# 2.1. Служба Веб

**WWW (World Wide Web), Веб, Всесвітня Павутина** - це розподілена інформаційна система, оскільки інформація зберігається на величезній кількості веб-серверів.

Служба Веб – світова інформаційна бібліотека, яка забезпечує засоби розміщення інформації і доступ до неї за допомогою кабелів і комп'ютерів (Інтернету). Служба Веб є доступною в основному через Інтернет, тому користувачі часто ототожнюють поняття Веб і Інтернет.

Веб можна віднести до внутрішнього наповнення, тобто, це віртуальний світ знань, тоді як Інтернет є фізичною стороною глобальної мережі у вигляді величезної кількості обладнання.

## Клієнт-серверна архітектура та її компоненти

Вузли в Інтернеті можуть містити різні пристрої, які виконують функції передачі, зберігання та обробки даних.

Основні типи пристроїв у вузлах інтернету

1. Комп'ютери та сервери

* Персональні комп'ютери використовуються кінцевими користувачами для доступу до Інтернету та виконання різних завдань.
* Сервери забезпечують різні послуги, такі як веб-хостинг, поштові послуги, бази даних тощо. Сервери можуть бути фізичними чи віртуальними.

2. Мережне обладнання

* Маршрутизатори (Routers) керують передачею даних між різними мережами та забезпечують маршрутизацію пакетів даних.
* Комутатори (Switches) з'єднують пристрої в локальній мережі (LAN) і керують передачею даних між ними.
* Концентратори (Hubs) це прості пристрої, які з'єднують кілька комп'ютерів у мережі, але не керують передачею даних так ефективно, як комутатори.
* Модеми перетворюють цифрові сигнали для передачі по лініях.

Самою поширеною в інтернеті є клієнт-серверній архітектура, де використовуються три компоненти (рис.1).



Рис.1. Клієнт-серверна архітектура

* **Клієнт** – апаратний або програмний компонент обчислювальної системи, який надсилає запити до серверу за відповідним протоколом. Клієнт може запитувати з сервера будь-які дані, маніпулювати даними безпосередньо на сервері, запускати на сервері нові процеси. Для різних інтернет служб використовуються різні клієнти. Для служби Веб це зазвичай браузер.
* **Сервер** - це комп'ютер зі спеціальним програмним забезпеченням, що має власну ІР-адресу. Основним завданням сервера є безперебійна робота та можливість обробляти мільйони запитів від користувачів. Оскільки веб-ресурси знаходяться на сервері, тисячі людей можуть звертатися до одного сервера і отримувати від нього потрібну інформацію.
* **База даних** — невід'ємна частина сучасного веб-проекту, де зберігається важлива інформація для роботи. Діапазон використання бази даних є широким - від зберігання налаштувань до зберігання опублікованих матеріалів. База даних надає просте та швидке управління даними: вибірка, додавання, видалення та зміна даних у базі. З використанням бази даних легко вирішуються такі завдання як пошук по сайту, реєстрація та авторизація користувачів, генерація динамічних сторінок тощо.

Взаємодія «клієнт-сервер» для служби Веб відбувається за прикладним протоколом HTTP (Hyper Text Transfer Protocol), але згодом його доопрацьовали для нових завдань і він став використовуватися повсюдно.

У початковому вигляді HTTP є незахищеним. Дані між користувачами передаються у відкритому вигляді, зловмисник може втрутитися в передачу даних, перехопити їх або змінити. Щоб захистити дані користувачів, створено протокол HTTPS - HyperText Transfer Protocol Secure (захищений протокол HTTP). HTTPS працює завдяки SSL/TLS-сертифікату.

SSL/TLS-сертифікат (Secure Sockets Layer/Transport Layer Security) - це цифровий підпис сайту, що підтверджує його справжність. Перед тим як встановити захищене з'єднання, браузер запитує цей документ і звертається до центру сертифікації, щоб підтвердити легальність документа. Якщо він дійсний, то браузер вважає цей сайт безпечним і починає обмін даними.

## Загальні відомості про веб-сайт

Основними об’єктами Веб є веб-проекти — це сукупності файлів, що об'єднані під однією доменною адресою і знаходяться на одному сервері за IP-адресою. В більшості випадків в Інтернеті один сайт/додаток має одну доменну адресу, за якою він ідентифікується. Можливими є інші варіанти: один сайт на кількох доменах або кілька сайтів під одним доменом.

Один сайт може бути доступним за різними адресами і зберігатися на різних серверах. Наприклад, *google.com.ua* і *google.pl* логічно є сайтом Google з різними мовами інтерфейсу, але технічно це різні сайти. Часто, великі сайти (веб-портали) використовують кілька доменів, щоб логічно відокремити різні види послуг (mail.google.com, news.google.com, maps.google.com).

Перший сайт з'явився в 1991 році завдяки Тіму Бернерсу-Лі, який представив нову технологію World Wide Web, засновану на протоколі HTTP. Цей сайт існує досі за адресою <http://info.cern.ch/> (рис.2).



Рис. 2. Перший веб-сайт

Цей історичний сайт є набором гіпертексту з посиланнями на основні поняття, напрями розвитку ресурсу тощо. У порівнянні з сучасними мережними проектами він виглядає дуже просто, але має історичне значення.

Розвиток веб-проектів безпосередньо пов'язаний з популяризацією Інтернету. Історія створення перших мов програмування для веб, таких як Perl та PHP, цікава та пов'язана з розвитком інтернету та потребами створення динамічних веб-сайтів.

Мову програмування Perl (Practical Extraction and Reporting Language) розроблено Ларрі Уоллом наприкінці 1980-х і на початку 1990-х років. Perl був створений для обробки текстової інформації та автоматизації різних завдань обробки даних. З розвитком інтернету стало зрозуміло, що Perl можна використовувати для створення динамічних веб-сайтів. Він надав потужні засоби для роботи з формами, базами даних та обробки HTTP-запитів.

PHP (PHP: Hypertext Preprocessor) розроблено Расмусом Лердорфом в 1994 році для управління персональними веб-сторінками та обробки даних форм.

PHP був створений як інтерпретована мова, що дозволило вставляти код прямо в HTML-сторінки. PHP надав потужні засоби для обробки даних, надісланих через веб-форми, та взаємодії з базами даних. Це зробило РНР ідеальним для створення динамічних веб-сайтів.

PHP має популярність завдяки своїй простоті та зручності використання. Він є безкоштовним, підтримується великою спільнотою розробників. Багато популярних веб-сайтів та веб-додатків, включаючи WordPress, Facebook та Wikipedia, використовують PHP у своїй роботі.

Довгий час веб-сторінки залишалися простими, оскільки не було інструментарію та потреб у їх ускладненні. Лише на початку 2000-х років веб-програмування стало набирати популярності і до сьогоднішнього моменту веб-додатки представлені величезною різноманітністю.

До сьогоднішнього дня історично склалося 3 типи додатків (рис.3).



Рис.3. Типи додатків

* **Десктопні додатки** передбачають встановлення клієнта на стороні користувача. Залежно від типу операційної системи, процесора, відеокарти та інших параметрів можуть знадобитися різні версії додатку. Це створює певні незручності як розробникам (їм потрібно постійно вишукувати баги в різних середовищах, розширювати обсяг коду для обліку всіх можливих комбінацій заліза клієнта), так і користувачам (необхідне завантаження постійних оновлень, нове залізо з операційною системою, яку підтримує додаток).
* **Мобільні додатки** розробляються виключно для смартфонів та планшетів з врахуванням встановленої там операційної системи (Android, iOS тощо). Це також додає навантаження для розробників софту. Багато мобільних додатків фактично є веб-додатками, оскільки можливості «движків» браузерів це дозволяють.
* **Веб-додаток** – програма з певним набором функціоналу, якому для здійснення бізнес-логіки потрібне мережне з'єднання та наявність на стороні користувача браузера.

Для більшості користувачів веб-додатки зовні не відрізняються від сайтів. Але сайти мають більш інформативний характер, а веб-додатки мають велику кількість вбудованих функцій – від роботи з графікою та таблицями до оформлення замовлень онлайн.

* **Сайт.** Не складні в технічному виконанні веб-проекти, що не вимагають реєстрації для перегляду контенту. Сторінки сайту не змінюються від дій користувача і надають зазвичай сталу інформацію. При переході між розділами, сторінки перезавантажуються. На сайті можна прочитати статтю, програти відео, переглянути фотогалерею чи анімацію. Користувач може лише скористатися пошуком, опублікувати відгук, підписатися на новини, залишити заявку, надіслати повідомлення до власника сайту. До сайтів можна віднести персональні сторінки, сайти-візитки, рекламні лендинги та інші не складні проекти.
* **Веб-додаток.** Інтерактивна комп'ютерна програма, що надає різні можливості для авторизованих користувачів та гостей. Повний функціонал доступний лише для зареєстрованих та авторизованих користувачів. У додатку користувач може виконувати широкий діапазон дій: вибирати та купувати товари, здійснювати комунальні платежі, бронювати житло чи квитки, прив’язувати власні події до календаря, публікувати власні пости та коментувати пости інших користувачів, спілкуватися в чатах та відеоконференцях, здійснювати пошук на основі своїх інтересів (рис.4).



Рис. 4. Типовий вигляд веб-додатку

Інтерактивні веб-додатки складаються з двох частин:

* Клієнтська частина завантажується у браузер, і з нею безпосередньо спілкується користувач.
* Серверна частина розміщується на сервері, містить всі сценарії та дані, що закладено у веб-додатку. Сервер отримує запит від користувача, виконує відповідні сценарії, витягує з бази даних відповідну інформацію, генерує HTML-сторінку та надсилає її до браузера.

#### Прикладами веб-додатків є:

* Інтернет-пошта (Gmail).
* Хмарні сховища (Dropbox).
* Текстові редактори (Google Документи) та онлайн-нотатки (Evernote).
* Соціальні мережі (Facebook, Tik-Tok, Instagram).
* Магазини електронної комерції (Amazon, Rozetka, Foxtrot, Comfy).
* Системи замовлень та продажів (Booking.com,Tickets.ua, Proizd.ua).
* Системи управління проектами (Trello).
* Онлайн-банкінги (Privat24, Monobank).
* Системи онлайн навчання та контролю знань (Coursera.org, Віртуальне навчальне середовище Львівської політехніки).

#### Загальні відмінності між сайтом і додатком

|  |  |  |
| --- | --- | --- |
|  | Інтерактивний веб-додаток | Веб сайт |
| Мета створення | Взаємодія з кінцевим споживачем | Донести до споживача інформацію |
| Доступ | Має обмеження. Для використання повного функціонал потрібна аутентифікація | Доступ мають всі відвідувачі |
| Взаємодія з користувачем | Користувач може маніпулювати даними як ознайомитися з інформацією, так й зробити певні маніпуляції (наприклад, працювати з текстом) | Користувач може лише ознайомитися з контентом, залишити відгук |
| Структура | Веб-додаток має більш складну структуру порівняно з сайтом |
| Компіляція | Потрібна. Крім того, якщо до веб-додатку буде внесено зміни, компіляцію потрібно буде зробити ще раз | Не потрібна, навіть якщо до сайту вносяться зміни. |
| Формування сторінок | Фрагменти HTML-сторінки генеруються та оновлюються в масштабі реального часу | Видає заздалегідь підготовлені HTML-сторінки в основному зі статичними файлами |

#### Переваги веб-додатків:

* Можуть працювати як на смартфоні, так і на персональному комп'ютері. Практично незалежні від заліза.
* Веб-додаток розробляється з врахуванням того, щоб його можна було відкрити з будь-якого браузера та операційної системи.
* Веб-додатки не вимагають встановлення на пристрій. Оновлення веб-додатку відбувається автоматично та централізовано. Всі користувачі отримують доступ до однієї версії веб-додатку, що дозволяє уникнути непорозумінь.
* Веб-додатки прості у використанні, не вимагають спеціальних знань та вмінь, що дозволяє працювати з різною аудиторією.
* Веб-додаток можна легко підлаштувати під потреби будь-якого бізнесу. Веб-додатки зменшують витрати як для бізнесу, так і для кінцевого користувача, оскільки вимагають менше ресурсів (людських та обладнання) для роботи і обслуговування.
* Веб-додатки рентабельні, оскільки дешевші за класичні програмні рішення, що дозволяє їх використовувати компаніям з обмеженим бюджетом. За функціоналом майже не поступаються десктопним аналогам.

## Веб-сервер

Сервер — це мережний комп'ютер з власною ІР-адресою, який обробляє запити від клієнтів або обмінюється даними з іншими серверами. У сервера багато завдань: він зберігає великий обсяг даних, запускає віртуальні машини, виконує обчислювальні операції. Власник і адміністратор сервера можуть гнучко міняти необхідні налаштування, дозволяти або забороняти доступ до його ресурсів, під’єднувати, налаштовувати і запускати ряд додаткових програм і функцій, тобто цілком конфігурувати його роботу за потребою.

Поняття «веб-сервер» може стосуватися як апаратного так і програмного забезпечення. Або навіть до обох частин, які працюють спільно.

* З точки зору апаратного забезпечення, веб-сервер - це потужний комп'ютер, на якому зберігаються файли сайту і він надсилає їх до пристрою кінцевого користувача.
* З точки зору програмного забезпечення, веб-серверна програма містить кілька компонентів, які контролюють доступ веб-користувачів до розміщених на сервері файлів.

### Веб-сервер (апаратне забезпечення)

Вимоги до серверу:

* Цілодобове функціонування, під’єднання до Інтернету через швидкісні магістралі.
* Має постійну IP адресу.
* Обслуговується спеціальною компанією.

### Операційні системи для серверів

Сервер буває фізичним чи віртуальним, і більшість серверних ОС розробляються для стабільної та швидкої роботи як на окремому фізичному сервері, так і у віртуальному середовищі.

Операційні системи для клієнтських і серверних комп'ютерів різняться у кількох ключових аспектах, оскільки вони призначені для різних цілей і виконують різні функції. Основні відмінності серверних операційних систем від стандартних:

* **Клієнтські ОС** орієнтовані на використання кінцевими користувачами для спільних завдань, таких як робота з офісними програмами, інтернет-серфінг, мультимедійні програми та ігри. Мають зручний графічний інтерфейс, який дозволяє користувачам взаємодіяти із системою за допомогою миші та клавіатури. Зазвичай, орієнтація на роботу одного користувача.
* **Серверні ОС** призначені для серверних комп'ютерів, які забезпечують обслуговування і доступ до даних та ресурсів. Надають можливість віддаленого адміністрування через командний рядок або веб-інтерфейс, при цьому графічний інтерфейс може бути вимкнений для підвищення безпеки та продуктивності. Надають сучасні функції управління ресурсами, такі як кластеризація, балансування навантаження, віртуалізація та управління сховищами даних.

На рис.5 представлено рейтинг серверних операційних систем з найбільшою кількістю користувачів по всьому світу (<https://uk.hostadvice.com/marketshare/os/>)



###### Рис.5. Рейтинг 2024 року серверних операційних систем

Серверні ОС поділяються на три сімейства: Windows, Unix та Linux.

* **Windows Server**. Поширеною операційною системою є ОС Windows, хоча її витісняють системи, що засновані на UNIX ядрі. Windows Server може виконувати функції багатьох програм-серверів: поштової та файлової служб, DNS, потоків мультимедіа тощо. Система має зручну графічну оболонку, високу стабільність роботи і відмінну технічну підтримку. Оновлення і доповнення виходять щомісяця.
* **Unix** – основа більшості нинішніх серверних ОС. Спочатку це були закриті та дорогі операційні системи для компаній - наприклад, macOS від Apple. Згодом на базі Unix розроблено безкоштовні популярні рішення, наприклад FreeBSD та Gentoo.
* **Linux** – сімейство операційних систем, що виросли на основі Unix. ОС з врахуванням Linux називаються дистрибутивами. Найвідоміші – Red Hat Enterprise Linux, CentOS, Ubuntu, Debian.

### Веб-сервер (програмне забезпечення)

На сервері встановлено спеціальну програму (веб-сервер або HTTP-cepвер), яка працює за протоколом HTTP або HTTPS. Вона приймає запити від клієнтської машини та розподіляє завдання між іншими серверними програмами, що їх обробляють. Серверні програми виконують завдання та надсилають результати до веб-серверної програми. Веб-серверна програма об’єднує результати, перетворює їх в HTML-код і надсилає до клієнта сформовану сторінку з відповіддю.

 Потужні веб-серверні програми можуть виконувати багато інших дій над об’єктами.

* Запуск сценаріїв, що написані серверними мовами програмування.
* Генерація веб-сторінок з даних, що витягуються з бази даних.
* Захист інформації від спотворення та знищення.
* Ідентифікація та облік активності користувачів.
* Обробка запитів інших видів (mailto, FTP тощо).

Веб-серверні програми, не мають інтерфейсу користувача і «спілкуються» лише з комп’ютером-сервером, приймають від нього надіслані користувачем дані і повертають до нього результат. Цим вони докорінним чином різняться від клієнтських програм, що працюють безпосередньо з користувачем.

На серверах використовують різні типи веб-серверних програм відповідно до рейтингу (<https://uk.hostadvice.com/marketshare/server/>) (рис.6)



Рис.6. Рейтинг 2024 року веб-серверних програм

### Веб-сервери для платформ, що сумісні із стандартом UNIX

**Apache і Nginx -** це дві популярні веб-серверні програми з відкритим вихідним кодом, разом їх частка становить понад 50%. Обидві програми пропонують всі необхідні можливості, що здатні нормально витримувати великі робочі навантаження і інтегруватися з іншими програмами, щоб забезпечити повноцінне функціонування.

Веб-сервери Apache і Nginx мають дуже багато схожих якостей, але їх не можна розглядати як взаємозамінні. Кожна програма має свої особливості, і тому найкраще підходить для певних ситуацій.

### Комерційні веб-сервери

* Microsoft IIS (Internet Information Server) розроблено Microsoft для серверів на платформі Windows.
* IBM WebSphere Application Server розроблено IBM, забезпечує користувача гнучким середовищем для створення та розгортання додатків, у тому числі хмарних.
* Oracle WebLogic Server розроблено Oracle, призначений для розгортання Java-додатків.

Кожна веб-серверна програма має свої особливості, характеристики та переваги, а вибір залежить від конкретних потреб та вимог веб-проекту.

## Браузери

Браузер - це вікно у світ Інтернету, без якого не може обійтися жоден користувач. Браузери поширюються безкоштовно або у комплекті з операційною системою чи прикладною програмою.

Спочатку браузери призначалися лише для перегляду гіпертекстових документів. Проте, на даний час браузери стають універсальними клієнтами, які можна використати для інших служб Інтернету. За допомогою браузерів можна пересилати файли з віддалених комп’ютерів на локальний і навпаки, доступатися до віддаленого комп’ютера і працювати на ньому як на власному, приймати участь в відео конференціях, прослуховувати радіопередачі, спілкуватися за ІР-телефонією тощо. Відображення медіа-контенту залежить від того, чи підтримує браузер конкретні типи файлів, або від наявності встановлених плагинів.

Браузери надсилають запити до сервера, обробляють отриману відповідь своїми механізмами і відображають результат у своєму вікні. Стандарти відображення розробляються і затверджуються Консорціумом W3C (<https://www.w3.org/>). У надісланій відповіді, окрім самої сторінки може міститися багато іншої інформації в різних форматах, наприклад, у вигляді файлів, електронних повідомлень, сторінок тощо.

Для безпомилкового отримання інформації в потрібному форматі використовується універсальний покажчик ресурсу URL (Universal Resource Locator), який однозначно ідентифікує будь-який ресурс в мережі Інтернет. Саме такий рядок відображається в адресному полі браузера.

<https://www.site.lviv.ua/documents/page.html> або <https://213.82.46.1/documents/page.html>

Універсальний покажчик ресурсу відображає:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| https:// | www. | site.lviv.ua/ | documents/ | page.html |
| протокол | служба | доменна адреса | шлях | файл |

* **Протокол відповідної служби.** В даному прикладі використано протокол https:// – протокол передачі гіпертексту.
* **Назва служби.** В даному прикладі це служба Веб - www
* **Доменна або IP-адреса**, яка однозначно ідентифікує веб-сервер в мережі Інтернет, на якому розміщено потрібний веб-ресурс.
* **Шлях**, що складається з назв директорій, розділених символом «/» (слеш), послідовно відкриваючи які, можна «дістатися» до потрібної інформації. У даному прикладі інформація знаходиться в директорії «documents».
* **Ім'я файлу**, який містить потрібну інформацію. В цьому прикладі інформація знаходиться у файлі page.html.

Якщо браузер використовують для служби Веб, то в адресному полі можна не вказувати назву протоколу і назву служби. Хоча іноді, залежно від налаштувань, веб-сервер скеровує запити *www.site.lviv.ua* та *site.lviv.ua* в різні директорії.

### Інтерфейс браузера

Інтерфейси сучасних браузерів є результатом багаторічної еволюції, хоча специфікації, яка визначає стандарти користувацького інтерфейсу браузера, поки не існує. Втім, зазвичай, присутні основні елементи: адресний рядок, рядок стану і панель інструментів. В різних браузерах існують і специфічні функції, наприклад, менеджер завантажень в Firefox.

Користувацькі інтерфейси різних браузерів мають багато спільного. Основними елементами інтерфейсу браузера є:

* Адресний рядок для введення доменної адреси чи URL.
* Кнопки навігації "Назад" і "Вперед".
* Закладки.
* Кнопки поновлення і зупинки завантаження сторінки.
* Кнопка "Додому" для переходу до головної сторінки.

### Основні компоненти браузера

Механізм браузера

Користувацький інтерфейс

Модуль рендерингу (відображення)

Мережний механізм

Java Script інтерпретатор

Внутрішній UI

Сховище
даних (кеш)

Ї

Ї

Ї

Ї

Ї

Ї

Ї

Рис. 7. Основні компоненти браузера

1. **Інтерфейс користувача** - містить адресний рядок, кнопки "Назад" і "Вперед", меню закладок і подібне. До нього відноситься будь-яка частина вікна браузера за винятком області, в якій відображається сторінка.
2. **Механізм (рушій) браузера** - керує взаємодією інтерфейсу користувача та модуля відображення.
3. **Модуль відображення** (**рендерингу**) - відповідає за виведення отриманої інформації на екран. Наприклад, якщо це веб-сторінка, то даний компонент браузера здійснює синтаксичний розбір HTML і CSS коду, а отриманий в результаті цього аналізу контент відображається на екрані пристрою.
4. **Мережний механізм** - призначений для виконання мережних операцій, таких як HTTP-запити. У ньому передбачено багатоплатформовий інтерфейс і низькорівневі реалізації для кожної з підтримуваних платформ.
5. **Внутрішній UI** (Виконавча частина користувацького інтерфейсу) - застосовується для відтворення графічних елементів і форм вікна браузера. Він надає типовий інтерфейс, сумісний із різними платформами, а на низькому рівні використовує методи, передбачені операційною системою.
6. **Інтерпретатор JavaScript** - використовується для синтаксичного аналізу та виконання коду JavaScript.
7. **Сховище даних (кеш)** для збереження даних різних типів.

### Модуль відображення (рендерингу)

Модуль відображення відповідає за виведення отриманої від сервера відповіді у вікні браузера у зручному для користувача вигляді. Сучасні браузери спроможні відображати файли в різних форматах.

Значна частина відмінностей браузерів випливає з відмінностей у модулі відображення сторінок. Ці відмінності приводять до того, що сторінки в різних браузерах можуть виглядати дещо по-різному.

5 основних модулів відображення для рендерингу сторінок

|  |  |
| --- | --- |
| Microsoft_Edge_logo | **EdgeHTML** - браузерний модуль від Microsoft, який втілено в Microsoft Edge.  |
|  | **Gecko** - движок з відкритим вихідним кодом, який розроблено ще за часів браузера Netscape, але почав використовуватися тільки з браузера Mozilla Firefox. |
|  | **WebKit** - движок з відкритим кодом, він розроблявся такими компаніями як Google, Apple, Nokia та іншими. Спочатку було розроблено для платформи Linux і адаптовано компанією Apple для Mac OS і Windows. Використовується в браузері Safari від Apple, і в багатьох браузерах мобільних пристроїв. |
|  | **Blink** - заснований на WebKit, що розробляється Google. Використовується браузером Google Chrome, а також браузерами Opera, і браузерами для мобільних пристроїв на Android. В цих браузерах використовується кілька екземплярів модуля відображення, по одному в кожній вкладці, які представляють собою окремі процеси. |

Модулі відображення постійно вдосконалюються і узгоджуються їх стандарти, але розробнику слід тестувати сторінки в кількох браузерах, щоб виконати умову кросбраузерності.

## Взаємодія браузера і сервера

Браузер надсилає до сервера запит, наприклад, отримати певну сторінку веб-сайту і відобразити на екрані користувача. Веб-сервер приймає запит, шукає запитану сторінку і надсилає до клієнта її вміст або повідомлення про помилку, якщо такої сторінки не знайдено чи доступ до неї є забороненим. Веб-сервер не аналізує вміст сторінки, яку передає. Всю роботу зі структуризації і відображення отриманої інформації виконує браузер (рис.8).



Рис.8. Етапи взаємодії користувача з функціоналом веб-програми

Такий зв'язок між клієнтом і сервером може встановлюватися багаторазово в міру зміни запитів, даних на сайті, оновлення серверної частини тощо. При цьому може здійснюватися перезавантаження сторінок або дозавантаження даних. У деяких випадках дані кешуються для прискорення обробки повторних запитів.

Браузер клієнта може лише отримувати і відображати інформацію з сервера. Розміщувати чи змінювати інформацію за допомогою браузера можна у випадку, коли завантаження файлів на сервер реалізовано за допомогою спеціальних скриптів, що втілено в адміністративну частину сайту.

В решті випадків доведеться користуватися так званим ftp-доступом. Розробник за допомогою ftp-клієнтів може вивантажувати необхідні файли у відведену для сайту директорію. В обох випадках потрібно буде знати реєстраційні дані і пароль для доступу до системи.

## Статичний та динамічний контент

Весь контент в інтернеті можна поділити на два типи: статичний та динамічний.

### Статичний контент

Статичний контент - це незмінні файли, які зберігаються на сервері і доставляються всім користувачам в однаковому вигляді. Це дуже простий тип інформації. Не важливо, що за користувач прийшов на сайт, заходить він туди вперше або повторно, - всі користувачі побачать один і той же вміст (рис.9).



Рис.9. Функціонування сайту зі статичним контентом

Класичний приклад статики - це картинки, відео, текстові файли, HTML-файли та інші подібні дані.

Приклади використання статичного контенту:

* Статті у блозі.
* Сторінки з описом послуг, які пропонує компанія.
* Сторінка з переліком документів.

### Особливості статичного контенту



Рис.10. Особливості сайтів зі статичним контентом

Головна особливість статики – її просто доставляти та кешувати (рис.10).

* Коли користувач мандрує сайтом, браузер надсилає запити до серверу.
* Сервер знаходить потрібний файл і передає його.
* Інформація відображається у користувача та паралельно зберігається в кеші.
* Коли наступного разу користувач зайде на цей же сайт, браузер не надсилатиме запит на сервер, а завантажить дані з кеша. Інформація з'явиться на екрані швидше.

Інші особливості:

* **Проста послідовність дій.** У відповідь на запит серверу потрібно просто знайти файл на диску і надіслати його.
* **Швидке завантаження.** Статика легко і швидко доставляється, її можна без проблем кешувати. Значить, і часу на завантаження потрібно небагато.
* **Зміни лише вручну.** Якщо власник веб-сервісу захоче змінити статичний контент, потрібно буде видаляти файли з сервера вручну і завантажувати нові.

Пошук потрібної сторінки здійснюється в певній директорії, яку відведено на сервері для сайту. Зазвичай, назва директорії збігається з назвою доменної адреси сайту. У разі звернення до сайту в цілому (наприклад, *site.ua*), веб-сервер автоматично надсилає «стартову сторінку» з назвою *index.*\* (*index.html*) або в деяких випадках — default.\* (*default.html*) (рис.11).



Рис.11. Робота веб-сервера зі статичним контентом

Цей документ повинен розташовуватися в кореневій директорії, яку відведено для розміщення сайту. Решта файлів можна розміщувати або в цій же директорії, або у вкладених директоріях, що іноді буває зручним, особливо у разі, коли сайт містить кілька тематичних розділів або рубрик.

* Деякі веб-сервери на відміну від комп'ютерів користувачів є чутливими до регістру літер, в якому представлено імена директорій і файлів, тому назви «documents», «DOCUMENTS» або «Documents» будуть позначати різні директорії. Варто назви всіх об’єктів сайту писати рядковими (маленькими) літерами.
* Щоб уникнути помилок, які пов'язані з відмінностями в обробці кодувань кириличної мови різними серверами, назви варто зазначати латиницею.
* Заборонено вживати в назвах ресурсів пробілів, спеціальних символів (\ / : \* ? " < > | +) та кирилиці.

### Динамічний контент

На відміну від статичного контенту, динамічний - це інформація, що постійно змінюється. Дані, які відображаються у користувача на екрані, змінюватимуться залежно від його поведінки. При цьому у кожного користувача контент на екрані буде свій, не такий, як у іншої людини.

Динамічний контент не зберігається в готовому вигляді, він формується в момент запиту та залежить від багатьох параметрів (рис.12).



Рис.12. Функціонування сайту з динамічним контентом

Класичні приклади динамічного контенту – це веб-сторінки, що генеруються кодом на PHP, Python або JS. Текст листування в чатах – це також динаміка. Користувачі обмінюються повідомленнями, що генеруються в реальному часі.

Приклади використання динамічного контенту:

* Підбірка контенту для стрічки в соціальній мережі. Стрічки новин формуються динамічно і для різних користувачів будуть різнитися.
* Рекламні банери на веб-сайтах. Зазвичай, кожен користувач бачить на сайтах або в соціальних мережах індивідуальні рекламні пропозиції, які ґрунтуються на його поведінці в інтернеті.
* Добірки товарів, рекомендації в інтернет-магазинах. Якщо зайти на головну сторінку магазину зі свого облікового запису та в режимі інкогніто, добірка рекомендацій відрізнятиметься.

Найчастіше динамічний контент відображається на екрані користувача разом із статичним. Наприклад, добірки товарів в онлайн магазинах: сама добірка формується динамічно, але фотографії товарів, які бачить користувач, - це статика.



Рис.13. Особливості сайтів з динамічним контентом

Оскільки динамічний контент постійно змінюється, його доставка користувачам організована складніше (рис.13).

* Браузер надсилає запит на сервер, щоб відобразити потрібний контент.
* Сервер надсилає запити до баз даних, сховищ та, можливо, якихось сторонніх сервісів, щоб зібрати унікальну відповідь.
* Сформована інформація надсилається користувачеві.

Через свою специфіку динамічний контент практично неможливо кешувати. Якусь частину даних можна зберегти в кеші на короткий час і вони не втратять своєї актуальності. Але в більшості випадків браузеру потрібно буде щоразу надсилати запити безпосередньо до джерела.

Інші особливості:

* **Персоналізація.** Веб-сервіси з динамічним контентом надають кожному користувачеві унікальний контент. При формуванні інформації враховуються різні фактори: наприклад, географічне знаходження користувача, тип браузера або пристрою, час доби, різні поведінкові фактори. Таким чином, те, що користувач побачить на екрані, більше відповідає його очікуванням.
* **Складна послідовність процесів.** Щоб сервер міг швидко формувати відповідь перед відправленням, потрібно правильно налаштувати роботу з базами даних та іншими сервісами. Це вимагатиме певних знань та зусиль від розробників.
* **Більше часу на завантаження.** На формування контенту витрачається більше часу, ніж надсилання готового файлу. Додати до цього складність кешування — виходить, що динамічний контент повинен завантажуватися довше, ніж статичний. Хоча це залежить від розміру даних: якщо це легкий текстовий контент, він завантажиться швидше, ніж важкі статичні відео.

## Статичні та динамічні додатки

Статичні та динамічні додатки, це програмовані сайти, які мають два рівні, що тісно взаємодіють один з одним

* **Клієнтська частина (Frontend).** Візуальна частина програми, яка виконує функції на стороні користувача (клієнта). Це все те, що відвідувач сайту бачить у браузері і з чим він може безпосередньо взаємодіяти (від дизайну до окремих елементів, на які можна натиснути, ввести текст, спонукати тощо).
* **Серверна частина (Backend).** Пов'язана з тим, що користувач візуально не може оцінити і до чого не має доступу. Це логіка роботи програми, що здійснюється на віддаленому сервері. Серверний код знаходиться на веб-сервері та відповідно до HTTP-запиту створює певну сторінку. Також він відповідає за зберігання різних даних, зокрема профілі клієнтів. В деяких випадках веб-додаток передбачає наявність лише одної серверної програми. Але є додатки, що вимагають кілька серверних програм та баз даних.

Статичні та динамічні додатки це послуги з переважанням певного типу контенту.

#### Приклади сайтів з переважанням статичного контенту:

* Сайти-візитки. Прості сайти з інформацією про компанії, яку не потрібно персоналізувати або часто змінювати.
* Прості блоги. Ресурси зі статтями без можливості залишати лайки, коментарі та інтерактивності. На таких сайтах тексти та картинки однакові для всіх і не змінюються.

Повністю статичних сайтів зараз дедалі менше. Навіть невеликі ЗМІ та блоги роблять індивідуальні добірки статей на головній сторінці, відображають рекомендації під прочитаним матеріалом, додають можливість залишати лайки та коментарі.

#### Приклади сервісів з великою часткою динамічного контенту:

* Соціальні мережі. Багато в них створюють самі користувачі: пости, листування, особисту інформацію про себе. Дані повинні постійно змінюватись і відображатися в реальному часі.
* Багато мобільних додатків. Наприклад, мобільні банки — кожен користувач бачить у них дані щодо своїх карток та рахунків, історію своїх операцій та індивідуальні банківські пропозиції. Відображається унікальна інформація для даного користувача.

#### Найчастіше у додатку є обидва типи контенту.

1. На сайті є і статичні та динамічні сторінки. Наприклад, типовий інтернет-магазин. Контент на головній сторінці, результати пошуку за каталогом - це в основному динаміка. Але на сайті є розділ "Про нас", є різні юридичні документи, публічна оферта - це на 100% статичний контент.
2. Обидва типи контенту можуть бути присутніми на одній сторінці, наприклад YouTube. Відео є типовим прикладом статичного контенту. Але на сторінці з відео справа буде добірка роликів на основі уподобань користувача, а знизу – лайки, кількість переглядів та коментарі – а це динамічний контент.

## Контрольні питання

1. Коротко пояснити суть клієнт-серверної взаємодії.
2. Які функції виконує веб-сервер?
3. Перелічіть популярні серверні операційні системи.
4. Назвати основне функціональне призначення браузера.
5. Перелічити основні компоненти браузера.
6. Яка функція покладена на модуль відображення браузера?
7. Перелічити основні модулі відображення для різних браузерів.
8. Пояснити структуру URL - Універсальний покажчик ресурсу
9. Коротко пояснити взаємодію браузера та веб-серверу.
10. Навести застереження щодо найменування різноманітних веб-об’єктів (файли, папки).