

УНІВЕРСИТЕТ «КРОК»

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ІНФОРМАЦІЙНИХ ТА КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

МЕТОДИЧНІ НАСТАНОВИ  
до виконання здобувачами вищої освіти  
Проєкту четвертого рівня

ЗАВТЕРДЖЕНО  
на засіданні кафедри  
комп'ютерних наук  
протокол № 8 від 21.04.2022

Київ, 2022

Методичні настанови до виконання здобувачами вищої освіти Проєкту четвертого рівня / уклад. Єпик О.М., Тимчук О.С.; Київ, Університет «КРОК». 2022. 33 с.

Укладачі:

Єпик М.О., канд. техн. наук; Тимчук О.С., канд. техн. наук.

Рецензенти:

Стативка Ю.І., канд. техн. наук

Барибін О.І., канд. техн. наук

Проєкт четвертого рівня виконується здобувачами спеціальності 122 «Комп'ютерні науки» у відповідності до навчального плану 3 курсу у 8 семестрі.

*Метою виконання Проєкту є проєктування та реалізація програмного забезпечення на основі **архітектури клієнт-сервер**, включаючи сховище даних, логіку доступу до даних, логіку програми та логіку інтерфейсу.*

Клієнт має відповідати за логіку інтерфейсу, а сервер – за сховище даних та логіку доступу до даних.

Логіка програми може бути зосереджена як на клієнті (товстий клієнт), так і на сервері (тонкий клієнт).

Типовим прикладом програмного забезпечення з архітектурою клієнт-сервер є веб-застосунки, мобільні веб-застосунки або мобільні застосунки.

Для створення серверної частини програмного забезпечення можуть бути використані різноманітні серверні технології та мови програмування, наприклад: PHP, Python, Java, ASP, Node.js, Ruby.

Для створення клієнтської частини можуть бути використані HTML, CSS, JavaScript, Java, .NET.

## ПЕРЕДМОВА

Положення методичних настанов рекомендовані до застосовування під час навчання здобувачами освітнього ступеня «Бакалавр» на освітній програмі за спеціальністю 122 «Комп'ютерні науки» відповідно до графіку навчального процесу.

Текст методичних настанов містить положення, які розкривають рекомендовані дії здобувачів під час виконання Проєкту.

Для більш глибокого розуміння положень методичних настанов та з'ясування додаткової довідкової інформації здобувачам рекомендовано ознайомитись з текстами релевантних документів, які регламентують організацію освітнього процесу в Університеті «КРОК», регулюють відносини, що виникають між його учасниками, є невід'ємною складовою системи забезпечення якості освітньої діяльності та якості вищої освіти (системи внутрішнього забезпечення якості) Університету.

До таких основних документів належать:

- 1) Положення про організацію освітнього процесу в Університеті «КРОК» [1];
- 2) Положення про академічну доброчесність в Університеті «КРОК» [2];
- 3) Положення про перевірку академічних та наукових текстів на плагіат в університеті «КРОК» [3].

Для якісного написання пояснювальної записки, підготовки до презентації та захисту рекомендовано також ознайомитись з текстами державних стандартів, які встановлюють вимоги до оформлення наукових текстів та бібліографічного опису використаних джерел:

- 1) ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура та правила оформлювання [4];
- 2) ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні вимоги та правила складання [5].

## ЗМІСТ

ВСТУП .....	5
1 ТЕОРЕТИЧНІ ВІДОМОСТІ .....	7
1.1 Архітектура клієнт-сервер .....	7
1.2 Тестування .....	14
2 СТРУКТУРА ПРОЄКТУ .....	21
3 ПОРЯДОК ВИКОНАННЯ ПРОЄКТУ .....	22
2.1 Етапи виконання Проєкту .....	22
2.2 Реліз програмного забезпечення .....	22
2.3 Структура та вимоги до пояснювальної записки .....	23
2.4 Структура та вимоги до презентації .....	26
4 КРИТЕРІЇ ОЦІНЮВАННЯ ПРОЄКТУ .....	29
5 ДОТРИМАННЯ ВИМОГ АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ ПРИ ВИКОНАННІ ПРОЄКТУ .....	30
6 ТЕХНІЧНЕ ОБЛАДНАННЯ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ .....	34
7 ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	35
ДОДАТОК А .....	36

## ВСТУП

Проєкт четвертого рівня здобувач вищої освіти освітньо-професійної програми за спеціальністю 122 «Комп'ютерні науки» [6] реалізує у восьмому семестрі.

Робота над Проєктом передбачає встановлення відповідності засвоєних здобувачами рівня й обсягу знань, умінь, інших компетентностей до вимог стандарту вищої освіти першого (бакалаврського) рівня ступеня «бакалавр» за галуззю знань 12 «Інформаційні технології» спеціальністю 122 «Комп'ютерні науки» (затверджено та введено в дію наказом Міністерства освіти і науки України від 10.07.2019 р. № 962) [6].

У ході виконання Проєкту здобувачі мають продемонструвати свою здатність:

- 1) до системного мислення, застосування методології системного аналізу для дослідження складних проблем різної природи, методів формалізації та розв'язування системних задач, що мають суперечливі цілі, невизначеності та ризику;
- 2) проєктувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління;
- 3) реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах;
- 4) застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника;
- 5) до аналізу предметних областей різних галузей знань, збору детальної інформації для формалізації функціональних та нефункціональних вимог до програмного забезпечення, ідентифікації, класифікації та пошуку методів і підходів щодо їх розв'язання;
- 6) до написання дизайн-документів з використанням UML-нотації;
- 7) проводити тестування компонентів програмного забезпечення;

- 8) обирати та застосовувати процедури набуття та захисту прав інтелектуальної власності на власні розробки у галузі інформаційних технологій.

У результаті роботи над Проектом здобувачі вищої освіти мають навчитися:

- 1) демонструвати знання методів інформаційних технологій для розробки клієнт-серверного програмного забезпечення;
- 2) демонструвати знання особливостей проектування, розробки, дистрибуції та монетизації клієнт-серверного програмного забезпечення;
- 3) демонструвати вміння використовувати інструментальні засоби розробки для написання програмного коду клієнт-серверного програмного забезпечення;
- 4) демонструвати знання системи управління проектом і системи контролю версій;
- 5) демонструвати знання методів конфігурації клієнт-серверного програмного забезпечення;
- 6) демонструвати знання підходів до управління якістю програмного забезпечення;
- 7) проектувати та розробляти компоненти клієнтської частини застосунку;
- 8) проектувати та розробляти компоненти серверної частини застосунку;
- 9) розробляти протокол взаємодії між клієнтською і серверною частинами застосунку;
- 10) мотивовано обирати мови програмування та технології для створення клієнтської та серверної частини застосунку;
- 11) обґрунтувати необхідність застосування клієнт-серверних ІТ-рішень для вирішення конкретних професійних завдань;
- 12) використовувати принципи об'єктно-орієнтованого програмування при створенні клієнт-серверного програмного забезпечення.

Написання, підготовка до захисту та захист Проекту базується на таких основних принципах, як усвідомлення важливості академічної доброчесності та відповідальності за її порушення, а також нульової толерантності до порушення академічної доброчесності [2].

Термін виконання та захисту Проекту визначається графіком освітнього процесу.

Успішно захищені Проекти здобувачів оприлюднюються на офіційному сайті (або у репозиторії) університету.

## 1 ТЕОРЕТИЧНІ ВІДОМОСТІ

### 1.1 Архітектура клієнт-сервер

*Архітектура клієнт-сервер* є одним із архітектурних шаблонів програмного забезпечення та домінуючою концепцією у створенні розподілених мережевих застосунків і передбачає взаємодію та обмін даними між ними.

Дана архітектура передбачає такі основні компоненти:

- 1) набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- 2) набір клієнтів, які використовують сервіси, що надаються серверами;
- 3) мережа, яка забезпечує взаємодію між клієнтами та серверами.

*Сервери* є незалежними один від одного. *Клієнти* також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів.

Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися різних серверів.

Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Правила взаємодії між клієнтом і сервером називаються *протоколом обміну* (протоколом взаємодії) [10].

Модель клієнт-серверної взаємодії визначається, перш за все, розподілом обов'язків між клієнтом та сервером. Логічно можна виокремити три рівні операцій:

- 1) рівень представлення даних, який є інтерфейсом користувача, відповідає за представлення даних і введення керуючих команд користувачем;
- 2) прикладний рівень, який реалізує основну логіку застосунку, на якому здійснюється необхідна обробка інформації;
- 3) рівень керування даними, який забезпечує зберігання даних та доступ до них.

*Дволанкова клієнт-серверна архітектура* передбачає взаємодію двох програмних модулів – клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють [10]:

- 1) модель *тонкого клієнта*, у рамках якої вся логіка застосунку та керування даними зосереджена на сервері. Клієнтська програма забезпечує лише функції рівня представлення;
- 2) модель *товстого клієнта*, у рамках якої сервер лише керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

*Триланкова клієнт-серверна архітектура* передбачає відокремлення прикладного рівня від керування даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка застосунку. Програми проміжного рівня можуть функціонувати під керуванням спеціальних серверних застосунків, але запуск таких програм може здійснюватися і під керуванням звичайного веб-серверу. Нарешті, керування даними здійснюється сервером даних.

*Дволанкова архітектура* простіша, оскільки усі запити обслуговуються одним сервером, але саме через це вона менш надійна і висуває підвищені вимоги до продуктивності сервера.

*Триланкова архітектура* складніша, але завдяки тому, що функції розподілені між серверами другого і третього рівнів, ця архітектура має:

- 1) високий ступінь гнучкості і масштабованості;
- 2) високу безпеку (захист можна визначити для кожного сервісу або рівня);
- 3) високу продуктивність (завдання розподілені між серверами).

*Основна ідея* архітектури «клієнт-сервер» полягає у розподілі мережевого додатку на декілька компонентів, кожен з яких реалізує специфічний набір сервісів. Компоненти такого додатку можуть виконуватися на різних комп'ютерах, виконуючи серверні і/або клієнтські функції. Це дозволяє підвищити надійність, безпеку і продуктивність мережевих додатків і мережі у цілому.



Можна виділити наступні типи серверів, в залежності від ролі, яку вони виконують [10]:

1. **Веб-сервер** (*Web Server*): сервер, що приймає HTTP-запити від клієнтів, зазвичай веб-браузерів, видає їм HTTP-відповіді, зазвичай разом з HTML-сторінкою, зображенням, файлом, медіа-потокком або іншими даними. Веб-сервер – це основа Всесвітньої павутини. Веб-сервером називають як програмне забезпечення, що виконує функції веб-сервера, так і комп'ютер, на якому це програмне забезпечення працює. Клієнти дістаються веб-сервера за URL-адресою потрібної їм веб-сторінки або іншого ресурсу.
2. **Сервер застосунків** (*Application Server*): сервер, що виконує деякі прикладні програми. Термін також відноситься і до програмного забезпечення, що встановлено на такому сервері і забезпечує виконання прикладного ПЗ.
3. **Сервер баз даних**: сервери баз даних використовуються для обробки користувацьких запитів на мові SQL. При цьому СКБД знаходиться на сервері, до якого і підключаються клієнтські додатки.
4. **Файловий сервер** (*File Server*): сервер що зберігає інформацію у вигляді файлів і представляє користувачам доступ до неї. Як правило файл-сервер забезпечує і певний рівень захисту від несанкціонованого доступу.
5. **Сервер друку** (*Print Server*): сервер друку використовується для надання та керування доступом до принтерів. Роль серверу друку дозволяє керувати принтерами через веб-оглядач, друкувати через URL принтера, використовуючи протокол IPP, а також підключати принтери, використовуючи Point and Print.
6. **Поштовий сервер** (*Mail Server*): дозволяє обслуговувати базові поштові скриньки користувачів і дозволяє приймати і відправляти пошту з сервера. Вхідна пошта може зберігатися на сервері, а далі забиратися користувачем згідно з протоколу POP3. Для поштового серверу треба мати: активне з'єднання з інтернет, зареєстроване доменне ім'я, запис MX у провайдера для поштового домен.
7. **Термінальний сервер** (*Terminal Server*): сервер, що надає клієнтам обчислювальні ресурси (процесорний час, пам'ять, дисковий простір) для вирішення завдань. Технічно термінальний сервер – надпотужний комп'ютер (або кластер), підключений до мережі з термінальними клієнтами, у яких є, як правило, малопотужні або застарілі робочі станції або спеціальні рішення для доступу до термінального сервера.

Термінальний сервер слугує для віддаленого обслуговування користувача з наданням робочого столу.

8. *Remote Access/VPN Server*: сервер віддаленого доступу і VPN надають точку входу у мережу для віддалених користувачів. Використовуючи роль Remote Access/VPN Server, реалізуються протоколи маршрутизації для середовищ LAN і WAN. Ця роль підтримує модемні з'єднання і VPN через інтернет.
9. *DNS Server*: служба DNS дозволяє перетворювати доменні імена (FQDN) у адреси IP.
10. *DHCP Server*: сервер DHCP дозволяє клієнтам отримувати свій IP за потребою. Сервер DHCP також надає додаткову інформацію для конфігурації мережі – адреси серверів DNS, WINS і т.п.
11. *Streaming Media Server*: використовується для керування і доставки мультимедійного контенту (потокowego відео та аудіо) через інтранет або інтернет.
12. *Ігровий сервер (Game server)*: сервер, що забезпечує зв'язок між різними клієнтами, надаючи їм можливість комунікації один з одним у межах програмної оболонки конкретної гри.

## 1.2 Тестування

*Модульне тестування (Unit testing)* – це метод тестування програмного забезпечення, який полягає в окремому тестуванні кожного модуля коду програми.

*Модулем* називають найменшу частину програми, яка може бути протестованою.

*Check-list* (контрольний список, що містить ряд необхідних перевірок для тестування) для тестування програмного продукту (Web-додаток) складається з:

- 1) тестування зручності використання;
- 2) функціонального тестування;
- 3) тестування сумісності;
- 4) тестування бази даних;
- 5) тестування безпеки;
- 6) тестування продуктивності.

**Тестування зручності використання** (юзабіліті) – це тестування доброзичливості додатку для користувача. При тестуванні зручності використання перевіряється, чи легко новому користувачеві розібратися в додатку і тестується системна навігація.

**Мета тестування зручності використання:** впевнитися у простоті і ефективності використання продукту при використанні стандартних практик тестування зручності використання.

Сценарії тестування **зручності використання:**

- 1) вміст web-сторінки вірний, без граматичних і орфографічних помилок;
- 2) всі шрифти відповідають вимогам;
- 3) всі тексти правильно вирівняні;
- 4) всі повідомлення про помилки вірні, без орфографічних і граматичних помилок, і відповідають заголовку вікна;
- 5) підказки існують для всіх полів;
- 6) всі поля правильно вирівняні;
- 7) між полями, колонками, рядами і повідомленнями про помилки залишено досить вільного місця;
- 8) всі кнопки повинні мати стандартний формат і розмір;
- 9) посилання на домашню сторінку має бути на кожній сторінці сайту;
- 10) неактивні поля мають бути сірими;
- 11) перевірте, що на сайті немає битих посилань і зображень;
- 12) підтверджуючі повідомлення повинні відобразитися для всіх операцій оновлення і видалення;
- 13) перевірте сайт при різних роздільних здатностях екрану (640 x 480, 600x800 і т. д.);
- 14) перевірте, що користувач може користуватися системою без роздратування;
- 15) перевірте, що *Tab* правильно працює;
- 16) панель скролу повинна з'являтися лише тоді, коли вона потрібна;
- 17) якщо при відправці форми є повідомлення про помилку, у ньому має міститися інформація, передана користувачем;
- 18) заголовок повинен відобразитися на кожній сторінці;
- 19) всі поля (текстові, випадаючі меню, радіо-кнопки і т. д.) і кнопки мають бути доступні з клавіатури, і користувач має бути в змозі користуватися сайтом, використовуючи лише клавіатуру;
- 20) переконайтеся, що дані у випадаючих списках не обрізуються із-за розмірів поля, і перевірте, чи зашиті дані в код або керуються адміністратором.

**Функціональне тестування** – тестування функціональностей і операційної поведінки продукту з метою переконатися, що вони відповідають специфікаціям. Даний вид тестування ігнорує внутрішні механізми системи або компоненту і концентрується виключно на вихідних даних, отриманих у відповідь на призначене для користувача введення і умови виконання сценаріїв.

**Мета функціонального тестування:** переконатися, що продукт відповідає потрібній функціональній специфікації, згаданій у документації з розробки.

Сценарії **функціонального тестування:**

- 1) протестуйте валідацію всіх обов'язкових полів;
- 2) переконайтеся, що знак зірочки відображається у всіх обов'язкових полях;
- 3) переконайтеся, що система не відображає вікно помилки при незаповнених необов'язкових полях;
- 4) переконайтеся, що коди коректно валідуються і не викликають помилок в розрахунках;
- 5) протестуйте числові поля: вони не повинні приймати літери, у випадку введення літери повинне відобразитися відповідне повідомлення про помилку;
- 6) протестуйте від'ємні значення в числових полях, якщо вони дозволені;
- 7) протестуйте, що ділення на нуль вірно обраховується;
- 8) протестуйте максимальну довжину кожного поля, щоб переконатися, що дані не обрізуються;
- 9) протестуйте спливаюче повідомлення («Це поле обмежене 500 знаками»), яке повинне відобразитися, якщо введені дані перевищують дозволений розмір поля;
- 10) переконайтеся, що підтверджуюче повідомлення відображається для операцій оновлення і видалення;
- 11) переконайтеся, що значення вартості відображуються в потрібній валюті;
- 12) протестуйте всі поля введення на спецсимволи;
- 13) протестуйте функціональність тайм-ауту;
- 14) протестуйте функціональність сортування;
- 15) протестуйте функціональність доступних кнопок;
- 16) протестуйте умови використання і питання, що часто ставляться: вони мають бути виразними і доступними користувачеві;
- 17) протестуйте, що при відмові функціональності користувач перенаправляється на спеціальну сторінку помилки;

- 18) протестуйте, що всі завантажені документи правильно відкриваються;
- 19) протестуйте, що користувач може скачати завантажені файли;
- 20) протестуйте поштову функціональність системи;
- 21) протестуйте, що Java Script вірно працює в різних браузерях (IE, Firefox, Chrome, Safari, Opera);
- 22) поглянете, що буде, якщо користувач видалить куки (cookies-файли), знаходячись на сайті;
- 23) поглянете, що буде, якщо користувач видалить куки після відвідування сайту;
- 24) протестуйте всі дані у випадючих списках: вони мають бути розташовані в хронологічному порядку.

*Тестування сумісності* перевіряє сумісність додатку з іншими елементами системи, в якій воно працює, – наприклад, браузерами, операційними системами або залізом.

*Мета тестування сумісності:* оцінка того, наскільки добре ПЗ працює в певному браузері, під певною ОС, з іншим ПЗ або залізом.

Сценарії *тестування сумісності:*

- 1) протестуйте сайт в різних браузерах (IE, Firefox, Chrome, Safari, Opera) і переконайтеся, що сайт правильно відображається;
- 2) переконайтеся, що використовувана версія HTML сумісна з відповідними версіями браузерів;
- 3) переконайтеся, що малюнки коректно відображаються в різних браузерах;
- 4) переконайтеся, що шрифти вірно відображуються в різних браузерах;
- 5) переконайтеся, що Java Script код працює в різних браузерах;
- 6) перевірте анімовані GIF в різних браузерах.

Інструмент для тестування сумісності *Spoon.net*: Spoon.net надає доступ до тисяч додатків (браузерів), що не вимагають установки. Цей інструмент допомагає тестувати додаток в різних браузерах на одній машині.

*Тестування бази даних* – перевіряються бекенд-записи, введені через web-додаток або десктоп-додаток. Дані, які відображаються у додатку, повинні збігатися з даними, що зберігаються в базі даних.

Що має знати тестувальник при тестування БД:

- 1) тестувальник має розуміти функціональні вимоги, бізнес-логіку, основний сценарій додатку і дизайн бази даних;
- 2) тестувальник має знатися на таблицях, тригерах, процедурах зберігання, способах відображення і покажчиках, використовуваних для додатку;
- 3) тестувальник має розуміти логіку тригерів, процедур зберігання, способів відображення і покажчиків;
- 4) тестувальник має розуміти, які таблиці зачіпаються, коли операції вставки, оновлення і видалення виконуються в додатку.

Сценарії тестування *бази даних*:

- 1) перевірте назву бази даних: вона повинна збігатися із специфікацією;
- 2) перевірте таблиці, колонки, типи колонок і значення за умовчанням: все це має збігатися із специфікацією;
- 3) перевірте, чи дозволяє колонка значення *null*;
- 4) перевірте первинний і зовнішній ключ кожної таблиці;
- 5) перевірте процедури зберігання;
- 6) протестуйте, чи встановлена процедура зберігання;
- 7) перевірте назву процедури зберігання;
- 8) перевірте назви параметрів, їх типи і кількість;
- 9) перевірте, обов'язкові параметри чи ні;
- 10) перевірте процедуру зберігання, видаливши деякі параметри;
- 11) перевірте базу даних, якщо на виході нуль – записи з нулем мають бути задіяні;
- 12) перевірте процедуру зберігання, задавши прості SQL-запити;
- 13) переконайтеся, що процедура повертає значення;
- 14) перевірте процедуру введенням тестових даних;
- 15) перевірте поведінку кожного прапора у таблиці;
- 16) переконайтеся, що дані правильно зберігаються в базі даних після кожного введення;
- 17) перевірте дані під час кожної операції оновлення, видалення і вставки;
- 18) перевірте довжину кожного поля: довжина на бекенді і фронтенді мають збігатися;
- 19) перевірте назви баз даних QA, UAT і продю, імена мають бути унікальними;
- 20) перевірте зашифровані дані в БД;
- 21) перевірте розмір БД і час відгуку на кожен запит;
- 22) перевірте дані, що відображаються на фронтенді, і переконайтеся, що вони збігаються з бекендом;
- 23) перевірте цілісність даних, вводючи невалідні значення в БД;
- 24) перевірте тригери.

*Тестування безпеки* націлено на пошук недоліків і пропусків з точки зору безпеки додатку.

Сценарії *тестування безпеки*:

- 1) переконайтеся, що сторінки, що містять важливі дані (пароль, номер кредитної картки, відповіді на секретні питання і т. п.) відкриваються через HTTPS (SSL);
- 2) переконайтеся, що важлива інформація (пароль, номер кредитної картки) відображається у зашифрованому вигляді;
- 3) переконайтеся, що правила створення паролів упроваджені на всіх сторінках авторизації (реєстрація, сторінка «забули пароль», зміна паролю);
- 4) переконайтеся, що якщо пароль змінений, користувач не може зайти під старим паролем;
- 5) переконайтеся, що повідомлення про помилки не містять жодної секретної інформації;
- 6) переконайтеся, що якщо користувач вийшов з системи або сесія завершена, він не може користуватися сайтом;
- 7) перевірте доступ до закритих і відкритих сторінок сайту безпосередньо без авторизації;
- 8) переконайтеся, що опція «Перегляд початкового коду» відключена і її не бачить користувач;
- 9) переконайтеся, що обліковий запис користувача блокується, якщо він кілька разів ввів пароль невірно;
- 10) переконайтеся, що пароль не зберігається в кукі;
- 11) переконайтеся, що якщо яка-небудь функціональність не працює, система не відображає інформацію про додаток, сервер або базу даних; замість цього відображається відповідне повідомлення про помилку;
- 12) перевірте сайт на SQL-ін'єкції;
- 13) перевірте права користувачів і їх ролі (наприклад, кандидат не має бути здатний дістати доступ до сторінки адміністратора);
- 14) переконайтеся, що важливі операції пишуться в логи і інформацію можна відстежити;
- 15) переконайтеся, що значення сесій відображаються в адресному рядку у зашифрованому вигляді;
- 16) переконайтеся, що кукі зберігаються в зашифрованому вигляді;
- 17) перевірте додаток на стійкість до брутфорс-атак (брутфорс атака (brute force) – метод підбору паролю, шляхом почергового перебору можливих комбінацій. Зазвичай брутфорс виробляється масово з

використанням стандартних комбінацій логінів (admin, administrator) і паролів (осмислені словосполучення, словники)).

*Тестування продуктивності* – проводиться для оцінки відповідності системи або компоненту специфічним вимогам до продуктивності.

Сценарії *тестування продуктивності*:

- 1) визначення продуктивності, стабільності і масштабованості додатку під різним навантаженням;
- 2) визначення, чи може актуальна архітектура підтримувати додаток при пікових навантаженнях;
- 3) визначення, яка конфігурація призводить до найкращих показників продуктивності;
- 4) визначення, чи не змінився час відгуку в новій версії додатку;
- 5) визначення пляшкового горла додатку і інфраструктури;
- 6) оцінка продукту і заліза з метою упевнитися, що вони витримають прогнозовані обсяги навантаження.

Неможливо провести *тестування продуктивності вручну* з ряду причин:

- 1) знадобиться велика кількість ресурсів;
- 2) неможливо одночасно здійснювати ряд дій;
- 3) відсутній відповідний спосіб відстежування поведінки системи;
- 4) складність виконання завдань, що повторюються.

Щоб впоратися з вище переліченими проблемами, використовуються спеціальні інструменти тестування продуктивності:

- 1) Apache Jmeter;
- 2) Load Runner;
- 3) Borland Silk Performer;
- 4) Rational Performance Tester;
- 5) WAPT;
- 6) NEO LOAD.



**2 СТРУКТУРА ПРОЄКТУ**

Обсяг	240 годин / 8 кредитів ECTS
Виконання	індивідуальне
Кількість тижнів	8
Кількість спринтів	9
Обов'язкові складові результату виконання проєкту	<ul style="list-style-type: none"><li>- реліз програмного забезпечення</li><li>- пояснювальна записка</li><li>- презентація продукту + текст доповіді</li><li>- спланований сценарій демонстрації роботи програмного забезпечення</li></ul>

### 3 ПОРЯДОК ВИКОНАННЯ ПРОЄКТУ

#### 3.1 Етапи виконання Проєкту

№	Спринт	Дедлайн	Результат
1	Вибір теми проєкту	до 12.01.202X	Затверджений лист завдання: - тема проєкту пропонується здобувачем при попередньому погодженні з керівником проєкту; - рекомендовано обирати тему проєкту близьку до теми бакалаврської роботи
2	Дослідження ринку та аналіз вимог	до 18.01.202X	Документ з пропозиціями щодо нового продукту
3	Проектування взаємодії з користувачем	до 22.01.202X	Візуалізовані приклади використання, спрощені та деталізовані прототипи продукту
4	Проектування рівня доступу до даних	до 26.01.202X	ER-діаграми рівня концептуальної, логічної та фізичної моделей даних
5	Проектування програмного забезпечення	до 7.02.202X	Файли з діаграмами та схемами компонент програмного забезпечення
6	Розробка програмного забезпечення	до 21.02.202X	Розроблений програмний продукт
7	Тестування програмного забезпечення	до 25.02.202X	Перелік помилок до виправлення
8	Підготовка пояснювальної записки та презентації продукту	до 03.03.202X	Пояснювальна записка
9	Захист	до 05.03.202X	Підсумкова оцінка за виконання проєкту

#### 3.2 Реліз програмного забезпечення

Реліз програмного забезпечення повинен відповідати перерахованим нижче вимогам:

- 1) мати клієнтську і серверну частини;
- 2) мати базові програмні компоненти: сховище даних, рівень доступу до даних, рівень програми та рівень інтерфейсу;
- 3) мати систему логування подій;
- 4) мати часткове покриття *unit*-тестами;
- 5) вихідний код програми відповідати стандарту оформлення коду;

- 6) мати відповідні коментарі або *doc-strings* до програмних елементів;
- 7) проєкт розміщено в репозиторії на GitHub;
- 8) проєкт розгорнуто на локальному сервері.

### 3.3 Структура та вимоги до пояснювальної записки

Під час підготовки пояснювальної записки треба дотримуватися основних вимог до її змісту.

Пояснювальна записка умовно поділяється на:

- 1) вступну частину;
- 2) основну частину;
- 3) додатки.

Вступна частина містить такі структурні елементи:

- 1) титульний аркуш;
- 2) реферат.

Основна частина містить такі структурні елементи:

- 1) вступ;
- 2) змістовну частину звіту (суть пояснювальної записки);
- 3) висновки;
- 4) перелік джерел посилання.

Рекомендований обсяг пояснювальної записки не повинен перевищувати 50 сторінок.

Короткі настанови до обов'язкових структурних елементів пояснювальної записки наведені у таблиці 1.

Таблиця 1 – Настанови до обов'язкових структурних елементів пояснювальної записки до Проєкту четвертого рівня

Назва розділу	Назва підрозділу	Вимоги та обов'язкові елементи
Титульний лист	-	Див. у Додаток А
Лист завдання	-	-

<b>Календарний план виконання проєкту</b>	-	-
<b>Реферат</b>	-	- не більше 1 сторінки - українською та англійською мовами
<b>Зміст</b>	-	-
<b>Вступ</b>	-	- актуальність теми - мета і задачі проєкту - практична цінність проєкту - структура та обсяг пояснювальної записки
<b>Аналіз предметної області і постановка задачі</b>	Проблеми розвитку інформаційних технологій	- відомі теоретичні і практичні результати, які отримані іншими вітчизняними та іноземними авторами за напрямом роботи, із посиланнями на їх праці
	Аналіз існуючих інформаційних технологій	- можливості, переваги і недоліки відомих інформаційних технологій, які в повній мірі або частково вирішують задачі, які будуть закладені у новий продукт
	Висновки за розділом	- результати аналізу предметної області - постановка задачі
<b>Проектування застосунок</b>	Аналіз варіантів використання	- функціональні та нефункціональні вимоги до нового продукту - діаграма прецедентів ( <i>Use case diagram</i> )
	Моделювання процесів	- діаграми діяльності ( <i>Activity diagram</i> ) - математична модель процесів (за наявності)
	Моделювання даних	- ідентифіковані сутності, атрибути сутностей та відношення між сутностями - ER-діаграма ( <i>Entity relationship diagram</i> ) - нормалізація ER-діаграми (приведення до третьої нормальної форми)
	Проектування клієнт-серверної архітектури	- ідентифіковані програмні компоненти: сховище даних, рівень доступу до даних, рівень програми та рівень інтерфейсу - ідентифіковані апаратні компоненти: клієнтські комп'ютери, сервери та мережеве середовище - схема розподілу програмних компонентів між апаратними компонентами - протокол взаємодії між клієнтською і серверною частинами - вимоги: експлуатаційні, до виконання, безпеки, культурні та політичні
	Проектування структури	- діаграма навігації по екранах ( <i>User flow diagram</i> )

	клієнтської частини	<ul style="list-style-type: none"> <li>- діаграма структури інтерфейсу (<i>Interface structure diagram</i>)</li> <li>- вайрфрейми (<i>Wireframe</i>)</li> <li>- прототип інтерфейсу (<i>UI prototype</i>)</li> <li>- для товстих клієнтів – діаграми класів (<i>Class diagram</i>), об'єктів (<i>Object diagram</i>), послідовності (<i>Sequence diagram</i>)</li> </ul>
	Проектування структури серверної частини	<ul style="list-style-type: none"> <li>- для всіх рівнів серверної частини – діаграми класів (<i>Class diagram</i>), об'єктів (<i>Object diagram</i>), послідовності (<i>Sequence diagram</i>), станів (<i>State machine diagram</i>)</li> </ul>
	Висновки за розділом	<ul style="list-style-type: none"> <li>- результати етапу проектування</li> </ul>
<b>Реалізація застосунку</b>	Реалізація клієнтської частини	<ul style="list-style-type: none"> <li>- опис засобів реалізації клієнтської частини</li> <li>- опис деталей реалізації клієнтської частини</li> </ul>
	Конструювання	<ul style="list-style-type: none"> <li>- діаграма компонентів (<i>Component diagram</i>)</li> <li>- діаграма пакетів (<i>Package diagram</i>)</li> <li>- діаграма розгортання (<i>Deployment diagram</i>)</li> </ul>
	Реалізація серверної частини	<ul style="list-style-type: none"> <li>- опис засобів реалізації серверної частини</li> <li>- опис реалізації сховища даних: для ORM-технології – модель даних та сценарій міграції; для non-ORM-технології – таблиці (назва, поля із типами даних, ключі, обмеження) та повна схема даних у вибраній системі управління базами даних</li> <li>- опис деталей реалізації рівня доступу до даних</li> <li>- опис деталей реалізації рівня програми</li> </ul>
	Документування	<ul style="list-style-type: none"> <li>- схема організації довідкових технічних документів</li> <li>- схема організації користувацьких документів</li> </ul>
	Адміністрування	<ul style="list-style-type: none"> <li>- опис панелі адміністратора</li> <li>- інструкції щодо інсталяції програмного забезпечення</li> <li>- інструкції щодо конфігурації елементів програмного забезпечення</li> <li>- інструкції щодо адміністрування даних</li> <li>- інструкції щодо підсистеми зберігання інформації про важливі програмні і апаратні події (<i>Event logging system</i>)</li> </ul>
	Тестування	<ul style="list-style-type: none"> <li>- модульне тестування</li> <li>- функціональне тестування</li> </ul>

	Висновки за розділом	- результати етапу реалізації
<b>Висновки</b>	-	- стислий виклад результатів вирішення поставленого завдання, що отримані в процесі аналізу предметної області, проектування та реалізації програмного забезпечення - відомості про достовірність отриманих результатів - не більше 1 сторінки
<b>Список посилань</b>	-	- перелік використаних джерел у порядку згадування по тексту мовою оригіналу джерел відповідно до вимог ДСТУ 8302:2015 - не менше 20 найменувань
<b>Додатки</b>	-	- екрани застосунку - фрагменти лістингу

**Вимоги до оформлення пояснювальної записки:**

1. Формат аркушу - А4.
2. Обсяг основного тексту пояснювальної записки не повинен перевищувати 50 сторінок; загальний обсяг не повинен перевищувати 70 сторінок.
3. Оформлення пояснювальної записки повинно відповідати вимогам стандарту ДСТУ 3008:2015 «Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання» [4].
4. Оформлення списку посилань повинно відповідати вимогам стандарту ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання» [5].
5. Вимоги до оформлення основного тексту:
  - a. параметри сторінки (поля): ліве – 2,5 см, праве – 1,5 см, верхнє і нижнє – 2 см;
  - b. основний шрифт – Times New Roman, 14 пт;
  - c. міжрядковий інтервал – 1,5;
  - d. інтервал до та після рядків – 0;
  - e. абзацний відступ – 1,25 см;
  - f. вирівнювання тексту - по ширині.
6. Кожний розділ починається з нової сторінки.
7. Назва розділу оформлюється у верхньому регістрі.
8. Номер рисунків та таблиць складається з двох цифр, перша – номер розділу, в якому розташований рисунок або таблиця, друга – порядковий номер рисунку або таблиці у розділі.

9. Рисунки та таблиці розміщуються після першого посилання на них у тексті (рис. 1.1 або табл. 1.1). Після підпису рисунку або таблиці обов'язково наводяться дані про джерело, звідки вони взяті (розроблено автором (-ами) / розроблено автором (-ами) за даними [номер посилання із зазначенням номеру сторінки] / [номер посилання із зазначенням номеру сторінки]).
10. Нумерація сторінок – внизу по центру.
11. Вимоги до оформлення тексту додатків:
- a. параметри сторінки (поля): ліве – 2,5 см, праве – 1,5 см, верхнє і нижнє – 2 см;
  - b. основний шрифт – Courier New, 12 пт;
  - c. міжрядковий інтервал – 1;
  - d. інтервал до та після рядків – 0;
  - e. абзацний відступ – 0;
  - f. вирівнювання тексту – по лівому краю.

### **3.4 Структура та вимоги до презентації**

Презентація має містити наступні слайди:

- 1) титульний слайд;
- 2) аналіз предметної області;
- 3) постановка задачі;
- 4) аналіз варіантів використання;
- 5) діаграми діяльності;
- 6) концептуальна модель даних;
- 7) логічна модель даних;
- 8) клієнт-серверна архітектура;
- 9) дизайн клієнтської частини;
- 10) дизайн серверної частини;
- 11) засоби реалізації клієнтської та серверної частин;
- 12) фізична модель даних;
- 13) конструювання застосунку;
- 14) система адміністрування;
- 15) екрани застосунку;
- 16) висновки.

Короткі настанови до обов'язкових структурних елементів презентації наведені у таблиці 2.

Таблиця 2 – Настанови до обов'язкових структурних елементів презентації до Проєкту четвертого рівня

Назва слайду (ів)	Вимоги та обов'язкові елементи
Титульний слайд	- назва продукту
Аналіз предметної області	- актуальність теми - мета і задачі проєкту
Постановка задачі	- стисло перелік основних задач - опис вхідних і вихідних даних
Аналіз варіантів використання	- список функціональних та нефункціональних вимоги до нового продукту - діаграма прецедентів ( <i>Use case diagram</i> )
Діаграми діяльності	- діаграма діяльності ( <i>Activity diagram</i> )
Концептуальна модель даних	- діаграми відношень ( <i>Entity relationship diagram</i> )
Логічна модель даних	- діаграми відношень ( <i>Entity relationship diagram</i> )
Клієнт-серверна архітектура	- ідентифіковані програмні і апаратні компоненти - протокол взаємодії між клієнтською і серверною частинами
Дизайн клієнтської частини	- UML-діаграми рівня проєктування програмного забезпечення
Дизайн серверної частини	- UML-діаграми рівня проєктування програмного забезпечення
Засоби реалізації клієнтської та серверної частини	- перелік засобів реалізації програмного забезпечення
Фізична модель даних	- повна схема даних у вибраній системі управління базами даних
Конструювання застосунку	- діаграма компонентів ( <i>Component diagram</i> ) - діаграма пакетів ( <i>Package diagram</i> ) - контракт взаємодії між клієнтською і серверною частинами
Система адміністрування	- панель адміністратора - інструкції щодо адміністрування даних - система логування подій
Екрани застосунку	- головне вікно застосунку - дочірні вікна застосунку
Висновки	- найбільш важливі практичні результати, одержані під час виконання проєкту - якісні та кількісні показники здобутих результатів

### **Вимоги до оформлення презентації:**

1. Для підготовки презентації рекомендовано використовувати корпоративний шаблон презентації, який доступний за адресою <https://www.krok.edu.ua/download/brand-book/krok-template-presentation.pptx>.
2. Всі слайди (крім першого) повинні містити порядковий номер, розташований у правому нижньому кутку.



3. Кожен слайд (крім першого) повинен мати коротку назву (заголовок, без крапки в кінці).
4. Вимоги до оформлення тексту презентації:
  - a. шрифт заголовків – Calibri, 36 пт, напівжирний;
  - b. шрифт порядкових номерів слайдів - Calibri, 20 пт, напівжирний;
  - c. шрифт основного тексту - Calibri, 22 пт;
  - d. міжрядковий інтервал – 1;
  - e. інтервал до та після рядків – 0;
  - f. абзацний відступ – 0 см;
  - g. вирівнювання тексту - по лівому краю.

#### 4 КРИТЕРІЇ ОЦІНЮВАННЯ ПРОЄКТУ

Підсумкове оцінювання за виконання Проєкту, яке відображає рівень досягнення запланованих результатів навчання, проводиться за 100-бальною накопичувальною шкалою.

Підсумкова оцінка розраховується за формулою:

$$G = 0,5 \cdot G_{team} + 0,3 \cdot G_{report} + 0,2 \cdot G_{presentation}$$

де

$G_{team}$  – середня оцінка членів команди, у т.ч. *Product Owner*,

$G_{report}$  – оцінка за пояснювальну записку,

$G_{presentation}$  – оцінка за презентацію продукту.

Для оцінювання рівня з  $G_{team}$ ,  $G_{report}$ , та  $G_{presentation}$  використовується 12-бальна оцінна шкала з подальшим переведенням у 100-бальну шкалу:

Рівні оцінної шкали	Відмітки	Інтерпретація рівня досягнення результатів навчання курсу
IV високий	12	Чудово
	11*	Відмінно
	10	майже відмінно
III середній	9	більш ніж добре
	8*	Добре
	7	майже добре
II достатній	6	більш ніж задовільно
	5*	Задовільно
	4	майже задовільно
I початковий	3	мало задовільно
	2*	Незадовільно
	1	знання майже відсутні

\* – базова оцінка рівня оцінної шкали

## 5 ДОТРИМАННЯ ВИМОГ АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ ПРИ ВИКОНАННІ ПРОЄКТУ

До початку виконання Проєкту необхідно уважно ознайомитися з Положенням про академічну доброчесність в Університеті КРОК [2], Положенням про перевірку академічних та наукових текстів на плагіат в університеті «КРОК» [3], Рекомендаціями МОН України щодо запобігання академічному плагіату та його виявлення в наукових роботах (авторефератах, дисертаціях, монографіях, наукових доповідях, статтях тощо) [9].

Основні види порушень академічної доброчесності, яких слід уникати під час виконання Проєкту, написання пояснювальної записки, підготовки презентації [2]:

**плагіат** – оприлюднення (опублікування), повністю або частково, чужого твору під іменем особи, яка не є автором цього твору (пункт в ст. 50 Закону України «Про авторське право і суміжні права»);

**академічний плагіат** – оприлюднення (частково або повністю) наукових (творчих) результатів, отриманих іншими особами, як результатів власного дослідження (творчості) та/або відтворення опублікованих текстів (оприлюднених творів мистецтва) інших авторів без зазначення авторства (ч. 4 ст. 42 Закону України «Про освіту»);

**самоплагіат** – оприлюднення (частково або повністю) власних раніше опублікованих наукових результатів як нових наукових результатів;

**фабрикація** – вигадкування даних чи фактів, що використовуються в освітньому процесі або наукових дослідженнях;

**фальсифікація** – свідомо зміна чи модифікація вже наявних даних, що стосуються освітнього процесу чи наукових досліджень.

При використанні фрагментів, запозичених з інших джерел, необхідно прийняти до уваги, що вони будуть **ідентифіковані як плагіат** за умови, якщо будуть відтворені у тексті роботи [9]:

- 1) без змін, з незначними змінами, або в перекладі, обсягом від речення і більше, без посилання на автора (авторів) відтвореного тексту;
- 2) повністю або частково, через перефразування чи довільний переказ без посилання на автора (авторів) відтвореного тексту;
- 3) як цитати з третіх джерел без вказування, за яким саме безпосереднім джерелом наведена цитата;
- 4) як наведена в іншому джерелі науково-технічна інформація (крім загальновідомої) без вказування на те, з якого джерела взята ця інформація;

- 5) як оприлюднені твори мистецтва без зазначення авторства цих творів мистецтва.

У контексті технічної (на текстові збіги) та експертної перевірки (керівниками) повного тексту роботи плагіатом буде вважатись наступне [3]:

- 1) видання виконаної іншим автором роботи за свою без внесення в неї жодних змін;
- 2) дослівне копіювання фрагментів тексту (від фрази до набору речень) без належного оформлення цитування;
- 3) внесення незначних правок у скопійований матеріал (переформулювання речень, зміна порядку слів у них тощо) та без належного оформлення цитування;
- 4) представлення суміші власних і запозичених аргументів без належного цитування;
- 5) *парафраза* – переказ своїми словами чужих думок, ідей або тексту; сутність парафрази полягає в заміні слів (знаків), фразеологічних зворотів або пропозицій при використанні будь-якої авторської наукової праці (збереженої на електронних або паперових носіях, у тому числі розміщеної в мережі Інтернет);
- 6) *компіляція* – створення значного масиву тексту без поглибленого вивчення проблеми шляхом копіювання тексту із низки джерел без внесення в нього правок, з посиланням на авторів та «маскуванням» шляхом написання перехідних речень між скопійованими частинами тексту.

Положенням про перевірку академічних та наукових текстів на плагіат в університеті «КРОК» передбачено виявлення рівня оригінальності тексту представленої роботи у відсотках, визначених за допомогою програмно-технічних засобів перевірки на текстові збіги. За результатами такої перевірки передбачені наступні дії:

- 1) при виявленні високого рівня оригінальності роботи (понад 75%) текст вважається оригінальним, додаткові дії щодо запобігання неправомірним запозиченням не проводитимуться;
- 2) при виявленні задовільного рівня оригінальності роботи (55-75%) вважається, що в тексті роботи наявні окремі ознаки академічного плагіату. Додатково електронний звіт з результатами антиплагіатної перевірки з гіперпосиланнями на запозичені джерела буде експертно перевірений (науковим керівником) на наявність посилань на першоджерела для цитованих фрагментів, в разі необхідності буде проведена повторна технічна перевірка;

- 3) при виявленні низького рівня оригінальності твору (35-55%) вважається, що в тексті роботи наявні певні ознаки академічного плагіату, проте робота може бути прийнята за умови доопрацювання з обов'язковою наступною перевіркою на оригінальність;
- 4) при виявленні рівня оригінальності роботи нижче за 35%, вважається, що в тексті роботи наявні істотні ознаки плагіату. Робота до розгляду (захисту) не приймається.

У разі виявлення академічного плагіату у Проєкті, він знімається із захисту без права повторного захисту.

Відсоток рівня оригінальності роботи може бути збільшений за рахунок правомірних запозичень, до яких належать:

- 1) власні назви (індивідуальні найменування окремих одиничних об'єктів, у т.ч. найменування установ, назви праць, які досліджувалися в роботі, бібліографічні посилання на джерела та ін.);
- 2) усталені словосполучення, що характерні для сфери ІТ, комп'ютерних наук;
- 3) належним чином оформлені цитування;
- 4) самоцитування (фрагменти текстів, що належать автору роботи, опубліковані або оприлюднені в електронній формі ним у інших творах).

Під час виконання Проєкту та написання пояснювальної записки, необхідно переконатися, що посилання у тексті пояснювальної записки зроблені правильно.

Посилання у тексті робиться при цитуванні джерела чи думки дослідника, при вказівці на певне важливе свідчення джерела, при запозиченні положень, використанні фактичного матеріалу, результатів досліджень інших авторів, посилань на досвід.

Кожну цитату обов'язково супроводжувати посиланням на джерело у тексті. Позначається таке посилання порядковим номером за переліком джерел посилання у клямрах (наприклад, «... у працях [1–7]...»).

Якщо використовуються відомості, матеріали з монографій, оглядових статей, інших джерел з великою кількістю сторінок, тоді у посиланні необхідно точно вказати номери сторінок, ілюстрацій, таблиць, формул з джерела, на які є посилання (наприклад, [3, с. 29]).

Якщо текст цитується не за першоджерелом, а за іншим виданням чи документом, то необхідно розпочинати посилання словами: «Цит. за:».

При непрямому цитуванні (переказі, викладенні думок інших авторів своїми словами), що дає значну економію тексту, треба бути гранично точними, коректними.

Необхідно уникати як надмірного, так і недостатнього цитування, оскільки це знижує рівень: надмірне цитування створює враження компілятивності праці, а недостатнє – знижує цінність викладеного матеріалу.

Якщо наводиться цитата і треба виділити у ній деякі слова, необхідно зробити спеціальний припис (після тексту, який пояснює виділення, треба поставити крапку, далі дефіс і вказати ініціали автора роботи, а весь текст припису помістити у дужки) (наприклад, (курсив наш – А.А.), (підкреслено мною – А.А.), (доповнено мною – А.А.)).

Бібліографічний опис використаних джерел оформлюйте відповідно до вимог ДСТУ 8302:2015 [5].

Відповідальність за перевірку Проєкту у встановлені терміни, прийняття рішення щодо доопрацювання та повторну перевірку на плагіат, про допуск до захисту несе Керівник Проєкту (Product Owner).

## 6 ТЕХНІЧНЕ ОБЛАДНАННЯ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

Технічне обладнання: комп'ютер/ноутбук.

Програмне забезпечення:

- 1) інтегроване середовище розробки (обирається індивідуально);
- 2) система керування базою даних (обирається індивідуально);
- 3) VCS GIT (<https://git-scm.com/>).

## 7 ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Положення про організацію освітнього процесу в Університеті «КРОК». URL:[https://library.krok.edu.ua/media/library/category/publicna-informatsiya/3-3\\_osvitnij-protses-2020-11-02.pdf](https://library.krok.edu.ua/media/library/category/publicna-informatsiya/3-3_osvitnij-protses-2020-11-02.pdf)
2. Положення про академічну доброчесність в Університеті «КРОК». URL:[https://www.krok.edu.ua/download/nakazi/2018-10-18\\_kodeks-akademichnoi-dobrochesnosti.pdf](https://www.krok.edu.ua/download/nakazi/2018-10-18_kodeks-akademichnoi-dobrochesnosti.pdf)
3. Положення про перевірку академічних та наукових текстів на плагіат в Університеті «КРОК»  
URL:[https://library.krok.edu.ua/media/library/category/publicna-informatsiya/2020-04-10\\_polozhennya-pro-perevirku-tekstiv-na-plagiat.pdf](https://library.krok.edu.ua/media/library/category/publicna-informatsiya/2020-04-10_polozhennya-pro-perevirku-tekstiv-na-plagiat.pdf)
4. ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура та правила оформлювання.  
URL:[http://www.knmu.kharkov.ua/attachments/3659\\_3008-2015.pdf](http://www.knmu.kharkov.ua/attachments/3659_3008-2015.pdf)
5. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні вимоги та правила складання.  
URL:<http://pdf.lib.vntu.edu.ua/books/2018/%D0%94%D0%A1%D0%A2%D0%A3%208302%20%D0%BF%D0%BE%D0%B2%D0%BD%D0%B8%D0%B9.pdf>
6. Освітньо-професійна програма «Комп'ютерні науки» Університету «КРОК».  
URL:<https://library.krok.edu.ua/media/library/category/litsenzuvannia-ta-akredytatsiia/osvitni-prohramy/komp-yuterni-nauki-bakalavr-2021-04-29.pdf>
7. Стандарт вищої освіти першого (бакалаврського) рівня ступеня «бакалавр» за галуззю знань 12 «Інформаційні технології» спеціальністю 122 «Комп'ютерні науки».  
URL:<https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/2019/07/12/122-kompyuterni-nauki-bakalavr.pdf>
8. Положення про політику та процедури врегулювання конфліктних ситуацій в Університеті «КРОК».  
URL:[https://library.krok.edu.ua/media/library/category/publicna-informatsiya/2021-07-07\\_polozhennya-pro-protseduri-vregulyuvannya-konfliktnikh-situatsij.pdf](https://library.krok.edu.ua/media/library/category/publicna-informatsiya/2021-07-07_polozhennya-pro-protseduri-vregulyuvannya-konfliktnikh-situatsij.pdf)
9. Рекомендації МОН України щодо запобігання академічному плагіату та його виявлення в наукових роботах (авторефератах, дисертаціях, монографіях, наукових доповідях, статтях тощо).  
URL:<https://zakon.rada.gov.ua/rada/show/v8681729-18#Text>
10. Архітектура клієнт-сервер. URL: <https://medium.com/@IvanZmerzlyi>



**ДОДАТОК А**  
**Форма титульного аркушу пояснювальної записки Проєкту**

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»

ПРОЄКТ ЧЕТВЕРТОГО РІВНЯ

Тема: «.....»

Ступінь вищої освіти – бакалавр  
Спеціальність – 122 «Комп'ютерні науки»  
Освітня програма «Комп'ютерні науки»

ПОЯСНЮВАЛЬНА ЗАПИСКА

Виконав (-ла): здобувач (ка) 4 курсу  
групи \_\_\_\_\_

*Прізвище, ініціали*

Керівник: *науковий ступінь, посада, ПІБ керівника*