

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ БІЗНЕСУ, ІННОВАЦІЙНОГО
РОЗВИТКУ ТА МІЖНАРОДНОЇ ДІЯЛЬНОСТІ
Факультет менеджменту

Кафедра економічної кібернетики, комп'ютерних наук та інформаційних
технологій

ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ
методичні рекомендації
до проходження навчальної практики
здобувачами першого (бакалаврського) рівня вищої освіти
ОПП «Комп'ютерні науки»
спеціальності 122 «Комп'ютерні науки»
денної форми здобуття вищої освіти

МИКОЛАЇВ
2023

Друкується за рішенням науково–методичної комісії факультету менеджменту Миколаївського національного аграрного університету від 05.10.2023 р., протокол № 3

Укладачі:

- С. І. Тищенко – канд. пед. наук, доцент кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій Миколаївського національного аграрного університету;
- Т. С. Кучмійова – канд. екон. наук, доцент кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій Миколаївського національного аграрного університету
- В. В. Співак – асистент кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій Миколаївського національного аграрного університету

Рецензенти:

- Л. М. Макарова – канд. техн. наук, доцент кафедри програмного забезпечення автоматизованих систем Національного університету кораблебудування ім. адм. Макарова;
- Є.Ю. Борчик – канд. ф-м. наук, доцент кафедри вищої та прикладної математики Миколаївського національного аграрного університету.

© Миколаївський національний аграрний університет, 2023

ПЕРЕДМОВА

Методичні рекомендації до проходження навчальної практики розроблені для здобувачів вищої освіти першого (бакалаврського) рівня спеціальності 122 «Комп'ютерні науки» факультету менеджменту денної форми навчання.

Основна мета навчальної практики – підготувати майбутніх ІТ-фахівців до самостійної та командної розробки програмного забезпечення, сформуванню інженерне мислення та навички управління життєвим циклом ІТ-проектів.

Методичні рекомендації містять структурований матеріал та покрокові інструкції, необхідні для реалізації наскрізного комплексного ІТ-проекту (від етапу створення технічного завдання до верстки веб-інтерфейсу) на основі сучасного підходу Project-Based Learning (Проектне навчання).

Метою проходження навчальної практики є глибоке закріплення теоретичних знань з архітектури інформаційних систем та набуття здобувачами вищої освіти практичних умінь із системного аналізу, обробки великих масивів даних і проектування реляційних баз даних.

Головні завдання навчальної практики:

– опанування міжнародних стандартів розробки технічної документації (Специфікацій вимог до програмного забезпечення) та візуального моделювання бізнес-процесів;

– набуття навичок математичного моделювання, генерації тестових даних та побудови аналітичних дашбордів;

– засвоєння принципів проектування фізичної архітектури реляційних баз даних, нормалізації таблиць та практичного застосування мови SQL (Data Manipulation Language);

– розробка клієнтського веб-інтерфейсу (Front-end) прототипу інформаційної системи з використанням технологій HTML5 та CSS3;

– формування комплексного Проектного портфоліо замість традиційної текстової звітності.

ЕТАП 1. ІНІЦІАЦІЯ ПРОЄКТУ ТА ВИБІР МЕТОДОЛОГІЇ РОЗРОБКИ

Теоретичні відомості

Будь-який процес створення програмного забезпечення починається з ініціації та базового розуміння того, що таке комп'ютерна система.

Апаратне забезпечення (Hardware) — це фізичні складові комп'ютера (процесор, пам'ять, жорсткий диск). **Програмне забезпечення (Software)** — це набір інструкцій, які вказують апаратному забезпеченню, що і як робити. **Інформаційна система (ІС)** — це сукупність апаратних, програмних засобів, даних, процесів та користувачів, об'єднаних для збору, зберігання, обробки та видачі інформації з метою підтримки прийняття рішень у певній предметній області (медицина, логістика, освіта тощо).

Життєвий цикл програмного забезпечення (Software Development Life Cycle - SDLC): Це безперервний процес, що складається з чітких фаз: ініціація, збір вимог, проєктування, розробка (кодування), тестування, розгортання та підтримка.

Існують дві фундаментальні парадигми (методології) управління цим циклом:

1. **Каскадна модель (Waterfall Model):** Класичний, лінійний підхід. Кожен наступний етап (наприклад, написання коду) починається суворо після повного завершення попереднього (наприклад, затвердження дизайну). Перевага: чітка документація. Недолік: неможливість швидко адаптуватися до змін вимог замовника.

2. **Гнучкі методології (Agile):** Сучасний ітеративний підхід. Розробка ведеться короткими циклами (спринтами, зазвичай 2-4 тижні). Після кожного спринту замовник отримує робочу частину програми (інкремент). Це дозволяє швидко реагувати на зміни ринку.

Інструкція до виконання

1. Оберіть предметну область для вашого проєкту (наприклад: інтернет-магазин, поліклініка, автосалон).
2. Сформулюйте головну бізнес-мету майбутньої інформаційної системи.
3. Визначте, яку методологію розробки ви обираєте для проєкту та обґрунтуйте вибір.

Зразок виконання (Проєкт «LogisTech»)

- **Предметна область:** Логістика та вантажні перевезення.
- **Назва ІС:** Інформаційна система логістичної компанії «LogisTech».

- **Бізнес-мета:** Автоматизація обліку автопарку, розподілу водіїв та відстеження статусів доставки вантажів для зменшення логістичних витрат на 15%.
- **Методологія:** Обрано Agile (Scrum), оскільки бізнес-логіка перевезень може змінюватися, і компанії важливо якомога швидше отримати базовий працюючий прототип для диспетчерів.

ЕТАП 2. СИСТЕМНИЙ АНАЛІЗ ТА ІНЖЕНЕРІЯ ВИМОГ

Теоретичні відомості

Інженерія вимог (Requirements Engineering) — це процес виявлення, документування та підтримки вимог до програмного продукту.

Специфікація вимог (Software Requirements Specification - SRS) — це офіційний документ, який слугує "контрактом" між замовником та розробниками. Він детально описує поведінку системи.

Класифікація вимог:

1. **Функціональні вимоги:** Описують, що система *повинна робити*. Це конкретні функції. (Наприклад: "Система повинна дозволяти користувачу скинути пароль через email").
2. **Нефункціональні вимоги:** Описують, *як* система повинна працювати. Сюди входять продуктивність (час відгуку), безпека (шифрування), зручність використання (UI/UX).

Основи візуального моделювання: Складні текстові описи важко сприймати, тому інженери використовують графічні схеми. Найпопулярнішим стандартом є **UML (Unified Modeling Language)**.

- **Актор (Actor):** Будь-яка зовнішня сутність (людина, інша програма або обладнання), яка взаємодіє з нашою системою.
- **Блок-схема (Flowchart):** Графічне представлення алгоритму. Має суворі правила нотації: овал позначає початок/кінець алгоритму; прямокутник — виконання певної дії (обчислення); ромб — логічну перевірку умови (розгалуження "Так/Ні").

Основи роботи з текстовими процесорами (MS Word): Текстовий процесор відрізняється від простого текстового редактора (як Блокнот) тим, що дозволяє форматувати текст та вставляти мультимедіа.

- **Шрифт (Font):** Графічний малюнок накреслення літер. Бувають із зарубками (Serif, наприклад Times New Roman) та без зарубок (Sans-Serif, наприклад Arial). Для екранів краще читаються шрифти без зарубок.

- **Стили (Styles):** Іменованій набір параметрів форматування. Використання стилів "Заголовок 1", "Заголовок 2" є обов'язковим, оскільки саме на їх основі програма "розуміє" ієрархію документа і може автоматично згенерувати Зміст (Table of Contents).

Інструкція до виконання

1. Створіть файл MS Word. Налаштуйте ієрархію стилів ("Заголовок 1", "Заголовок 2") та згенеруйте Автоматичний зміст.
2. Пропишіть 5 функціональних вимог та визначте Акторів системи (ролі користувачів).
3. Розробіть **Словник даних** (таблиці, що описують, яка саме інформація буде зберігатися).
4. У сервісі Draw.io побудуйте блок-схему ключового бізнес-процесу. Вставте схему у Word, використовуючи інструмент "Вставити назву" (для підпису рисунка) та "Перехресне посилання".

Зразок виконання (Проект «LogisTech»)

- **Актори:** Диспетчер (керує автопарком), Водій (змінює статуси), Клієнт (відстежує вантаж).
- **Словник даних (Сутність "Транспорт"):** Поля: Vehicle_ID (Числовий), Plate_Number (Текст, макс 10 симв.), Capacity_Tons (Числовий), Status (Текст: Справний/Ремонт).
- **Схема алгоритму:** Процес «Призначення транспорту». Кроки: Диспетчер обирає Вантаж -> Ромб (Чи є вільне авто з відповідною вантажопідйомністю?) -> Так: Авто закріплюється за вантажем. -> Ні: Повідомлення "Немає вільних машин".

Схема алгоритму:
Процес «Призначення транспорту»

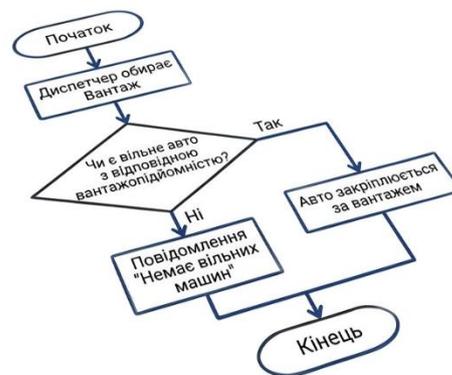


Рис 1.1.
Схема алгоритму: Процес «Призначення транспорту»

ЕТАП 3. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ВІЗУАЛІЗАЦІЯ ДАНИХ

Теоретичні відомості

Дані (Data) — це сирі, неорганізовані факти. Наприклад, цифри "120", "Київ", "2024". **Інформація (Information)** — це оброблені дані, які мають контекст і цінність. Наприклад: "У 2024 році до Києва було доставлено 120 вантажів".

Основи табличних процесорів (MS Excel): Електронна таблиця складається зі стовпців (позначаються латинськими літерами A, B, C) та рядків (позначаються цифрами 1, 2, 3). Перетин стовпця та рядка утворює **Клітинку (Cell)**, яка має унікальну адресу (наприклад, B5).

- **Відносні посилання:** При копіюванні формули адреса клітинки автоматично змінюється (наприклад, A1 стає A2).
- **Абсолютні посилання:** Використовується знак долара \$A\$1. При копіюванні формули адреса "заморожується" і залишається незмінною. Це критично важливо при посиланні на фіксовані курси валют чи тарифи.

Business Intelligence (BI) та агрегація даних: BI — це технології та інструменти для перетворення сирих даних на значущу інформацію для бізнес-аналізу. В Excel основою BI є **Зведені таблиці (Pivot Tables)**. Вони реалізують технологію OLAP (багатовимірний аналіз), дозволяючи миттєво групувати тисячі рядків даних (наприклад, підсумувати всі прибутки по місяцях), не пишучи жодної формули математичного сумування вручну.

Дашборди (Dashboards): Це візуальна панель управління. Її мета — показати ключові метрики (KPI) на одному екрані.

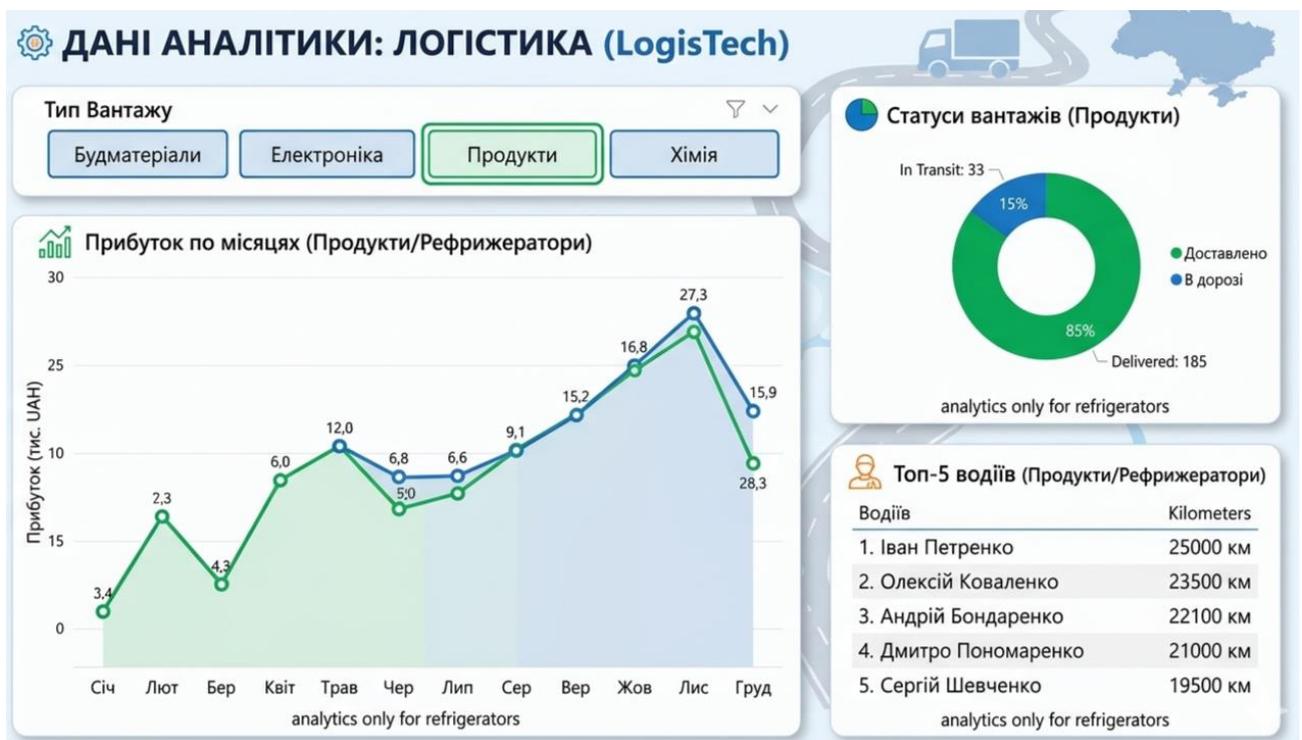
Правила: кругова діаграма використовується тільки для показу часток цілого (наприклад, % у загальному обсязі); лінійний графік — для показу змін у часі (динаміка по роках/місяцях); гістограма — для порівняння категорій.

Інструкція до виконання

1. В MS Excel згенеруйте масив тестових даних (Mock Data) — таблицю на 100-150 рядків.
2. Створіть розрахункові стовпці: використайте функцію VLOOKUP (або XLOOKUP) для підтягування даних з інших аркушів (імітація зв'язків), та логічні функції (IF, AND) для розрахунків за умовами.
3. Побудуйте 3 зведені таблиці.
4. На окремому аркуші побудуйте інтерактивний Дашборд зі Зведеними діаграмами та підключіть до них Зрізи (Slicers) для миттєвої фільтрації даних.

Зразок виконання (Проект «LogisTech»)

- **Генерація:** Таблиця на 200 маршрутів (ID_Маршруту, Дата, Вантаж, Вага, Відстань, Статус).
- **Формули:** Розрахунок вартості реалізовано формулою $\text{=Відстань} * 35$ (тариф за 1 км). Визначення типу транспорту логікою: $\text{=IF(Вага} \leq 2; \text{"Мікроавтобус"; IF(Вага} \leq 10; \text{"Вантажівка"; "Фура"})$.
- **Дашборд:** Побудовано графік "Прибуток по місяцях", кругову діаграму "Статуси вантажів" (В дорозі/Доставлено) та топ-5 водіїв. Додано Зріз "Тип Вантажу". При кліку на "Продукти" всі графіки показують аналітику лише для рефрижераторів.



ЕТАП 4. ПРОЄКТУВАННЯ РЕЛЯЦІЙНИХ БАЗ ДАНИХ ТА SQL (BACK-END LAYER)

Теоретичні відомості

База даних (БД) — це структурований масив інформації. **СУБД (Система управління базами даних)** — це програма (наприклад, MS Access, MySQL, Oracle), яка дозволяє створювати БД, записувати туди дані та швидко їх шукати.

Реляційна модель та Ключі: Найпопулярніша сьогодні модель — реляційна (від англ. *relation* — відношення/таблиця). Вся інформація зберігається у двовимірних таблицях.

- **Типи даних:** Кожен стовпець має чіткий тип (Integer — цілі числа, String — текст, Boolean — логічний тип Правда/Брехня, Date/Time).

- **Первинний ключ (Primary Key - PK):** Унікальний ідентифікатор рядка (наприклад, номер паспорта людини або унікальний артикул товару). Двох однакових PK у таблиці бути не може.

- **Зовнішній ключ (Foreign Key - FK):** Стовпець, який посилається на Первинний ключ іншої таблиці. Він фізично створює зв'язок між таблицями і забезпечує *посилальну цілісність*.

Нормалізація таблиць: Це процес оптимізації БД для усунення дублювання.

- *1-ша Нормальна форма (1NF):* У кожній клітинці має бути лише одне значення (не можна в клітинку "Телефон" записати два номери через кому).

- *2-га та 3-тя Нормальні форми:* Вимагають, щоб усі дані залежали тільки від Первинного ключа. Наприклад, у таблиці "Замовлення" не можна зберігати адресу клієнта текстом. Треба зберігати лише його ID_Клієнта, а саму адресу винести в окрему таблицю "Клієнти".

Мова запитів SQL (Structured Query Language): Це міжнародний стандарт спілкування з базами даних. Базові команди (так звані CRUD - Create, Read, Update, Delete):

- SELECT — читання даних.
- INSERT INTO — додавання нових записів.
- UPDATE — оновлення існуючих.
- DELETE — видалення рядків. Команда JOIN дозволяє "склеїти" під час запиту дані з кількох пов'язаних таблиць.

Інструкція до виконання

1. Відкрийте MS Access. У режимі "Конструктор" створіть 3-4 таблиці згідно з вашим Словником даних. Задайте типи даних та Первинні ключі.

2. У вікні "Схема даних" налаштуйте зв'язки (Один-до-багатьох) та поставте прапорець "**Забезпечення цілісності даних**".

3. Заповніть таблиці тестовими даними (по 10 записів).

4. Перейдіть у режим SQL-view та напишіть 4 запити: SELECT (з використанням INNER JOIN), INSERT, UPDATE, DELETE. Збережіть їх.

разок виконання (Проект «LogisTech»)

- **Таблиці:** Drivers (Водії), Vehicles (Транспорт), Cargoes (Вантажі), Routes (Маршрути - зв'язуюча таблиця, що містить зовнішні ключі на інші три).
- **Схема:** Налаштовано зв'язки 1:М. Один водій може здійснити багато маршрутів, але один маршрут виконується одним водієм.

- **Приклад SQL (Вибірка вантажів у дорозі):**

SQL

```
SELECT Cargoes.Client, Drivers.Name, Vehicles.Plate
```

```
FROM (Routes INNER JOIN Drivers ON Routes.Driver_ID = Drivers.ID)
```

```
INNER JOIN Vehicles ON Routes.Vehicle_ID = Vehicles.ID
```

```
INNER JOIN Cargoes ON Routes.Cargo_ID = Cargoes.ID
```

```
WHERE Cargoes.Status = 'В дорозі';
```

ЕТАП 5. РОЗРОБКА КЛІЄНТСЬКОГО ВЕБ-ІНТЕРФЕЙСУ (FRONT-END LAYER)

Теоретичні відомості

Інтернет — це глобальна мережа з'єднаних між собою комп'ютерів. Взаємодія в ній базується на **Клієнт-серверній архітектурі**. Сервер (потужний комп'ютер десь у дата-центрі) зберігає базу даних та логіку. Клієнт (ваш браузер, наприклад Chrome) надсилає запити на сервер і відображає отриманий результат.

Розробка зовнішнього вигляду для клієнта називається Front-end. Вона базується на трьох "китах": HTML (структура), CSS (дизайн) та JavaScript (інтерактивність). У межах практики ми фокусуємось на HTML та CSS.

HTML5 (HyperText Markup Language): Це не мова програмування, а мова розмітки. Вона складається з **Тегів** (елементів у кутових дужках < >), які вказують браузеру, що саме знаходиться на сторінці (зображення, абзац, таблиця).

Семантична верстка: Сучасний стандарт HTML5 вимагає використання тегів, що мають смислове значення: <header> для шапки, <footer> для підвалу, <nav> для меню. Це критично для SEO (щоб Google розумів сторінку) та доступності для людей з інвалідністю.

CSS3 (Cascading Style Sheets): Каскадні таблиці стилів відповідають за "одяг" веб-сторінки: кольори, шрифти, позиціонування.

Box Model (Блокова модель): У CSS кожен елемент — це невидима коробка. Вона складається з контенту, внутрішнього відступу (padding), рамки (border) та зовнішнього відступу (margin).

Flexbox: Сучасна технологія CSS для створення гнучких сіток. Дозволяє легко вибудувати елементи в один рядок, відцентрувати їх або рівномірно розподілити вільний простір між ними (що раніше було дуже складною задачею).

Інструкція до виконання

1. Створіть файл *index.html*. Напишіть базову семантичну структуру (<header>, <main>, <footer>).
2. У блоці <main> зверстайте таблицю (<table>), яка імітує виведення даних з вашої БД, та форму (<form> з полями <input>) для додавання нового запису.
3. Створіть файл *style.css* і підключіть його до HTML.
4. Застосуйте CSS-стили: налаштуйте шрифти, використайте Flexbox для навігаційного меню, додайте ефекти наведення миші (:hover) для кнопок та тіні (box-shadow) для форми.

Зразок виконання (Проект «LogisTech»)

- **HTML:** Створено шапку <header> з логотипом. Основна частина містить семантичний тег <section>, всередині якого лежить таблиця активних маршрутів. Нижче розміщено форму додавання вантажу з полями: Замовник (type="text"), Вага (type="number") та кнопка <button>.
- **CSS:** Використано display: flex; justify-content: space-between; для того, щоб логотип був зліва, а меню — справа. Таблиці задано рамки border: 1px solid #ccc; та зелений колір тексту для статусу "Доставлено".

Файл *index.html* (Структура сторінки)

Цей файл формує «скелет» нашого веб-інтерфейсу, використовуючи сучасні семантичні теги HTML5.

HTML

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>LogisTech - Панель диспетчера</title>
  <link rel="stylesheet" href="style.css">
</head>
```

```

<body>

  <header class="header">
    <div class="logo">LogisTech</div>
    <nav class="nav-menu">
      <a href="#">Головна</a>
      <a href="#">Маршрути</a>
      <a href="#">Водії</a>
    </nav>
  </header>

  <main>
    <section class="routes-section">
      <h2>Активні маршрути</h2>
      <table class="routes-table">
        <thead>
          <tr>
            <th>ID</th>
            <th>Замовник</th>
            <th>Вантаж (т)</th>
            <th>Статус</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>101</td>
            <td>ТОВ "БудПостач"</td>
            <td>5</td>
            <td>В дорозі</td>
          </tr>
          <tr>
            <td>102</td>
            <td>АТ "Сільпо"</td>
            <td>2</td>
            <td class="status-delivered">Доставлено</td>
          </tr>
        </tbody>
      </table>
    </section>

    <section class="form-section">
      <h2>Додати новий вантаж</h2>
      <form class="add-cargo-form">
        <label for="customerName">Замовник:</label>

```

```

        <input type="text" id="customerName" name="customerName"
placeholder="Назва компанії" required>

        <label for="cargoWeight">Вага (тонни):</label>
        <input type="number" id="cargoWeight" name="cargoWeight" min="1"
placeholder="0" required>

        <button type="submit">Створити маршрут</button>
    </form>
</section>
</main>

</body>
</html>

```

Файл style.css (Оформлення сторінки)

Цей файл відповідає за дизайн. Тут реалізовано Flexbox для шапки, рамки для таблиці та зелений колір статусу.

CSS

```

/* Базові налаштування шрифту та фону */
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f8f9fa;
    color: #333;
}

/* ===== */
/* ШАПКА САЙТУ (HEADER) */
/* ===== */
.header {
    background-color: #004085;
    color: white;
    padding: 15px 30px;
    /* Використовуємо Flexbox для розміщення лого зліва, а меню справа */
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.logo {
    font-size: 24px;
    font-weight: bold;
    letter-spacing: 1px;
}

.nav-menu a {

```

```

    color: white;
    text-decoration: none;
    margin-left: 20px;
}

.nav-menu a:hover {
    text-decoration: underline;
}

/* ===== */
/* ОСНОВНИЙ КОНТЕНТ (MAIN) */
/* ===== */
main {
    padding: 30px;
}

/* ===== */
/* ТАБЛИЦЯ (TABLE) */
/* ===== */
.routes-table {
    width: 100%;
    border-collapse: collapse; /* Прибирає подвійні лінії між комірками */
    background-color: white;
    margin-bottom: 40px;
}

.routes-table th,
.routes-table td {
    border: 1px solid #ccc; /* Ті самі рамки з нашого завдання */
    padding: 12px;
    text-align: left;
}

.routes-table th {
    background-color: #e9ecef;
}

/* Клас для статусу "Доставлено" */
.status-delivered {
    color: #28a745; /* Зелений колір */
    font-weight: bold;
}

/* ===== */
/* ФОРМА (FORM) */
/* ===== */
.add-cargo-form {
    background-color: white;
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    width: 300px;
}

```

```
.add-cargo-form label {
  display: block;
  margin-bottom: 5px;
  font-weight: bold;
}

.add-cargo-form input {
  width: 100%;
  padding: 8px;
  margin-bottom: 15px;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box; /* Щоб відступи не ламали ширину */
}

.add-cargo-form button {
  width: 100%;
  padding: 10px;
  background-color: #007bff;
  color: white;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  font-weight: bold;
}

.add-cargo-form button:hover {
  background-color: #0056b3;
}
```

ЕТАП 6. ЗАБЕЗПЕЧЕННЯ ЯКОСТІ (QA) ТА ЗАХИСТ ПРОЄКТНОГО ПОРТФОЛІО

Теоретичні відомості

Забезпечення якості (Quality Assurance, QA) — це процес перевірки того, що розроблене програмне забезпечення відповідає початковим вимогам (написаним у SRS на Етапі 1) та не містить помилок (багів).

Види тестування:

1. **White-box testing (Тестування "білої скриньки"):** Тестувальник бачить і аналізує вихідний код програми.
2. **Black-box testing (Тестування "чорної скриньки"):** Тестувальник не знає, як написаний код. Він просто вводить дані в інтерфейс (як звичайний користувач) і дивиться на результат.

Архівація даних: Для передачі готового продукту замовнику (або викладачу) файли збираються в архів (.zip або .rar).

Архівування — це процес стиснення даних за допомогою математичних алгоритмів (наприклад, пошуку повторюваних послідовностей бітів), що дозволяє зменшити розмір файлів і об'єднати їх у єдиний пакет (Release Candidate).

Інструкція до виконання

1. Проведіть самотестування (Black-box) свого проекту: перевірте, чи всі посилання працюють, чи відкривається HTML-файл у різних браузерах, чи коректно відпрацьовують SQL-запити.

2. Сформууйте фінальний дистрибутив: створіть ZIP-архів. До нього покладіть файл документації (Word), аналітики (Excel), бази даних (Access) та папку з веб-інтерфейсом (HTML/CSS).

3. Завантажте архів у систему дистанційного навчання та підготуйтеся до публічного захисту (демонстрації екрана з поясненням логіки роботи розробленої системи).

Зразок виконання (Проект «LogisTech»)

- Здобувач формує архів *LogisTech_Portfolio_Ivanov.zip*.
- Під час захисту здобувач відкриває дашборд Excel, демонструючи, як клік на зрізі змінює фінансову аналітику; потім відкриває Access, запускає SQL-запит на оновлення статусу вантажу і доводить, що дані в таблиці змінилися; врешті демонструє зверстаний веб-інтерфейс диспетчера.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Антоненко В. М., Мамченко С. Д., Рогушина Ю. В. Сучасні інформаційні системи і технології: управління знаннями : навчальний посібник. Ірпінь : Національний університет ДПС України, 2023. 212 с.
2. Грінова Л. В., Сибірякова І. Г. Інформаційні системи та технології. Частина 1. Технічне та програмне забезпечення інформаційних технологій та систем : навчальний посібник. Харків : Monograf, 2023. 121 с.
3. Литвин В. В., Гошко С. В., Висоцька В. А. Бази даних та знань. Проектування інформаційних систем : підручник. Львів : Видавництво Львівської політехніки, 2021. 272 с.
4. Комп'ютери та комп'ютерні технології : метод. реком. до виконання практичних робіт в табличному процесорі MS Excel для здобувачів вищої освіти освітнього ступеня «Молодший бакалавр» початкового рівня (короткий цикл спеціальності 208 «Агроінженерія» денної форми навчання / уклад. : Ю. В. Волосяк, Л. О. Борян. Миколаїв : МНАУ, 2020. 86 с. URL: <https://dspace.mnau.edu.ua/jspui/handle/123456789/10942>
5. Грицюк Ю. І. Проектування та розробка інформаційних систем : навчальний посібник. Львів : Вид-во ЛДУ БЖД, 2022. 302 с.
6. Савченко С. О., Шевченко І. В. Сучасні технології розробки веб-орієнтованих систем (Front-end та Back-end) : конспект лекцій. Київ : КПІ ім. Ігоря Сікорського, 2022. 185 с.
7. Бондаренко Л. В., Мороз Т. О. Інформаційні технології в системному аналізі : методичні рекомендації до практичних робіт. Миколаїв : МНАУ, 2021. 75 с.
8. Мельник А. О., Клименко О. В. Архітектура та організація баз даних: реляційна модель : навчальний посібник. Тернопіль : ТНТУ імені Івана Пулюя, 2023. 210 с.

ЗМІСТ

ПЕРЕДМОВА	3
ЕТАП 1. ІНІЦІАЦІЯ ПРОЄКТУ ТА ВИБІР МЕТОДОЛОГІЇ РОЗРОБКИ.....	4
ЕТАП 2. СИСТЕМНИЙ АНАЛІЗ ТА ІНЖЕНЕРІЯ ВИМОГ	5
ЕТАП 3. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ВІЗУАЛІЗАЦІЯ ДАНИХ	7
ЕТАП 4. ПРОЄКТУВАННЯ РЕЛЯЦІЙНИХ БАЗ ДАНИХ ТА SQL (BACK-END LAYER).....	8
ЕТАП 5. РОЗРОБКА КЛІЄНТСЬКОГО ВЕБ-ІНТЕРФЕЙСУ (FRONT-END LAYER).....	10
ЕТАП 6. ЗАБЕЗПЕЧЕННЯ ЯКОСТІ (QA) ТА ЗАХИСТ ПРОЄКТНОГО ПОРТФОЛІО.....	15
СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ	17

Навчальне видання

ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Методичні рекомендації

Укладачі: **Тищенко** Світлана Іванівна
Кучмійова Тетяна Сергіївна
Співак Вадим Вікторович

Формат 60x84 1/16. Ум. друк. арк. 1,2

Тираж 10 прим. Зам. № _

Надруковано у видавничому відділі
Миколаївського національного аграрного університету
54020, м. Миколаїв, вул. Георгія Гонгадзе, 9

Свідоцтво суб'єкта видавничої справи ДК № 4490 від 20.02.2013 р.