ЗАТВЕРДЖУЮ
Начальник кафедри інформаційних технологій та систем електронних комунікацій, к.т.н., доцент
Назарій БУРАК
«____» _____ 20___ р.

МЕТОДИЧНА РОЗРОБКА

для проведення лабораторного заняття із здобувачами освіти першого (бакалаврського) рівня вищої освіти спеціальності F3 «Комп'ютерні науки»
з дисципліни «WEB-технології та WEB-програмування»

Лабораторна робота № 1: «Налаштування середовища розробки (JDK, IntelliJ IDEA, Git). Дослідження HTTP-взаємодії засобами розробника браузера. Створення статичної вебсторінки»
(тема 1.1)

Мета: підготувати робоче середовище Java-розробника, опанувати базові операції Git і GitHub, навчитися аналізувати HTTP-обмін засобами розробника браузера та утилітою curl, створити семантичну статичну вебсторінку.

Час: 4 години.

Місце заняття: комп'ютерна лабораторія.

Матеріальне забезпечення: ПК з доступом до мережі Internet; JDK 21; IntelliJ IDEA (Community або Ultimate за освітньою ліцензією) або Visual Studio Code; Git; браузер Chrome/Firefox; обліковий запис GitHub; демо-проєкт https://github.com/SidiromUA/shelter-registry (тег `lab-01`).

Після лабораторного заняття здобувачі освіти повинні:

вміти: встановлювати й перевіряти JDK, створювати репозиторій, фіксувати зміни і позначати їх тегами в Git, публікувати проєкт на GitHub; знаходити у DevTools метод, код стану, заголовки та cookies запиту; надсилати HTTP-запити утилітою curl; створювати вебсторінку із семантичною розміткою;

знати: структуру HTTP-запиту і відповіді, класи кодів стану, призначення основних заголовків, базові команди Git.

I. Підготовка до заняття

Викладач перевіряє наявність на ПК лабораторії JDK 21, IntelliJ IDEA, Git і доступ до github.com. Здобувачі освіти до заняття реєструють обліковий запис на GitHub і опрацьовують матеріал лекції 1.

II. Структура заняття

Короткі теоретичні відомості

Вебзастосунок працює за архітектурою «клієнт-сервер»: браузер надсилає HTTP-запит (метод, URL, заголовки, тіло), сервер повертає відповідь (код стану, заголовки, тіло). Засоби розробника браузера (DevTools, клавіша F12) дають змогу переглянути кожен запит сторінки. **Git** – розподілена система контролю версій: *комітт* фіксує стан файлів, *тег* позначає важливу версію (у курсі – стан на момент здачі лабораторної роботи), *віддалений репозиторій* на GitHub зберігає копію проєкту і використовується для здачі робіт.

2.1. Встановлення і перевірка середовища

1. Встановіть JDK 21 (Eclipse Temurin, Amazon Corretto або Oracle OpenJDK) та IntelliJ IDEA. Перевірте в терміналі:

```
java -version          # очікується: openjdk version "21..."
javac -version
git --version
```

2. Налаштуйте Git (ім’я та пошта мають збігатися з обліковим записом GitHub):

```
git config --global user.name  "Ім’я Прізвище"
git config --global user.email "you@example.com"
git config --global init.defaultBranch main
```

2.2. Ознайомлення з демо-проєктом

```
git clone https://github.com/SidiromUA/shelter-registry.git
cd shelter-registry
git tag                # список етапів: lab-01 … lab-08
git checkout lab-01    # стан проєкту для ЛР 1
cd web
jwebserver -p 8000     # простий вебсервер з JDK 18+
```

Відкрийте http://localhost:8000 – сторінку «Реєстр укриттів міста». Перегляньте файл `web/index.html`: зверніть увагу на семантичні елементи `<header>`, `<nav>`, `<main>`, `<section>`, `<footer>`, атрибут `lang="uk"`, таблицю з `<caption>` і `<th scope="col">`.

2.3. Дослідження HTTP засобами розробника браузера

1. Відкрийте DevTools (F12) → вкладка **Network**, увімкніть *Disable cache*, оновіть сторінку (F5).
2. Для запиту до `/` і до `css/style.css` запишіть: метод, URL, код стану, заголовки відповіді `Content-Type`, `Content-Length`, `Last-Modified`; заголовки запиту `User-Agent`, `Accept`, `Accept-Language`.
3. Вимкніть *Disable cache* і оновіть сторінку ще раз. Знайдіть відповіді з кодом **304 Not Modified** і поясніть, чому сервер не передав тіло.
4. Відкрийте неіснуючу адресу http://localhost:8000/missing.html – отримайте код **404**.
5. Відкрийте сайт https://ldubgd.edu.ua, знайдіть перенаправлення (3xx) і заголовок `Location`, cookies (вкладка **Application** → Cookies) та їхні атрибути `HttpOnly`, `Secure`, `SameSite`.

2.4. Запити утилітою curl

```
# лише заголовки (метод HEAD)
curl -I http://localhost:8000/
# повний обмін: запит і відповідь
curl -v http://localhost:8000/css/style.css
# неіснуючий ресурс: 404
curl -i http://localhost:8000/missing.html
# умовний запит: 304 Not Modified
curl -I -H "If-Modified-Since: Wed, 01 Jan 2031 00:00:00 GMT"
http://localhost:8000/
# лише код стану і час відповіді
curl -s -o /dev/null -w "%{http_code} %{time_total}s\n"
https://developer.mozilla.org
```

Порівняйте рядки, що починаються з `>` (запит) і `<` (відповідь) у виводі `curl -v`, з даними DevTools.

2.5. Індивідуальне завдання

Оберіть предметну область власного вебсервісу (вона стане основою наступних лабораторних робіт і курсової роботи; можна запропонувати власну тему або обрати з таблиці 1, погодивши з викладачем).

№	Тема	№	Тема
1	Облік пожежно-рятувальної техніки	11	Запис до сервісного центру
2	Реєстр пожежних гідрантів	12	Бібліотека та видача книг
3	Журнал навчань з цивільного захисту	13	Облік комп’ютерної техніки кафедри
4	Облік засобів індивідуального захисту	14	Бронювання аудиторій

5	Реєстр волонтерів і заявок на допомогу	15	Розклад занять і викладачі
6	Пункти незламності та їхнє обладнання	16	Спортивний клуб: абонементи
7	Облік вогнегасників установи	17	Каталог курсів і записи студентів
8	Черговість чергувань підрозділу	18	Ветеринарна клініка: пацієнти і прийоми
9	Склад гуманітарної допомоги	19	Прокат обладнання
10	Інциденти кібербезпеки організації	20	Подання та розгляд звернень громадян

Таблиця 1 – Орієнтовні теми власного вебсервісу

1. Створіть локальний репозиторій і порожній публічний репозиторій на GitHub (назва англійською, напр. `hydrant-registry`).
2. Створіть сторінку `web/index.html` для своєї предметної області: заголовок, навігація, короткий опис сервісу, таблиця щонайменше з 5 записами (з `<caption>` і `<th scope>`), список, підвал; файл стилів `web/css/style.css`; файл `README.md` з описом сервісу і майбутніх сутностей (3–4 сутності з полями).
3. Запустіть сторінку через `jwebserver`, проаналізуйте її завантаження в DevTools.
4. Зафіксуйте зміни, створіть тег і надішліть на GitHub:

```
git init
git add .
git commit -m "ЛР1: статична сторінка сервісу"
git remote add origin https://github.com/<логін>/<репозиторій>.git
git push -u origin main
git tag lab-01
git push origin lab-01
```

2.6. Вимоги до звіту

Звіт (PDF або DOC, до 5 сторінок) містить: титульну сторінку; мету роботи; знімки екрана DevTools (Network – запит з кодом 200, 304 і 404; Application – cookies обраного сайту) з поясненням заголовків; вивід команд `curl -v` і `curl -I` з коротким аналізом; знімок екрана власної сторінки; посилання на репозиторій GitHub з тегом `lab-01`; відповіді на 3 контрольні запитання (за вибором); висновки. У ВУ в завданні «ЛР 1» у поле «Текстова відповідь» вставте посилання на репозиторій, файлом додайте звіт.

2.7. Контрольні запитання

1. З яких частин складаються HTTP-запит і HTTP-відповідь?

2. Чим відрізняються коди 200, 304 і 404? Коли сервер повертає 304?
3. Для чого призначені заголовки Content-Type, Cache-Control, Location?
4. Які атрибути cookie захищають від XSS і CSRF?
5. Чим коміт відрізняється від тегу в Git? Навіщо потрібен `git push origin lab-01`?
6. Навіщо використовувати семантичні елементи HTML замість `<div>`?

2.8. Критерії оцінювання (максимально 5 балів)

Бали	Критерій
5	усі пункти виконано самостійно і вчасно; репозиторій з тегом `lab-01`, сторінка семантична; аналіз HTTP повний; під час захисту здобувач вільно пояснює роботу
4	роботу виконано повністю, є незначні недоліки в сторінці, звіті чи відповідях
3	виконано не менше 60 % роботи або є суттєві помилки; здобувач частково пояснює виконане
0–2	виконано менше 60 % або здобувач не може пояснити поданий результат

Робота, подана після встановленого терміну без поважних причин, оцінюється з пониженням на 1 бал.

III. Підведення підсумків заняття

Викладач перевіряє наявність репозиторіїв і тегів, відповідає на запитання, нагадує про необхідність погодити тему власного вебсервісу до лабораторної роботи № 2.

Методичну розробку розробив:

Доцент кафедри ІТтаСЕК, к.т.н. Роман ГОЛОВАТИЙ

Методична розробка обговорена на засіданні кафедри ІТтаСЕК
Протокол № ____ від «____» _____ 20___ р.