

Кафедра інформаційних технологій та систем електронних комунікацій
Спеціальність 122 «Комп'ютерні науки», ступінь вищої освіти «бакалавр»,
VI семестр

Дисципліна «Web-технології та web-дизайн»

ЗМІСТОВИЙ МОДУЛЬ 1. ВЕБ-РОЗРОБКА ТА БАЗИ ДАНИХ В JAVA ЛЕКЦІЯ №1.1. ВВЕДЕННЯ В ВЕБ-ТЕХНОЛОГІЇ ТА ВЕБ-ПРОГРАМУВАННЯ

План:

Вступ до курсу

1. Основні поняття та техніко-економічні передумови розвитку веб-технологій
2. Архітектура та принцип роботи веб-додатків (модель «клієнт–сервер»)
3. Класифікація веб-технологій
4. Роль веб-технологій у сучасному світі та програмні засоби розробки

Література:

1. Java 8. The Complete Reference. 9 edition / Herbert Schildt. – New York : Oracle Press, 2015. – 1274 с.
2. Spring 5 для професіоналів / Кларенс Хо, Кріс Шефер, Роб Харроп, Юліана Козміна. – Наука і техніка, 2020. – 1120 с.
3. Stallings W. Data and Computer Communications. 10th ed. – Pearson, 2013. – 912 p.

Вступ до курсу

Освітня програма підготовки бакалавра зі спеціальності «Комп'ютерні науки» передбачає набуття здобувачами освіти низки фахових компетентностей в області вебтехнологій та веб-програмування. Ці компетентності формуються через систематичне вивчення низки курсів, зокрема «Основи програмування», «Основи веб-розробки», «Об'єктно-орієнтоване програмування», «Клієнт-серверне програмування» тощо. Курс «Web-технології та web-дизайн» є логічним продовженням зазначених дисциплін та слугує містком між базовими навичками програмування і реальною розробкою сучасних веб-додатків.

Основною метою курсу є надання здобувачам освіти глибокого розуміння вебтехнологій і навичок веб-програмування, необхідних для створення та підтримки сучасних вебдодатків та веб-сайтів. Курс орієнтовано на вивчення технологій розробки клієнтської та серверної частин веб-додатків, поглиблену роботу з мовою програмування Java, роботу з фреймворками (насамперед Spring Boot), а також основи роботи з базами даних як складовими веб-орієнтованих систем.

Навчальний матеріал дисципліни структуровано за модульним принципом і складається з двох змістових модулів: «Веб-розробка та бази даних в Java» та «Spring Boot». Засвоєння теоретичного матеріалу передбачає 16 годин лекційних занять, освоєння практичної частини зосереджено в межах 48 годин лабораторних занять. Загальний обсяг дисципліни становить 5,5 кредита ЄКТС (165 академічних годин), з яких 101 година відведена на самостійну роботу. Підсумковим етапом вивчення курсу є написання курсової роботи, ключовим завданням якої є створення та побудова веб-сервісу, а семестровий контроль проводиться у формі екзамену.

Здобувачам освіти надано можливість здобуття поглиблених знань шляхом проходження відкритих он-лайн курсів «Learn Spring», «Data Structures & Backend with Java» та «Spring

Framework», розміщених на платформах Codecademy та Coursera. Здобувачі, які успішно опанували рекомендований он-лайн курс, що засвідчено відповідним сертифікатом, звільняються від виконання обов'язкових індивідуальних завдань (відповідно до пройденого курсу).

1. Основні поняття та техніко-економічні передумови розвитку веб-технологій

Веб-технології (Web-технології) – це сукупність методів, протоколів, мов та програмних засобів, що забезпечують створення, розміщення, передавання та відображення інформаційних ресурсів у мережі Інтернет (Всесвітній павутині). Інакше кажучи, це технології, які роблять можливим функціонування веб-сайтів і веб-додатків.

Всесвітня павутина (World Wide Web, WWW) – це розподілена інформаційна система, що надає доступ до взаємопов'язаних документів та ресурсів, розміщених на серверах по всьому світу. Її було запропоновано британським ученим Тімом Бернерсом-Лі у 1989 році в Європейській організації з ядерних досліджень (CERN). Саме тоді сформувалися три наріжні технології, що й донині лежать в основі вебу: **HTML** (мова розмітки гіпертексту), **URL** (уніфікований локатор ресурсу — адреса) та **HTTP** (протокол передавання гіпертексту).

Веб-сайт – це сукупність логічно пов'язаних веб-сторінок, об'єднаних спільною темою, дизайном та доменним іменем. **Веб-додаток (веб-застосунок)** – це програма, що виконується на веб-сервері, а взаємодія з користувачем відбувається через браузер за допомогою мережевих протоколів. На відміну від статичного сайту, веб-додаток формує вміст динамічно — залежно від дій користувача та даних, що зберігаються на сервері.

Розвиток вебу прийнято умовно поділяти на кілька етапів (поколінь):

- ◆ **Web 1.0 (≈1991 – початок 2000-х).** Епоха статичних сайтів. Сторінки склалися переважно з HTML, були орієнтовані лише на читання (read-only), а користувач виступав пасивним споживачем інформації. Взаємодія була мінімальною.
- ◆ **Web 2.0 (≈2004 – 2010-ті).** Епоха інтерактивності та користувацького контенту. З'явилися соціальні мережі, блоги, вікі-сервіси. Користувач став активним учасником, який сам створює та редагує вміст (read-write). Поширилися технології AJAX, динамічні сторінки та насичені веб-інтерфейси.
- ◆ **Web 3.0 (з 2010-х рр.).** Етап «семантичного» та децентралізованого вебу. Акцент робиться на інтелектуальній обробці даних, машинному навчанні, персоналізації, а також на технологіях розподілених реєстрів. Дані стають машиночитними та пов'язаними між собою.

Серед основних **техніко-економічних передумов** бурхливого розвитку веб-технологій виокремлюють: стрімке здешевлення та поширення доступу до Інтернету; зростання обчислювальної потужності серверів і клієнтських пристроїв; стандартизацію протоколів та мов розмітки; появу хмарних сервісів, що знизили вартість розгортання й обслуговування додатків; а також глобалізацію бізнесу, який потребує цілодобової присутності в мережі та можливості обслуговувати клієнтів незалежно від їхнього місцезнаходження.

2. Архітектура та принцип роботи веб-додатків

В основу переважної більшості сучасних веб-додатків покладено **архітектуру «клієнт-сервер»**. Це модель взаємодії, у якій обчислювальні задачі розподілено між двома типами

вузлів: *клієнтом*, який надсилає запити, та *сервером*, який ці запити обробляє і повертає результат (відповідь).

Основні складові архітектури веб-додатка:

- ◆ **Клієнтська частина (front-end)** – виконується у браузері користувача. Відповідає за відображення інтерфейсу та взаємодію з користувачем. Базові технології: HTML (структура), CSS (оформлення), JavaScript (поведінка та інтерактивність).
- ◆ **Серверна частина (back-end)** – виконується на віддаленому сервері. Відповідає за бізнес-логіку, обробку запитів, автентифікацію, роботу з базою даних. У межах цього курсу серверну логіку реалізують мовою Java з використанням контейнерів сервлетів (Tomcat) та фреймворку Spring Boot.
- ◆ **База даних** – забезпечує довготривале зберігання даних додатка (наприклад, СКБД MySQL). Доступ до неї здійснюється з серверної частини, як правило, мовою запитів SQL.
- ◆ **Веб-сервер** – програмний компонент, що приймає HTTP-запити від клієнтів і передає їх відповідним обробникам, а також повертає сформовані відповіді.

Взаємодію між клієнтом і сервером забезпечує **протокол HTTP (HyperText Transfer Protocol)** або його захищена версія **HTTPS**. Протокол є *без станів* (stateless): кожен запит обробляється незалежно від попередніх. Для збереження стану сеансу (наприклад, факту авторизації користувача) застосовують допоміжні механізми — *cookies*, *сесії* та токени.

Принцип роботи веб-додатка можна описати такою послідовністю кроків:

- Користувач у браузері вводить адресу (URL) або виконує дію (натискає кнопку, надсилає форму). Браузер формує **HTTP-запит** і надсилає його на сервер.
- Система доменних імен (**DNS**) перетворює символічне ім'я домену на IP-адресу сервера, після чого встановлюється з'єднання.
- Веб-сервер приймає запит і передає його відповідному обробнику (сервлету, контролеру). Серверний код виконує бізнес-логіку: за потреби звертається до бази даних.
- На основі отриманих даних сервер формує **HTTP-відповідь** (зазвичай HTML-сторінку або дані у форматі JSON) і повертає її браузеру.
- Браузер отримує відповідь, інтерпретує HTML, застосовує CSS і виконує JavaScript, відображаючи готову сторінку користувачеві. За потреби цикл повторюється.

Наведений опис є узагальненим — у реальних системах застосовують кешування, балансування навантаження, проміжні сервіси тощо. Проте розуміння цього базового циклу є необхідним для формування цілісної уяви про процеси, які відбуваються під час роботи будь-якого веб-додатка.

3. Класифікація веб-технологій

Веб-технології зручно класифікувати залежно від того, на якому боці системи «клієнт–сервер» вони виконуються, а також за призначенням. Розрізняють **клієнтські (front-end)** та **серверні (back-end)** технології.

До **клієнтських технологій** належать:

- **HTML (HyperText Markup Language)** – мова розмітки, що задає структуру та зміст веб-сторінки (заголовки, абзаци, посилання, зображення, форми).
- **CSS (Cascading Style Sheets)** – каскадні таблиці стилів, що відповідають за візуальне оформлення: кольори, шрифти, розташування елементів, адаптивність.
- **JavaScript** – мова програмування, що виконується в браузері й надає сторінці інтерактивність (обробка подій, динамічна зміна вмісту, асинхронні запити AJAX).

До серверних технологій належать:

- **Мови та платформи серверної розробки** – Java (Servlet, Spring), а також PHP, Python, Node.js, C# тощо. У цьому курсі основна увага приділяється Java та Spring Boot.
- **Веб-сервери та контейнери** – Apache Tomcat, Nginx, які приймають та опрацьовують HTTP-запити.
- **Системи управління базами даних (СКБД)** – MySQL, PostgreSQL та інші реляційні СКБД, доступ до яких здійснюється мовою SQL.
- **Засоби обміну даними** – REST API, формати JSON та XML для взаємодії між клієнтом і сервером або між окремими сервісами.

За способом формування вмісту сайти поділяють на **статичні** (вміст незмінний і повертається у вигляді готових файлів) та **динамічні** (вміст формується сервером «на льоту» залежно від запиту та стану бази даних). Окремо виокремлюють **односторінкові додатки (SPA, Single Page Application)**, у яких сторінка завантажується один раз, а подальша взаємодія відбувається без повного перезавантаження.

4. Роль веб-технологій у сучасному світі та програмні засоби розробки

Сьогодні веб-технології пронизують практично всі сфери життя суспільства. На їх основі функціонують системи електронної комерції, інтернет-банкінг, державні електронні сервіси, освітні платформи (зокрема середовище «Віртуальний університет»), соціальні мережі, корпоративні інформаційні системи та хмарні сервіси. Веб став основним каналом взаємодії бізнесу зі споживачем і важливим інструментом цифрової трансформації економіки.

Переваги веб-додатків порівняно з класичними настільними програмами: **кросплатформність** (працюють у будь-якому браузері незалежно від операційної системи), **відсутність потреби в інсталяції** на пристрій користувача, **централізоване оновлення** (зміни одразу доступні всім користувачам) та **доступність** з будь-якої точки, де є мережа.

Для розробки веб-додатків у межах курсу використовують такі програмні засоби:

- ◆ **Інтегровані середовища розробки (IDE)** – IntelliJ IDEA, Visual Studio Code, Sublime Text, що надають зручний інтерфейс для написання, налагодження та запуску коду.
- ◆ **Комплект розробника Java (JDK)** – набір інструментів для компіляції та виконання програм мовою Java.
- ◆ **Фреймворк Spring Boot** – засіб для швидкого створення серверної частини веб-додатків на Java з мінімальними налаштуваннями.
- ◆ **Веб-сервер (контейнер сервлетів) Apache Tomcat** – середовище виконання Java-веб-застосунків.
- ◆ **Система управління базами даних MySQL** – для зберігання даних веб-додатка.

Як приклад, найпростіша веб-сторінка мовою HTML, що повертається клієнту, має такий вигляд:

```
<!DOCTYPE html>
<html lang="uk">
  <head>
    <meta charset="UTF-8">
    <title>Моя перша сторінка</title>
  </head>
  <body>
    <h1>Вітаємо у світі веб-технологій!</h1>
  </body>
</html>
```

Висновок

Висновок: основним завданням розглянутої теми є донесення загальних понять про веб-технології та веб-програмування, що необхідно для формування у майбутніх фахівців цілісної уяви про принципи побудови та функціонування сучасних веб-додатків. Було розглянуто базову термінологію, історію та передумови розвитку вебу, архітектуру «клієнт–сервер», класифікацію веб-технологій та основні програмні засоби розробки. Поглиблене вивчення серверної частини, баз даних та фреймворку Spring Boot буде реалізовано в наступних темах курсу, а отримані знання слугуватимуть фундаментом для виконання лабораторних робіт та курсової роботи.

Завдання на самопідготовку:

- Створити власний акаунт (за відсутності) та зареєструватися на курс «Web-технології та web-дизайн» у віртуальному середовищі «Віртуальний університет» (адреса: <https://virt.ldubgd.edu.ua/course/view.php?id=2111>).
- Опрацювати конспект лекції (доступний у віртуальному середовищі) та підготуватися до складання тестового завдання за матеріалами теми 1.1.
- Самостійно ознайомитися з призначенням технологій HTML, CSS та JavaScript і навести приклад їх застосування на одному обраному веб-сайті.

Для поглибленого вивчення:

1. Head First Java / Kathy Sierra, Bert Bates. – «O'REILLY», 2012. – 678 с.
2. Codecademy. Курс «Learn Spring». [Електронний ресурс]. – Доступний з <https://www.codecademy.com/learn/learn-spring>
3. Coursera. Курс «Data Structures & Backend with Java». [Електронний ресурс]. – Доступний з <https://www.coursera.org/learn/data-structures--backend-with-java>