через `docker compose up`) і тестовими обліковими записами для кожної ролі.

Рецензування. Керівник перевіряє відповідність роботи завданню і цим методичним вказівкам, наявність обов'язкових складових записки, працездатність вебсервісу (запуск за інструкцією з README, проходження тестів) і може перевірити текст і код на ознаки запозичень (розділ 9). У разі невідповідності роботу повертають на доопрацювання із зауваженнями; після їх усунення роботу допускають до захисту.

Захист відбувається перед комісією у складі керівника і 1–2 викладачів кафедри за графіком, погодженим із групою. Тривалість виступу – до 7 хвилин, після нього – запитання комісії. Під час захисту здобувач:

1. коротко представляє ідею, ролі та модель даних (можна з презентацією до 8 слайдів);
2. демонструє роботу вебсервісу за основними сценаріями для різних ролей, REST API у Swagger UI або Postman, запуск тестів;
3. пояснює ключові фрагменти коду і обґрунтовує прийняті рішення;
4. відповідає на запитання комісії; комісія може попросити внести невелику зміну в код або пояснити довільний фрагмент репозиторію, щоб переконатися в самостійності виконання.

Захист командної роботи проводиться спільно: керівник команди представляє загальну ідею і архітектуру, кожен учасник демонструє і пояснює свій модуль. Внесок кожного учасника оцінюється окремо, зокрема за історією комітів.

Захист проводиться один раз. Здобувач, який отримав 0 балів за критерієм «Вебсервіс» або «Захист» (розділ 8), вважається таким, що не виконав завдання, отримує нове завдання і виконує роботу повторно. Найкращі роботи

рекомендуються для подальшого розвитку в студентських R&D-проєктах, участі в конкурсах і хакатонах.

8. КРИТЕРІЇ ОЦІНЮВАННЯ

Курсову роботу оцінюють за 100-бальною шкалою за формулою $\mathbf{К = К_{сайт} + К_{записка} + К_{захист}}$, де складові визначають за критеріями, наведеними в табл. 8.1 (відповідно до силабусу дисципліни).

Бали	**Критерій**
Якість реалізації вебсервісу $К_{сайт}$ (0–50 балів)	
40–50	вебсервіс повністю відповідає вимогам методичних вказівок щодо архітектури, функціональності, дизайну, безпеки та доступності; всі функції працюють належним чином і відповідають індивідуальному завданню
30–39	незначні недоліки у функціональності, дизайні або безпеці; окремі частини можуть працювати нестабільно, проте загалом вебсервіс працездатний і відповідає основним вимогам
20–29	суттєві недоліки у функціональності, дизайні або безпеці; окремі функції не реалізовано або вони працюють некоректно
1–19	вебсервіс суттєво не відповідає вимогам та індивідуальному завданню, функціональність реалізовано частково або вебсервіс практично не працює
0	вебсервіс не представлено на захисті, він непрацездатний або запозичений з відкритих ресурсів чи в інших здобувачів
Якість пояснювальної записки $К_{записка}$ (0–30 балів)	
25–30	записка повністю відповідає вимогам, містить опис архітектури, функціональності, дизайну та безпеки вебсервісу, обґрунтування вибору технологій, увесь необхідний ілюстративний матеріал
20–24	записка відповідає вимогам не менше ніж на 80 %, має незначні зауваження щодо повноти ілюстративного матеріалу або опису реалізації
10–19	значні зауваження щодо якості оформлення та відповідності вимогам, проте всі обов’язкові розділи подано; обґрунтування вибору технологій неповне або не відповідає реалізації
1–9	записка суттєво не відповідає вимогам, відсутні обов’язкові складові
0	записку не подано або вона не відповідає поставленому завданню
Захист курсової роботи $К_{захист}$ (0–20 балів)	
15–20	здобувач вільно володіє матеріалом, оперує термінами вебтехнологій і вебпрограмування, ідея та шляхи її реалізації креативні, правильно відповідає на всі додаткові запитання
10–14	відповідає основним критеріям, проте під час відповідей на додаткові або уточнювальні запитання допускає неточності
1–9	під час презентації допускаються неточності або невідповідності матеріалу записки, відповіді на додаткові запитання неправильні чи неточні, що може свідчити про несамостійне виконання значної частини роботи
0	здобувач не може пояснити поданий на захист матеріал, що свідчить про несамостійність виконаної роботи

Таблиця 8.1 – Критерії оцінювання курсової роботи

Під час визначення $К_{сайт}$ враховують виконання вимог табл. 2.1 (кожна невиконана обов’язкова вимога знижує оцінку), складність і завершеність бізнес-логіки, якість коду (структура, іменування, відсутність дублювання), а також додаткові можливості. Робота, подана після встановленого терміну без поважних причин, оцінюється з пониженням балів відповідно до силабусу дисципліни. Підсумкову оцінку переводять у національну шкалу і шкалу ЄКТС (табл. 8.2).

Сума балів	**91–100**	**81–90**	**71–80**	**61–70**	**51–60**	**36–50**	**0–35**
Оцінка ЄКТС	A	B	C	D	E	FX	F
Національна шкала	відмінно	добре		задовільно		незадовільно	

Таблиця 8.2 – Шкала оцінювання

9. АКАДЕМІЧНА ДОБРОЧЕСНІСТЬ І ВИКОРИСТАННЯ ЗАСОБІВ ШІ

Курсова робота виконується самостійно відповідно до Кодексу академічної доброчесності та корпоративної культури ЛДУ БЖД. Пояснювальна записка і програмний код можуть перевірятися на наявність запозичень; рекомендовані показники оригінальності тексту записки: понад 70 % – текст оригінальний; 60–70 % – задовільно, необхідно перевірити оформлення цитувань і посилань; 40–60 % – робота приймається до захисту після доопрацювання; менше 40 % – робота до захисту не приймається. Встановлення факту академічного плагіату є підставою для повторного виконання роботи за іншим завданням.

Використання відкритих бібліотек, офіційних посібників (spring.io/guides) і фрагментів коду з документації допускається за умови посилання на джерело. Копіювання чужих проєктів, зокрема робіт інших здобувачів і демонстраційного проєкту «Реєстр укриттів», як основи власної роботи не допускається: демонстраційний проєкт є зразком підходів, а не шаблоном для заміни назв.

Відповідно до силабусу дисципліни **допускається використання інструментів штучного інтелекту** (зокрема асистентів програмування) для пошуку інформації, отримання пояснень, аналізу помилок і генерування ідей за умови критичного оцінювання результатів. Використання ШІ не повинно підміняти самостійну роботу: здобувач несе відповідальність за зміст і коректність поданого коду і тексту та має бути здатним пояснити кожен фрагмент коду під час захисту; нездатність пояснити код розглядається як ознака несамостійного виконання. У разі суттєвого використання ШІ в пояснювальній записці зазначають інструмент і мету його застосування.

СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

1. Віртуальне навчальне середовище ЛДУ БЖД «Віртуальний університет». Курс «WEB-технології та WEB-програмування» : лекції, методичні вказівки до лабораторних робіт [Електронний ресурс]. – Доступний з: https://virt.ldubgd.edu.ua/course/view.php?id=2111
2. Демонстраційний проєкт «Реєстр укриттів міста» [Електронний ресурс]. – Доступний з: https://github.com/SidiromUA/shelter-registry
3. Spring Boot Reference Documentation [Електронний ресурс]. – Доступний з: https://docs.spring.io/spring-boot/
4. Spring Guides [Електронний ресурс]. – Доступний з: https://spring.io/guides
5. Spring Data JPA Reference; Spring Security Reference [Електронний ресурс]. – Доступний з: https://docs.spring.io/
6. Thymeleaf Documentation [Електронний ресурс]. – Доступний з: https://www.thymeleaf.org/documentation.html
7. MDN Web Docs [Електронний ресурс]. – Доступний з: https://developer.mozilla.org/uk/
8. Web Content Accessibility Guidelines (WCAG) 2.2 [Електронний ресурс]. – Доступний з: https://www.w3.org/TR/WCAG22/
9. OWASP Top 10 [Електронний ресурс]. – Доступний з: https://owasp.org/www-project-top-ten/
10. Walls C. Spring in Action. 6th ed. – Shelter Island : Manning, 2022. – 520 p.
11. Schildt H. Java: The Complete Reference. 13th ed. – New York : McGraw Hill, 2024. – 1280 p.
12. Хорстманн К. С. Java. Бібліотека професіонала. Т. 1. Основи. 11-е вид. – Київ : Наука і техніка, 2020. – 864 с.
13. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання.

ДОДАТОК А

Титульний аркуш пояснювальної записки

ДЕРЖАВНА СЛУЖБА УКРАЇНИ З НАДЗВИЧАЙНИХ СИТУАЦІЙ
ЛЬВІВСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ БЕЗПЕКИ ЖИТТЄДІЯЛЬНОСТІ
Навчально-науковий інститут цивільного захисту
Кафедра інформаційних технологій та систем електронних комунікацій

КУРСОВА РОБОТА

з дисципліни «WEB-технології та WEB-програмування»
на тему: «Проєктування та розроблення вебсервісу на Spring Boot»
Індивідуальне завдання (напрям № ___):
«_____»
Вебсервіс: «_____»

Виконав (-ла): здобувач (-ка) групи КН-___
Ім’я ПРІЗВИЩЕ
Керівник: доцент кафедри ІТтаСЕК, к.т.н.
Роман ГОЛОВАТИЙ
Репозиторій: https://github.com/_____/_____
Оцінка: _____ балів, ЄКТС ___

Львів – 20___

ДОДАТОК Б

Контрольний перелік вимог до вебсервісу

Перед поданням роботи на перевірку позначте виконані вимоги. Перелік можна додати до README репозиторію.

✓	**Вимога**
☐	Тему, ролі і бізнес-правила погоджено з керівником
☐	Не менше 5 сутностей; зв’язки 1:N і M:N; обмеження цілісності в БД
☐	Spring Data JPA: похідні запити, щонайменше один JPQL, пагінація; без N+1 на основних сторінках
☐	Міграції Flyway (схема і тестові дані); `ddl-auto: none`
☐	Шари контролер – сервіс – репозиторій; DTO; впровадження через конструктор
☐	Щонайменше 3 бізнес-правила в сервісах, покриті модульними тестами
☐	Не менше 10 сторінок Thymeleaf з фрагментами макета; форми з валідацією і PRG
☐	REST API: CRUD для 2 ресурсів + вкладений ресурс; коди 200/201/204/400/401/403/404/409
☐	Swagger UI з описом ендпоінтів
☐	Централізована обробка помилок: Problem Details для API, сторінки помилок
☐	Spring Security: вхід через форму, HTTP Basic/токен для API, ≥ 2 ролі, BCrypt, CSRF
☐	Профілі dev і prod; секрети лише зі змінних оточення; журналювання; Actuator health
☐	Не менше 10 тестів (Mockito, @DataJpaTest, @WebMvcTest); `mvn test` проходить
☐	Dockerfile і docker-compose.yml; запуск командою `docker compose up`
☐	Lighthouse Accessibility ≥ 90 для основних сторінок
☐	README: опис, запуск, тестові облікові записи; тег `kursova`
☐	Пояснювальна записка оформлена за розділами 5–6; подано PDF і посилання у ВУ