# 2.1. Служба Веб

**WWW (World Wide Web), Веб, Всесвітня Павутина** - це розподілена інформаційна система, що базується на концепції гіперпосилань.

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.

Служба Веб – світова інформаційна бібліотека, яка забезпечує засоби розміщення інформації і доступ до неї за допомогою кабелів і комп'ютерів (Інтернету). Веб є розподіленою інформаційною системою, оскільки інформація зберігається на величезній кількості веб-серверів.

Користувачі Інтернету отримують інформацію за допомогою браузера – це програма-клієнт для служби Веб. Програма-клієнт надсилає по комп'ютерній мережі запит до віддаленого сервера, на якому зберігається необхідний веб-ресурс. На віддаленому сервері знаходиться спеціальний програмний пакет – веб-сервер, який приймає і виконує запит. Після виконання поставлених завдань веб-сервер надсилає до програми-клієнта результати виконання або повідомлення про відмову, якщо завдання з різних причин є невиконаним (рис.1).



Рис.1. Клієнт-серверна взаємодія

Така взаємодія називається «клієнт-сервер» і для служби Веб відбувається за прикладним протоколом HTTP (Hyper Text Transfer Protocol). Спочатку HTTP використовувався тільки як протокол передачі гіпертексту (тексту з перехресними посиланнями). Однак пізніше стало зрозуміло, що він є доречним для передачі даних між користувачами. Протокол був доопрацьований для нових завдань і став використовуватися повсюдно.

Незважаючи на свою функціональність у HTTP є один дуже важливий недолік - незахищеність. 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 - 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-сайти — це сукупності електронних документів (файлів), що об'єднані під однією доменною адресою і знаходяться на одному сервері за IP-адресою.

Спочатку веб-сайти були лише збіркою статичних документів. На сьогодні веб-документи окрім тексту та графіки містять анімацію, відео і звук. Основним у веб-документах є гіперпосилання, які можуть бути як внутрішніми перехресними посиланнями, так і посиланнями на інші веб-документи, що зберігаються на різних веб-серверах.

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 і кожен може з ним ознайомитись (<http://info.cern.ch/>).



Рис. 3. 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 просто і непоказно, але має історичне значення.

Розвиток веб-сайтів безпосередньо пов'язаний з популяризацією Інтернету. В розвитком нових технологій та інструментів вдосконалювалися та набирали популярності веб-додатки. Перші спроби програмувати веб-сторінки почалися задовго, як Інтернет став масовим. Ще в 1987 році Ларрі Уолл розробив Perl, серверну мову сценаріїв.

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

Завдяки HTML (мові гіпертекстової розмітки), CSS (каскадним таблицям стилів), Javascript (для пожвавлення статики сторінок), серверним технологіям (Apache , Nginx , AJAX) та мовам програмування (Python , Java , PHP тощо) стало можливим робити потужні масові веб-додатки, які мають великий функціонал.

### Веб-сайт vs веб-додаток

**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 оцінювати пости інших користувачів, спілкуватися в чатах та відеоконференцях, здійснювати пошук на основі своїх інтересів.



Рис. 2. Типовий вигляд веб-додатку

Веб-додатки створюють з різною метою: взаємодія з користувачами, налагодження бізнес-процесів всередині компанії, організація навчального процесу, проведення фінансових операцій, синхронізація роботи над спільними документами.

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).
* Багато інших програм.

#### 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 усі 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 |

#### Особливості веб-додатків:

* Можуть працювати як на смартфоні, так і на персональному комп'ютері.
* Практично незалежні від заліза.
* За функціоналом майже не поступаються десктопним аналогам.
* 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 зменшують витрати як для бізнесу, так і для кінцевого користувача, оскільки вимагають менше ресурсів (людських та обладнання) для роботи і обслуговування.
* Веб-додатки прості у використанні, не вимагають спеціальних знань та вмінь, що дозволяє працювати з різною аудиторією.
* Веб-додатки рентабельні, оскільки дешевші за класичні програмні рішення, що дозволяє їх використовувати компаніям з обмеженим бюджетом.
* Веб-додаток можна легко підлаштувати під потреби будь-якого бізнесу.

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 (напр., google.com). Копія оригінального сайту у такому разі називається дзеркалом. Зазвичай, великі сайти (веб-портали) використовують кілька доменів, щоб логічно відокремити різні види послуг (mail.google.com, news.google.com, maps.google.com). Іноді, окремі домени виділяють для іншої мовної версії. Наприклад, *google.com.ua* і *google.pl* логічно є сайтом Google з різними мовами інтерфейсу, але технічно це різні сайти.

Об'єднання кількох сайтів під одним доменом є характерним для безкоштовних хостингів. Тут, зазвичай, використовується домен третього рівня (*mysite.example.com*), але іноді для ідентифікації сайтів в адресі після вказування доменної адреси хосту зазначено назву сайту (*example.com/my-site-name/*).

## Сервер

**Сервер** — це комп'ютер зі встановленим на ньому спеціальним програмним забезпеченням, що має власну ІР-адресу. 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.

Основним завданням веб-серверу є прийом від клієнта (браузера) запиту на завантаження ресурсів (сторінок, графічних зображень, фільмів, музики тощо), пошук цих файлів на дисках серверного комп'ютера і надсилання знайдених файлів назад до комп’ютера користувача. Потужні веб-сервери можуть виконувати багато інших дій над потрібними об’єктами.

* Отримання та надсилання запитів.
* Запуск сценаріїв, що написані серверними мовами програмування.
* Складання веб-сторінок з даних, які витягуються з бази даних.
* Надсилання веб-сторінок до користувача.
* Захист інформації від спотворення та знищення.
* Ідентифікація клієнтів.
* Облік активності користувачів.
* Обробка запитів інших видів (mailto, FTP тощо).

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-сервер містить кілька компонентів, які контролюють доступ веб-користувачів до розміщених на сервері файлів.

### Веб-сервер (апаратне забезпечення)

Залежно від типу хостингу сервер може зберігати один або багато веб-сайтів. Там містяться файли веб-сайту, а саме всі HTML-документи та пов'язані з ними ресурси, включаючи зображення, CSS-стилі, JavaScript-файли, шрифти та відео.

Технічно, можна розмістити всі ці файли на будь якому комп'ютері, але варто зберігати їх на спеціальному сервері, який:

* Завжди запущений та працює.
* Завжди підключений до Інтернету через швидкісні магістралі.
* Має постійну IP адресу (не всі провайдери надають статичну IP-адресу для домашнього підключення).
* Обслуговується третьою, сторонньою компанією.

З усіх цих причин пошук хорошого хостинг-провайдера є ключовою частиною створення сайту.

### Операційні системи для серверів

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



Рис.1. Рейтинг 2023 року серверних операційних систем

* **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, достатню стабільність роботи і відмінну технічну підтримку. Оновлення і доповнення виходять щомісяця.
* **Ubuntu**. Ця операційна система є самою популярною з сімейства GNU/Linux, оскільки її інтерфейс (максимально наближений до інтерфейсу Windows) є інтуїтивно зрозумілим, простим і наочним. Але за простотою інтерфейсу ховається ядро, яке не поступається за своєю продуктивністю системі Debian, на основі якої і побудована Ubuntu.
* **CentOS**. Серверний дистрибутив, що спонсорується компанією Red Hat і підтримується спільнотою. Використовується в основному для тестування роботи нових продуктів. Проект є великий тестовим майданчиком для розробників різних додатків, які потім будуть використовуватися на комерційній основі в ОС Red Hat. Вважається найкращим, безкоштовним аналогом RedHat Enterprise Linux.
* **Debian**. Дана операційна система, яка об'єднує кращі технології з високою швидкістю роботи, підійде практично під будь-які проекти. Зараз Debian вважається одним з надійніших дистрибутивів Linux, самим продуманим і має одну з кращих систем управління пакетами.
* **FreeBSD**. На відміну від абсолютної більшості ОС - FreeBSD спочатку створювалася як серверна операційна система, що спеціалізована під встановлення на робочі станції. Її ядро ​​має цільну, закінчену структуру, що робить роботу сервера безпрецедентно стабільною. Система є повністю безкоштовною, має хорошу підтримку, проста у встановленні, налаштуванні і адмініструванні. Має значну кількість додаткових програм і модулів, які можна встановити. За деякими оцінками, більше 40% всіх серверів працюють саме під керуванням системи FreeBSD.

### 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, обробляє їх і надсилає до клієнта результати обробки. Клієнтом у цьому випадку є браузер на комп'ютері користувача або мобільний додаток на смартфоні або планшеті.

Веб-сервери доставляють клієнту лише статичний HTML-контент, такий як документи, зображення, відео, шрифти тощо. Зазвичай, веб-сервери не обробляють динамічний контент, але опціонально, деякі веб-сервери дозволяють додавати компоненти, що дозволяють працювати з динамічним контентом.



Рис.2. Функціонування веб-сервера зі статичним контентом

Веб-сервер відповідає на запити користувача щодо доставки контенту, а також за спілкування з зовнішніми ресурсами, наприклад СУБД та Інтернет-сервісами.

## Типи веб-серверних програм

На рис. представлено рейтинг 2023 року веб-серверів з найбільшою кількістю користувачів по всьому світу (https://uk.hostadvice.com/marketshare/server/)



Рис.4. Рейтинг 2023 року популярності веб-серверів

Серверні програми, не мають інтерфейсу користувача і «спілкуються» лише з комп’ютером-сервером, приймають від нього надіслані користувачем дані і повертають до нього результат.

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

**Nginx і Apache -** це два найпопулярніших веб-сервера з відкритим вихідним кодом, які використовуються для розміщення сайтів по всьому світу. Разом їх частка становить понад 50% всього трафіку в Інтернеті. 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.

#### Apache

Apache HTTP Server розробники представили у 1996 року. Це один із перших веб-серверів на ринку. Після запуску програмного забезпечення у вільний доступ, Apache випередив усіх конкурентів, і продовжує лідирувати в наші дні. У світі його використовують на 40,89% сайтів.

Ще нещодавно Apache використовували як основний і єдиний веб-сервер. Зараз, коли даних побільшало, а разом з ними збільшилася і кількість підключень, його завжди достатньо. Apache часто використовують як головний веб-сервер та комбінують з іншими рішеннями. Веб-сервер знаходиться у вільному доступі та використовувати його можна безкоштовно.

Apache – кросплатформне програмне забезпечення. Підтримує всі основні операційні системи, у тому числі Microsoft Windows, Mac OS та UNIX-подібні ОС. З точки зору архітектури Apache складається з ядра, яке виконує основні завдання, і велика кількість додаткових модулів. Ядро працює завжди. Модулі можна підключати та вимикати, що дозволяє контролювати функціональність сервера.

#### Nginx

Nginx – веб-сервер з відкритим кодом, який представлено у 2004 році., на сьогодні на його частку припадає 28% всіх майданчиків світу. Nginx використовують не тільки власники рядових сайтів, а й гіганти ринку – Facebook, Pinterest, Netflix, Tumblr, Instagram та багато інших. Як і Apache, Nginx – open-source продукт. Але його можна використовувати тільки на UNIX-подібних операційних системах Microsoft Windows, через що цей веб-сервер трохи поступається світовому лідеру.

При розробці було враховано деякі вагомі недоліки вже існуючих веб-серверів, у тому числі Apache. Завдяки цьому Nginx чудово справляється зі збільшенням навантаження і легко витримує кілька тисяч запитів одночасно.

#### IIS

Microsoft IIS (Internet Information Services) – пакет серверного ПЗ, пропрієтарний набір сервісів від компанії Microsoft. IIS розповсюджується з пакетом Windows NT. IIS підтримує протоколи HTTP, HTTPS, FTP, POP3, SMTP, NNTP.

IIS або Internet Information Services – продукт Microsoft. Отже, використовувати його можна лише на операційній системі Windows. Версія 1.0 була випущена 1995 року. Наразі веб-сервер IIS використовують на 11% всіх сайтів у світі.

Технічно це пакет серверів із головним компонентом у вигляді веб-сервера. На відміну від Apache та Nginx, IIS не представлений у вільному доступі, але постачається разом із ОС з лінійки Windows NT.

За функціональністю та продуктивністю IIS знаходиться приблизно на одному рівні з більш популярними безкоштовними рішеннями. Велику увагу розробники приділяють безпеці та покращують її з випуском кожної нової версії – у цьому питанні IIS перевершує Apache та Nginx. Через те, що рішення є комерційним, користувачі можуть розраховувати на підтримку Microsoft, що особливо важливо для великого бізнесу. Істотні мінуси – робота лише на одній платформі та необхідність купувати ліцензію Windows. Хоча другий фактор говорить про те, що IIS - умовно-безкоштовний, оскільки докуповувати продукт не потрібно, він уже йде в комплекті.

## Браузери

Браузер - це вікно у світ Інтернету, без якого не може обійтися жоден користувач. Практично всі популярні сучасні браузери поширюються безкоштовно або у комплекті з іншим застосуванням.

Спочатку браузери призначалися лише для перегляду гіпертекстових документів. Проте, в 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 використовують прикладний протокол HTTP, щоб надіслати запит до сервера для отримання вмісту сторінки і отримати цей вміст. Після того, як дані отримано, вони обробляються движком браузера, і результат відображається у вікні. Те, яким чином браузер обробляє і відображає HTML-файли, визначено специфікаціями HTML і CSS. Вони розробляються Консорціумом W3C, який впроваджує стандарти для Інтернету.

### Основне функціональне призначення браузера

Основним призначенням браузера є відображення веб-ресурсів. Для цього на сервер надсилається запит, а результат виводиться у вікні браузера. Під ресурсами в основному розуміють HTML-документи, однак, це також може бути PDF-файл, картинка чи інший зміст. Розташування ресурсу визначається за допомогою URL (Uniform Resource Locator).

### URL - Універсальний покажчик ресурсу

IP-адреса чи доменна адреса дозволяють однозначно ідентифікувати веб-сервер в мережі Інтернет, але на сервері може бути багато різної інформації в різних форматах, наприклад, у вигляді файлів, електронних повідомлень, сторінок тощо.

Для безпомилкового отримання потрібної інформації і 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.

<http://www.site.lviv.ua/documents/page.html> або <http://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.
* **Назву служби.** В даному прикладі це служба Веб - www
* **Доменну або IP-адресу**, яка однозначно ідентифікує веб-сервер в мережі Інтернет, на якому розміщено потрібний сайт чи інший ресурс.
* **Шлях**, що складається з імен 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.

Браузер можна використати як програму клієнт для інших служб, наприклад для служби FTP. В цьому випадку до ресурсів FTP-серверу можна доступитися, якщо ввести в адресному полі браузера *ftp://ftp.site.lviv.ua*

### Інтерфейс браузера

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.

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.

### Основні компоненти браузера

Рушій браузера

Ї

Ї

Ї

Ї

Ї

Ї

Ї

Ї

Ї

Ї

Ї

Користувацький інтерфейс

Ї

Ї

Ї

Ї

Ї

Ї

Ї

Ї

Ї

Ї

Ї

Модуль рендерингу (відображення)

Ї

Ї

Ї

Ї

Ї

Ї

Ї

Ї

Ї

Ї

Ї

Мережний механізм

Java Script інтерпретатор

Внутрішній UI

Сховище
даних (кеш)

Ї

Ї

Ї

Ї

Ї

Ї

Ї

Рис. 2. 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 відображає 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. 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)** - необхідно для збереженості процесів. Браузер зберігає на твердий диск дані різних типів, наприклад файли cookie.

### Модуль відображення (рендерингу)

Модуль відображення відповідає за виведення запитаного змісту на екрані браузера.

За замовченням він здатний відображати HTML і XML-документи, а також зображення. Спеціальні модулі (розширення для браузерів) роблять можливим відображення іншого змісту, наприклад PDF-файлів.

### 5 основних модулів відображення для рендерингу сторінок

|  |  |
| --- | --- |
|  | **Trident** - розроблено компанією Microsoft. Використовується в багатьох браузерах: Internet Explorer, Windows Mobile, AOL Browsers |
| Microsoft_Edge_logo | **EdgeHTML** - браузерний модуль від Microsoft, який втілено в Microsoft Edge. Є гілкою від Trident з видаленням застарілого коду і заміною його на більш сучасний. Вперше реалізований в Internet Explorer 11 як частина Windows 10 Preview 9926 build. |
|  | **Gecko** - движок з відкритим вихідним кодом, який було 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. |

Значна частина відмінностей браузерів випливає з відмінностей у модулі відображення сторінок. Ці відмінності приводять до того, що сторінки в різних браузерах виглядають дещо по-різному. Chrome, на відміну від більшості браузерів, використовує кілька екземплярів модуля відображення, по одному в кожній вкладці, які представляють собою окремі процеси.

Модулі відображення постійно вдосконалюються, але старші версії браузерів працюють на попередніх версіях движків. Тому, розробнику слід тестувати сторінки в кількох браузерах, а також в кількох версіях цих браузерів.

## Взаємодія браузера і сервера

При введенні адреси певного веб-ресурсу, від браузера надходить запит до серверу. Сервер отримує цей запит, обробляє його, витягує з бази даних відповідні відомості та надсилає назворот до клієнту. На стороні клієнта завантажується сторінка, яка представляється в необхідному вигляді за допомогою HTML, JS та CSS.

Взаємодія користувача зі сторінкою (клікі по кнопці, надсилання форми, вибір певної опції, пошуковий запит тощо) також призводить до формування запитів на сервер.



Рис.6. 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, відбуватися дозавантаження даних відразу без перезавантаження програми. У деяких випадках дані кешуються для прискорення обробки повторних запитів.

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.

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 тому самому 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.



Класичний приклад статики - це картинки, відео, текстові файли, HTML-файли та інші подібні дані. Всі вони спочатку були кимось створені, розміщені в інтернеті і не можуть змінитися.

Приклади використання статичного контенту:

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

### Особливості статичного контенту



Головна особливість статики – її просто доставляти та кешувати .

* Коли користувач заходить сайт, браузер відправляє запит 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 вийняти файл з диска і відправити його — і більше.
* Швидке завантаження. Статика легко і швидко доставляється, її можна без проблем кешувати. Значить, і часу на завантаження потрібно небагато. Але тут відіграє роль та обсяг контенту: важкі відео та картинки будуть передаватися довше, ніж легкий текст.
* 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*).

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.

### 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 його параметрів.



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:

* Контент у соціальних мережах здебільшого динамічний. Ваша стрічка у соціальній мережі буде зовсім не такою, як стрічка вашого друга.
* 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 одного магазину зі свого облікового запису та в режимі інкогніто, добірка рекомендацій відрізнятиметься.

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



Оскільки динамічний контент постійно змінюється, його доставка користувачам організована складніше.

* 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 втратить своєї актуальності. Але в більшості випадків браузеру потрібно буде щоразу надсилати запити безпосередньо до джерела.

Інші особливості:

* Персоналізація. Веб-сервіси з динамічним контентом дають кожному користувачеві унікальний досвід. При формуванні інформації враховуються різні фактори: наприклад, географія, тип браузера або пристрої, 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 відправкою, потрібно правильно налаштувати роботу з базами даних та іншими сервісами. Це вимагатиме певних знань та зусиль від вашої IT-команди.
* Більше часу на завантаження. На формування контенту визначення йде більше часу, ніж відправлення готового файлу. Додайте до цього складність кешування — виходить, що динамічний контент завантажуватися повинен довше, ніж статичний. Хоча це знову ж таки 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)

Frontend призначений для створення візуальної частини програми, яка виконує функції на стороні користувача. 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).

Для написання клієнтського коду зазвичай використовують комбінацію HTML (відповідає за структурування змісту сторінки), CSS (надає веб-сторінці певного вигляду), JavaScript (клієнтська мова програмування). Ці засоби вбудовано в браузери, тому немає потреби окремо налаштовувати пристрій. Клієнтський код взаємодіє із сервером лише через протокол HTTP і не може безпосередньо отримувати інформацію з сервера.

Щоб відчути, що представляє frontend-розробка, достатньо в браузері відкрити будь-який сайт, натиснути праву кнопку миші і вибрати пункт "Подивитися код" (Inspect). Відкриються "Інструменти розробника", де можна побачити HTML-розмітку сайту, а у вкладці Sources знаходяться таблиці стилів (CSS) та JavaScript сценарії. Тут описуються кольори, картинки, таблиці та інші сторінки, поведінка при взаємодії з ними.

Розробка клієнтської частини додатків пов'язана з низкою фронтенд-фреймворків (Angular, React тощо).

### Серверна частина (Backend)

Backend пов'язаний 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. Розробка серверної частини додатків пов'язана з низкою мов програмування (РНР, Java, Python, C#, C++), а спрощення такої розробки досягається використанням бекенд-фреймворків (Django, Laravel тощо) та веб-серверів (Nginx , Node.js тощо).

Серверний компонент складається з 2 частин – логіки додатку (сервер додатку) та бази даних. Перший – головний центр управління веб-додатком, другий – місце, де зберігається інформація.

Бази даних потрібні для зберігання масивів даних, які при запиті користувача виймаються та відображаються у веб-додатку. На практиці можуть використовуватися різні бази даних, найбільш популярні з яких: PostgreSQL, MySQL, MongoDB. Для роботи з базами даних існує багато бібліотек, орієнтованих на різні серверні мови програмування.

Отже, використовуючи веб-додаток маємо справу з клієнтською та серверною частинами. У деяких випадках веб-додаток передбачає наявність лише одного сервера. Але є додатки, що вимагають кілька серверів 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.

Проте 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. В усіх буде своя унікальна інформація.

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

1. На сайті є і статичні та динамічні сторінки. Візьмемо, 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. Візьмемо YouTube. Ми вже сказали, що відео є типовим прикладом статичного контенту. Але на сторінці з відео справа буде добірка роликів на основі ваших уподобань, 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).

## Використані джерела

1. Як влаштовано Інтернет <https://thecode.media/how-internet-works-1/>
2. Клієнт-серверна технологія https://galtsystems.com/blog/start/klient\_server\_o\_tekhnologii\_prostymi\_slovami/
3. Як працює Веб <https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/How_the_Web_works>
4. Установка базового програмного забезпечення <https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/Installing_basic_software>
5. Що таке веб-сервер <https://servergate.ru/articles/chto-takoe-veb-server-kakoy-iz-nikh-vybrat-i-kak-organizovat-ego-rabotu/>
6. Серверні операційні системи <https://server-shop.ua/kakuyu-os-vyibrat-dlya-servera.html>
7. Визначення та функції браузера <https://sendpulse.ua/ru/support/glossary/browser>
8. Функції та можливості браузера <https://timeweb.com/ru/community/articles/chto-takoe-brauzer>
9. Модуль відображення браузера <https://habr.com/ru/post/459814/>
10. Небезпечні розширення для браузерів <https://www.kaspersky.ru/blog/browser-extensions-security/19575/>
11. Браузерні війни <https://habr.com/ru/post/541180/>