### ЛЕКЦІЯ 5. Клієнтські технології. Веб-клієнти: браузери, мобільні додатки. Клієнтські бібліотеки та фреймворки (Ajax, Fetch API). Інтерактивність клієнтів

Клієнтські технології є важливою складовою клієнт-серверної архітектури, адже вони забезпечують взаємодію користувачів з веб- та мобільними додатками. У сучасних веб-системах клієнти мають виконувати більше завдань, таких як обробка даних та забезпечення інтерактивності. Веб-клієнти представлені переважно браузерами, а також мобільними додатками, які взаємодіють з серверною частиною. Клієнтські бібліотеки та фреймворки, такі як Ajax та Fetch API, відіграють важливу роль у забезпеченні інтерактивності й динамічного обміну даними з сервером.

**1. Веб-клієнти: Браузери**

Браузери — це програмне забезпечення, яке дозволяє користувачам переглядати та взаємодіяти з веб-сторінками. Вони виконують функцію клієнта в клієнт-серверній архітектурі, надсилаючи запити до сервера та отримуючи відповіді у вигляді HTML, CSS, JavaScript та інших ресурсів.

Основні компоненти браузера:

* **Інтерпретатор HTML**: аналізує HTML-документ і відображає його на екрані користувача.
* **CSS-движок**: забезпечує відображення стилів, що описують зовнішній вигляд елементів.
* **JavaScript-движок**: виконує JavaScript-код для надання веб-сторінці інтерактивних можливостей.
* **Мережевий модуль**: відповідає за обмін даними з сервером за допомогою HTTP(S).
* **Кешування**: браузер кешує дані для швидшого завантаження повторно відвіданих сторінок.

Браузери забезпечують виконання як статичного, так і динамічного контенту, що дозволяє користувачам отримувати інтерактивний досвід. Також вони підтримують додаткові можливості, такі як локальне зберігання даних (Local Storage), сеансове зберігання (Session Storage), файли cookie для збереження стану сесії та взаємодії з веб-додатками.

Основні браузери:

* **Google Chrome** — один із найпоширеніших браузерів, побудований на основі движка Chromium.
* **Mozilla Firefox** — браузер з відкритим кодом, відомий своєю безпекою та налаштовуваністю.
* **Microsoft Edge** — браузер, який перейшов на платформу Chromium.
* **Safari** — браузер для пристроїв Apple (iPhone, iPad, Mac).

**2. Мобільні додатки**

Мобільні додатки стають дедалі важливішими клієнтами в сучасній архітектурі клієнт-серверної взаємодії. Вони можуть бути розроблені як **нативні**, **гібридні** або **веб-додатки**.

* **Нативні мобільні додатки** розробляються для конкретної операційної системи (iOS або Android) і безпосередньо використовують ресурси пристрою (GPS, камеру, сенсори).
* **Гібридні додатки** створюються з використанням веб-технологій (HTML, CSS, JavaScript), але вбудовуються у контейнер, що дозволяє їм працювати на мобільному пристрої та використовувати його функціонал. Прикладом таких фреймворків є **React Native** та **Ionic**.
* **Веб-додатки** — це додатки, доступні через браузер мобільного пристрою, що мають інтерфейс, схожий на нативні програми, але не потребують інсталяції.

Мобільні додатки взаємодіють із сервером через API (Application Programming Interface) та обмінюються даними з використанням таких протоколів, як HTTP та HTTPS.

**3. Клієнтські бібліотеки та фреймворки**

**3.1 Ajax (Asynchronous JavaScript and XML)**

**Ajax** — це набір технологій, що дозволяють веб-сторінкам надсилати запити до сервера асинхронно, без перезавантаження всієї сторінки. Це створює більш інтерактивний і швидкий користувацький досвід.

Ключові компоненти Ajax:

* **JavaScript** — для ініціації запитів до сервера.
* **XMLHttpRequest (XHR)** — об'єкт для відправки HTTP-запитів та отримання відповідей.
* **XML або JSON** — формат для обміну даними між клієнтом та сервером.

Переваги Ajax:

* **Швидкість та інтерактивність**: Користувачі не повинні чекати на повне перезавантаження сторінки, що покращує користувацький досвід.
* **Мінімізація трафіку**: Передаються лише необхідні дані, а не вся сторінка.

Недоліки Ajax:

* **Складність**: Використання Ajax може ускладнити структуру коду через обробку асинхронних операцій.
* **Проблеми з індексацією**: Деякі пошукові системи можуть не індексувати динамічно змінюваний контент.

**3.2 Fetch API**

**Fetch API** — це новіший стандарт для виконання HTTP-запитів у JavaScript, що є більш сучасною альтернативою XMLHttpRequest. Fetch API використовує обіцянки (Promises), що полегшує роботу з асинхронними операціями.

Приклад використання Fetch API:

javascript

fetch('https://api.example.com/data')

.then(response => response.json())

.then(data => console.log(data))

.catch(error => console.error('Помилка:', error));

Переваги Fetch API:

* Простота використання у порівнянні з XHR.
* Покращена підтримка асинхронних запитів та обіцянок.
* Краще управління помилками.

Fetch API підтримує обробку HTTP-запитів на GET, POST, PUT, DELETE та інші методи, що робить його важливим інструментом для роботи з API в веб-додатках.

**4. Інтерактивність клієнтів**

Інтерактивність у веб-до ого досвіду. Інтерактивні веб-додатки дозволяють динамічно змінювати вміст сторінки без перезавантаження та реагувати на дії користувачів у реальному часі.

**4.1 JavaScript і DOM (Document Object Model)**

JavaScript є основною мовою для забезпечення інтерактивності на клієнтській стороні. За допомогою JavaScript можна динамічно змінювати елементи сторінки, що визначаються у структурі DOM.

Приклад інтерактивності через JavaScript:

javascript

document.getElementById('button').addEventListener('click', function() {

document.getElementById('text').innerHTML = 'Кнопку натиснуто!';

});

**4.2 Реактивні фреймворки**

Сучасні бібліотеки та фреймворки, такі як **React**, **Vue.js**, **Angular**, забезпечують потужну інтерактивність, дозволяючи створювати динамічні односторінкові додатки (SPA). Ці фреймворки використовують концепцію "реактивності", що дозволяє автоматично оновлювати інтерфейс, коли змінюються дані.

* **React**: Бібліотека для створення користувацьких інтерфейсів, що дозволяє динамічно оновлювати компоненти сторінки на основі стану.
* **Vue.js**: Легкий фреймворк, що підтримує реактивність та дозволяє швидко створювати інтерактивні додатки.
* **Angular**: Повноцінний фреймворк для створення веб-додатків з потужними можливостями для побудови клієнтських інтерфейсів.

**Висновки**

Клієнтські технології відіграють ключову роль у клієнт-серверній архітектурі, забезпечуючи ефективну взаємодію між користувачем і сервером. Сучасні веб-клієнти, такі як браузери та мобільні додатки, використовують різноманітні бібліотеки та фреймворки для надання інтерактивних функцій і динамічного контенту. Інструменти, такі як Ajax і Fetch API, дозволяють покращити швидкість і ефективність взаємодії з сервером, забезпечуючи зручність користувачів.

# У чому відмінність вебдодатків від мобільних додатків

Вебдодаток — це програма, яка працює в браузері, як сайт.

[Мобільний додаток](https://brander.ua/what-we-offer/application-development) — той, який завантажується з App Store або Google Play.

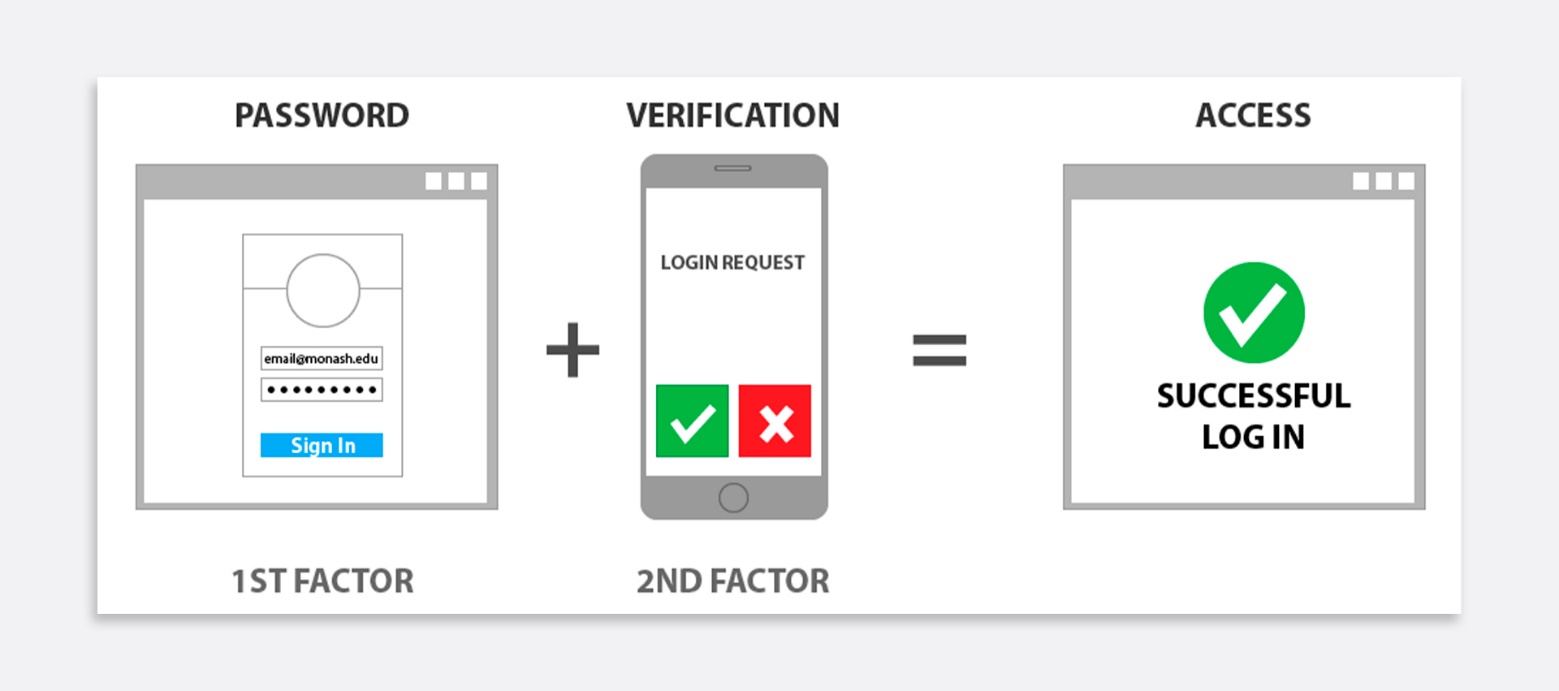
Дуже просто, чи не так? Якщо у вас на робочому столі смартфона або планшета є окрема іконка для запуску програми — це мобільна програма, яка, до речі, може частково або повністю працювати й без інтернету. Якщо для того, щоби скористатися функціями програми, потрібно зайти до Chrome, Safari або іншого вашого улюбленого браузера — це вебдодаток.

## **Що таке вебдодаток**

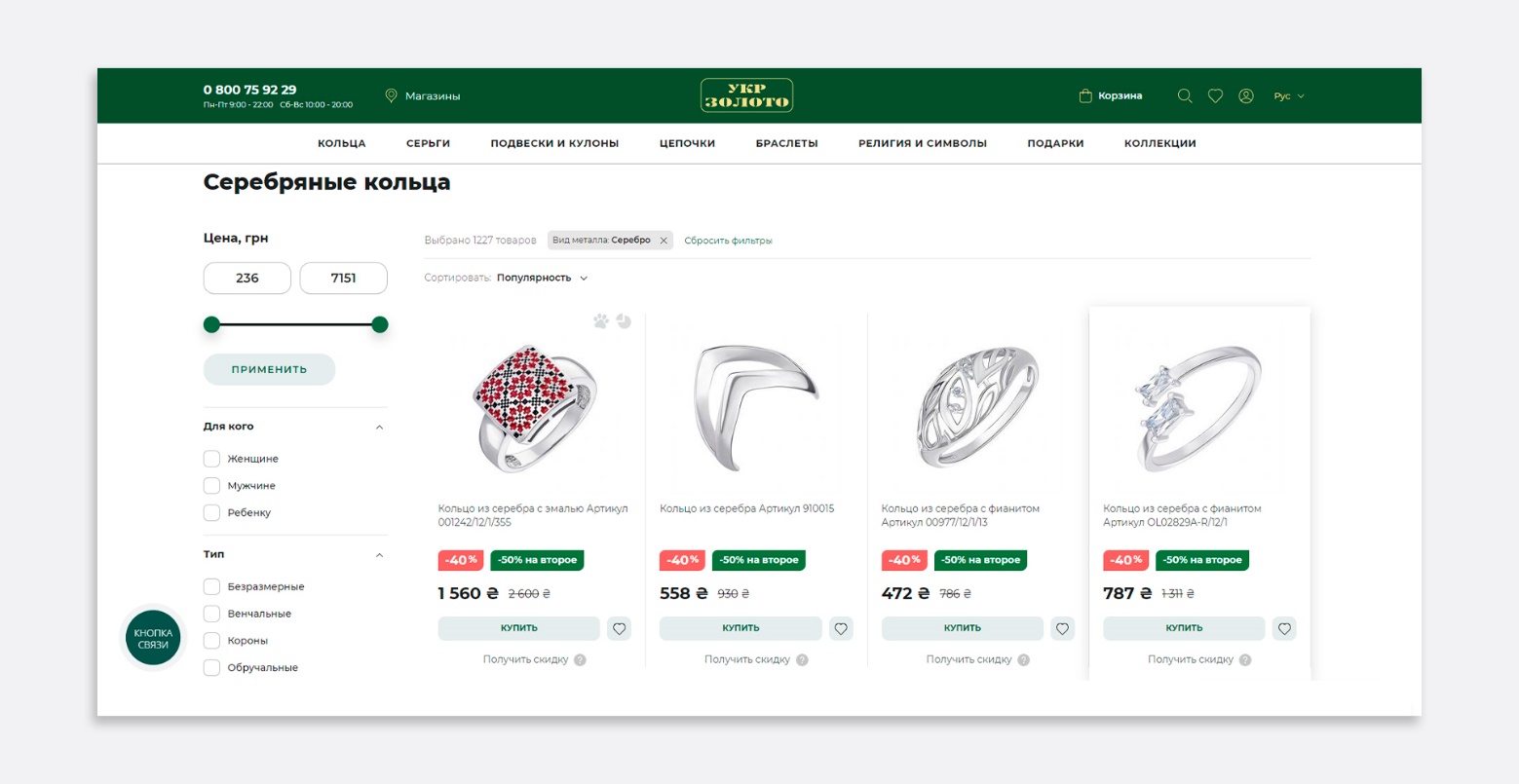
Але, хіба браузери не відкривають сайти? Виходить вебдодаток це просто модна назва для сайту? А от і не виходить! Сучасні браузери відкривають і сайти та вебдодатки. До перших, відноситься все «просте» — портали новин, особисті блоги й корпоративні сторінки, лендінги і т.д. До других, усе «складне», наприклад, ваша сторінка у Facebook, кабінет у Google Analytics або вебверсія Zoom. А у якій категорії тоді інтернет-магазини? У тій, де захотілося їх власникам!

Три базові відмінності вебдодатку від сайту:

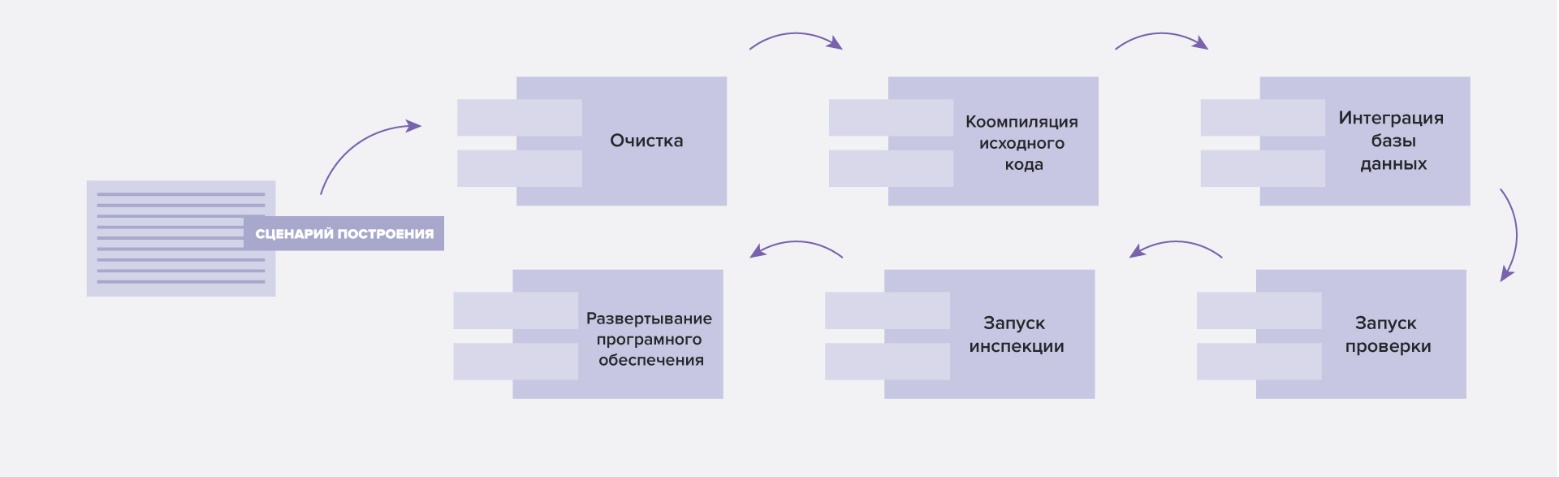
1. Розширена автентифікація. Користувачі вебзастосунків можуть не тільки переглядати, але і створювати/завантажувати/змінювати контент.

[](https://brander.ua/assets/2022-02/v-chem-otlichie-veb-prilozhenij-ot-mob-prilozhenij-3.jpg)

1. Складні функції. «Прості» функції — це ті, що доступні в різних [CMS](https://brander.ua/blog/cms-ta-framework-shcho-obraty-dlya-internet-mahazynu), на базі яких можна зібрати сайт, як конструктор. Умовно, це все, що ми бачимо в стандартних інтернет-магазинах — відсортувати товари, відправити їх у кошик та сплатити. Для будь-якої нестандартної бізнес-логіки, починаючи від можливості вести щоденник або забронювати готель, і тим більше обмінюватися повідомленнями, створюється вебдодаток на основі фреймворків.

[](https://brander.ua/assets/2022-02/v-chem-otlichie-veb-prilozhenij-ot-mob-prilozhenij-4.jpg)

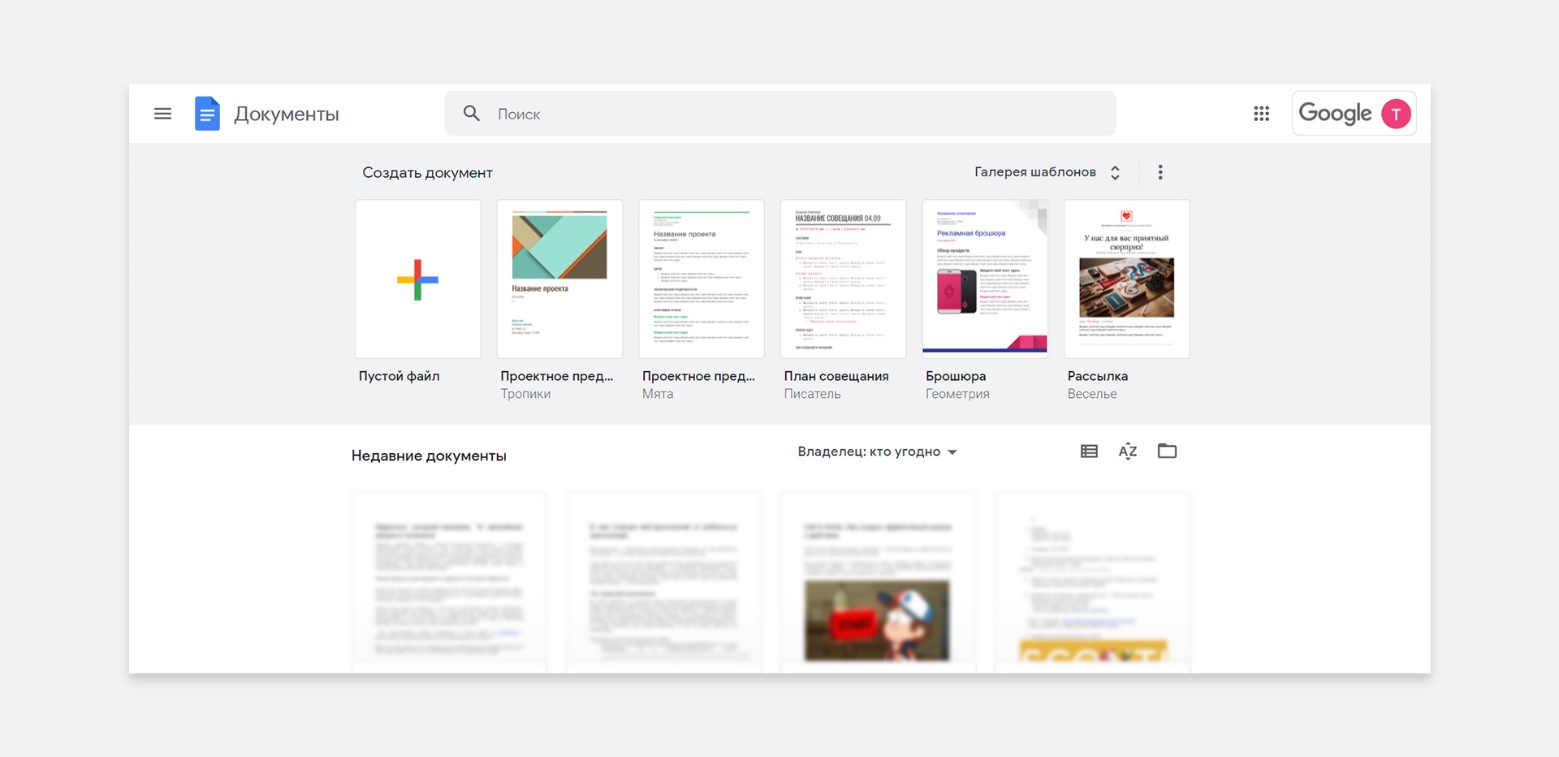
1. Деплоймент. Змінити щось у звичайному сайті просто, як оновити HTML-код сторінок. Вебдодаток вимагає повноцінної компіляції та розгортання нового програмного забезпечення.

[](https://brander.ua/assets/2022-02/v-chem-otlichie-veb-prilozhenij-ot-mob-prilozhenij-5.jpg)

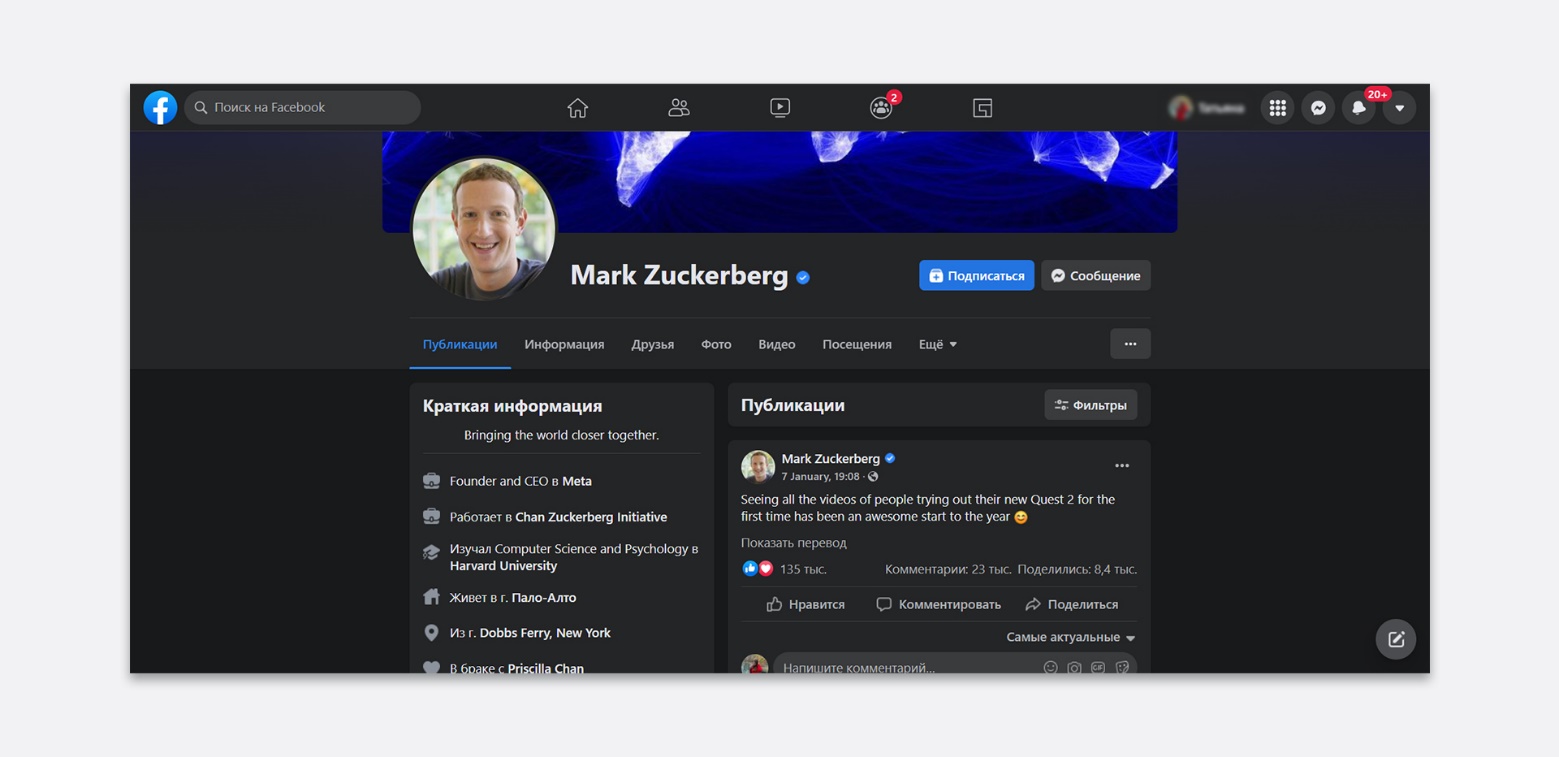
## Приклади вебзастосунків

Де ми можемо побачити приклади всього цього? Та скрізь! Вебпрограми — це:

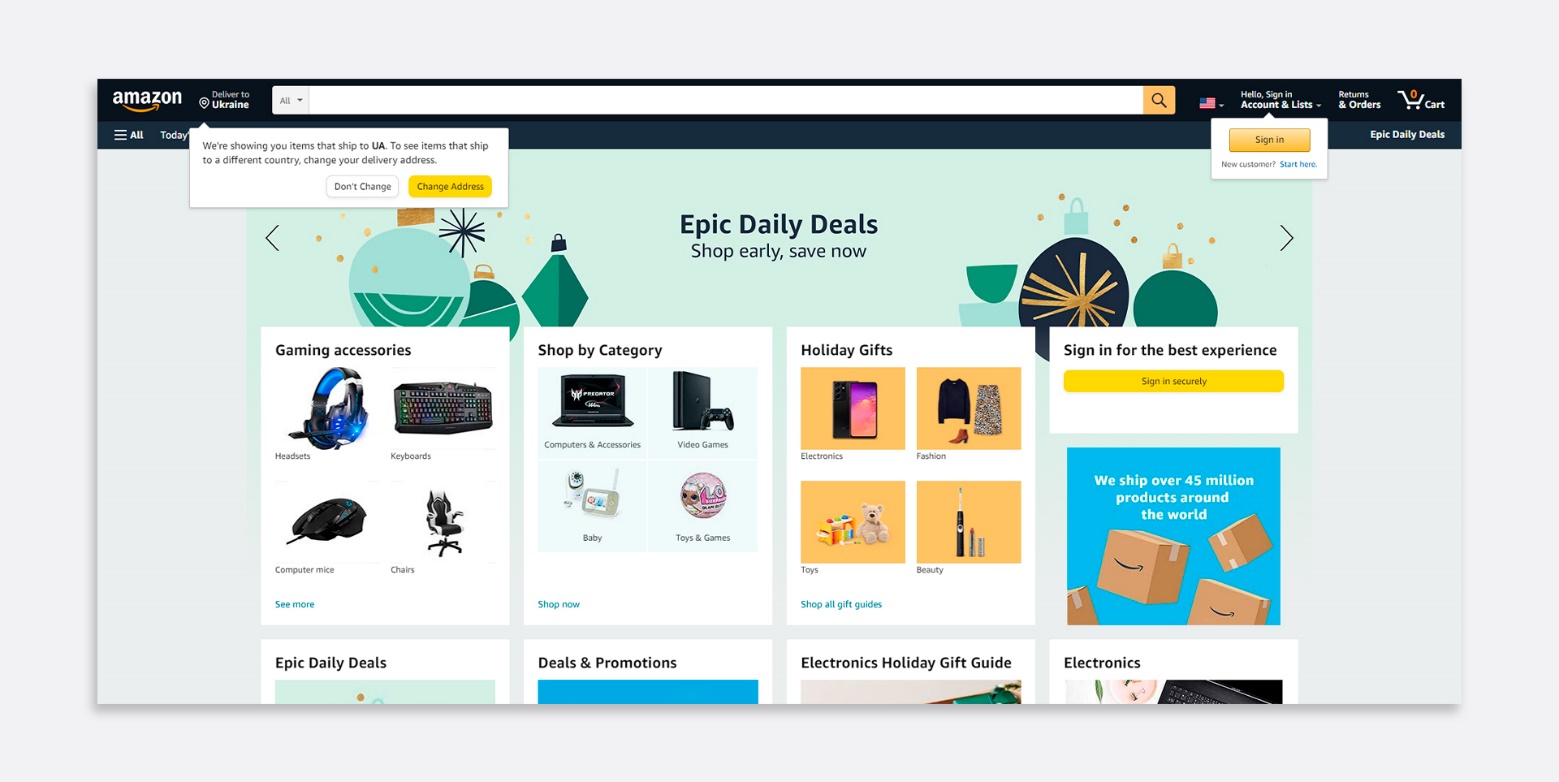
1. Онлайн-редактори. Найбільш наочний і відомий усім приклад — Google Docs і Sheets, а також аналогічні їм Word та Excel 360, якими можна користуватися онлайн у браузері. Сюди ж відносяться вебпрограми для роботи з графікою, від Photoshop і Canva, до всіляких простих сервісів для накладання фільтрів на фото.

[](https://brander.ua/assets/2022-02/v-chem-otlichie-veb-prilozhenij-ot-mob-prilozhenij-6.jpg)

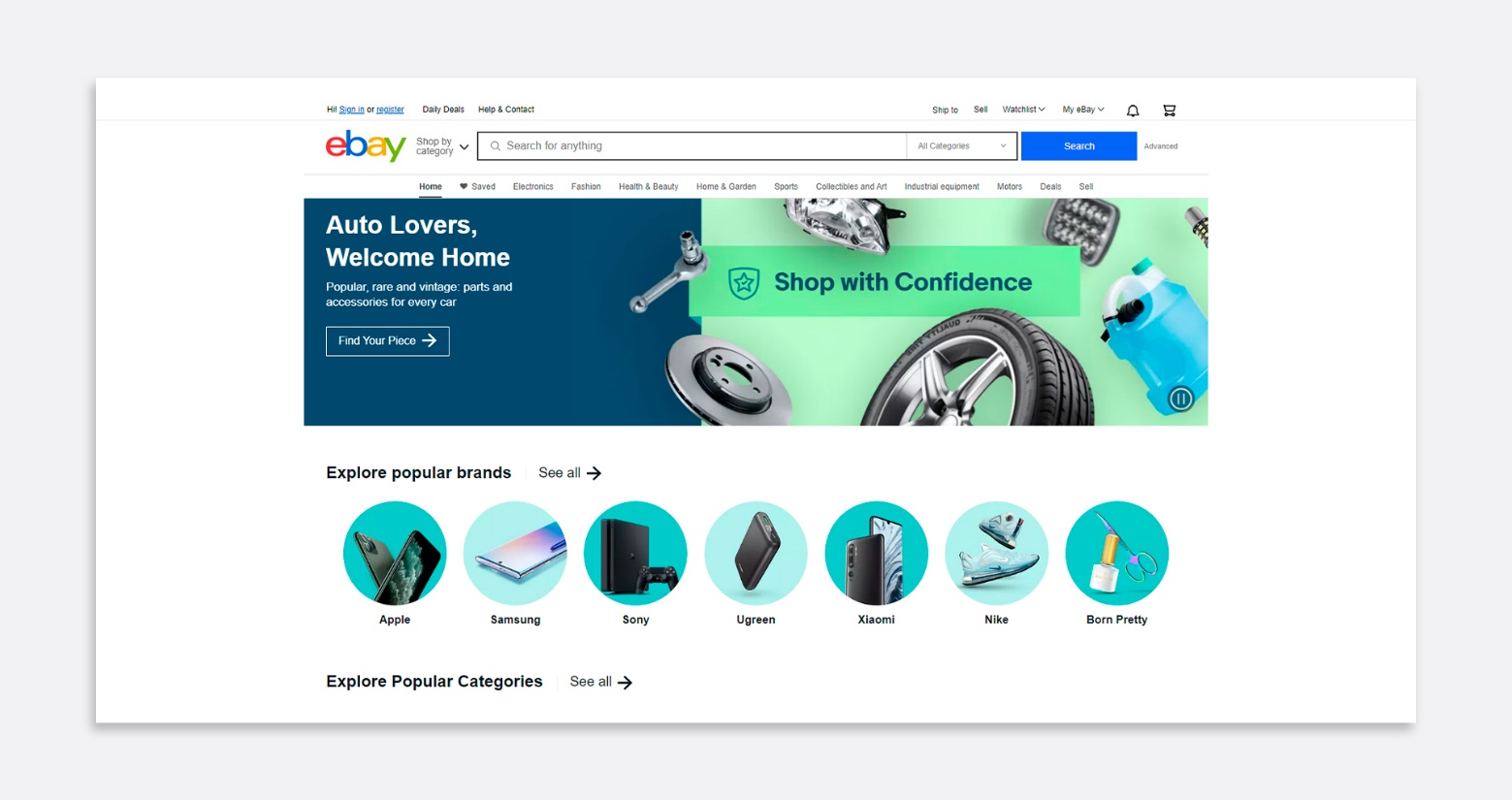
1. Соціальні мережі та месенджери. FB та його Messenger, Instagram, Twitter, Телеграм, Skype, Zoom та web-версії всього подібного їм — це вебдодатки.

[](https://brander.ua/assets/2022-02/v-chem-otlichie-veb-prilozhenij-ot-mob-prilozhenij-7.jpg)

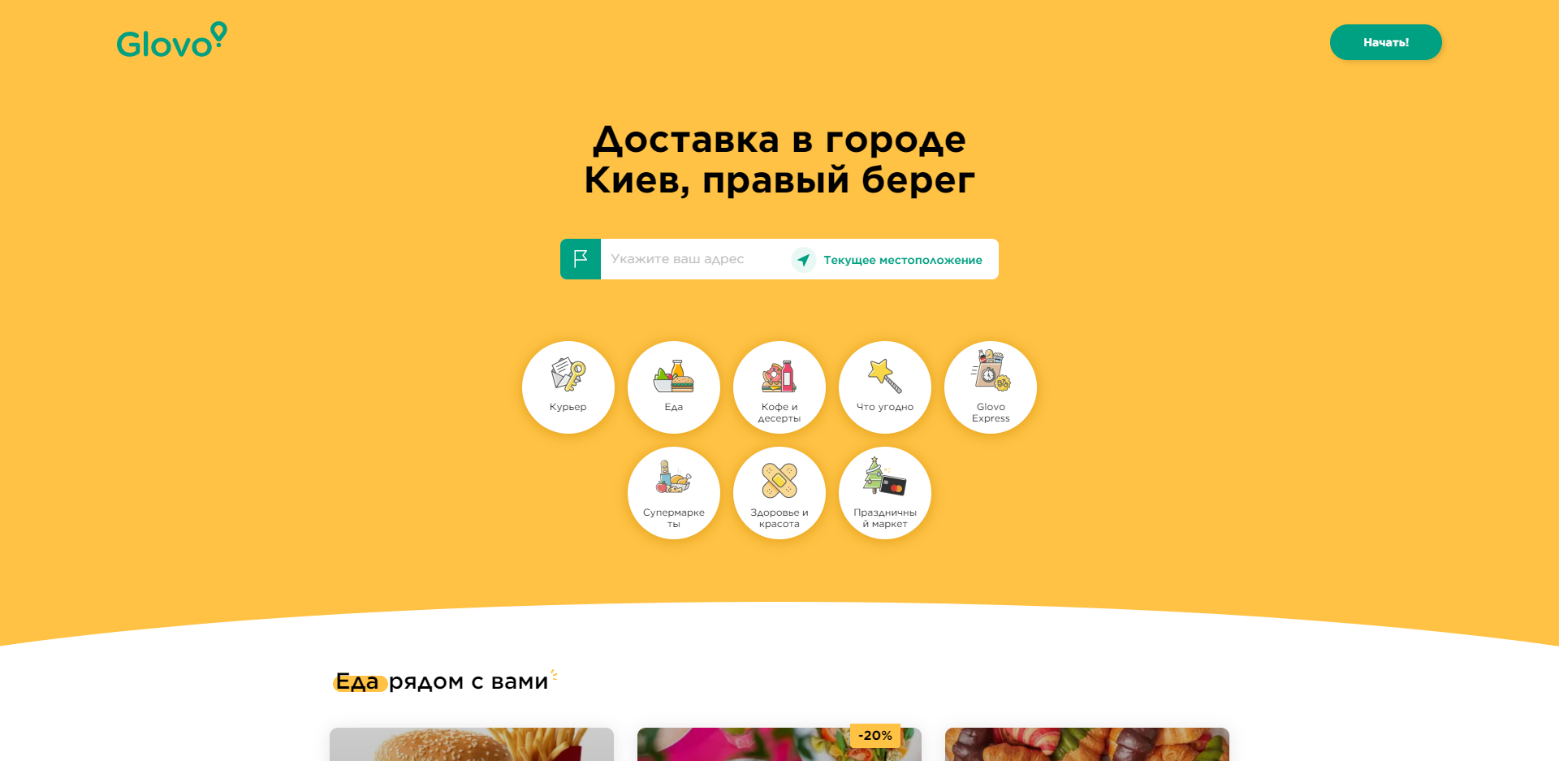
1. E-commerce. Усі маркетплейси як Amazon і Prom.ua, а також великі торгові майданчики — це теж вебдодатки.

[](https://brander.ua/assets/2022-02/v-chem-otlichie-veb-prilozhenij-ot-mob-prilozhenij-8.jpg)

1. Банки, біржі та аукціони. Сайти всіх фінансових установ, від e-Bay та трейдингових платформ, до МФО, також вебдодатки.

[](https://brander.ua/assets/2022-02/v-chem-otlichie-veb-prilozhenij-ot-mob-prilozhenij-9.jpg)

1. Усе інше. Кур’єрські служби, сервіси бронювання авіаквитків та житла, краудфандингові платформи тощо. Список можна продовжувати дуже довго, головне, щоби проєкт був складним, нестандартним та передбачав авторизацію.

[](https://brander.ua/assets/2022-02/v-chem-otlichie-veb-prilozhenij-ot-mob-prilozhenij-10.png)

## **Які бувають вебпрограми**

Є три види вебзастосунків: PWA, HTML-5 і SPA. Чим вони відрізняються, і яке саме рішення буде кращим для вашого бізнесу, розповімо трохи нижче.

### Еволюція сайтів до вебдодатків

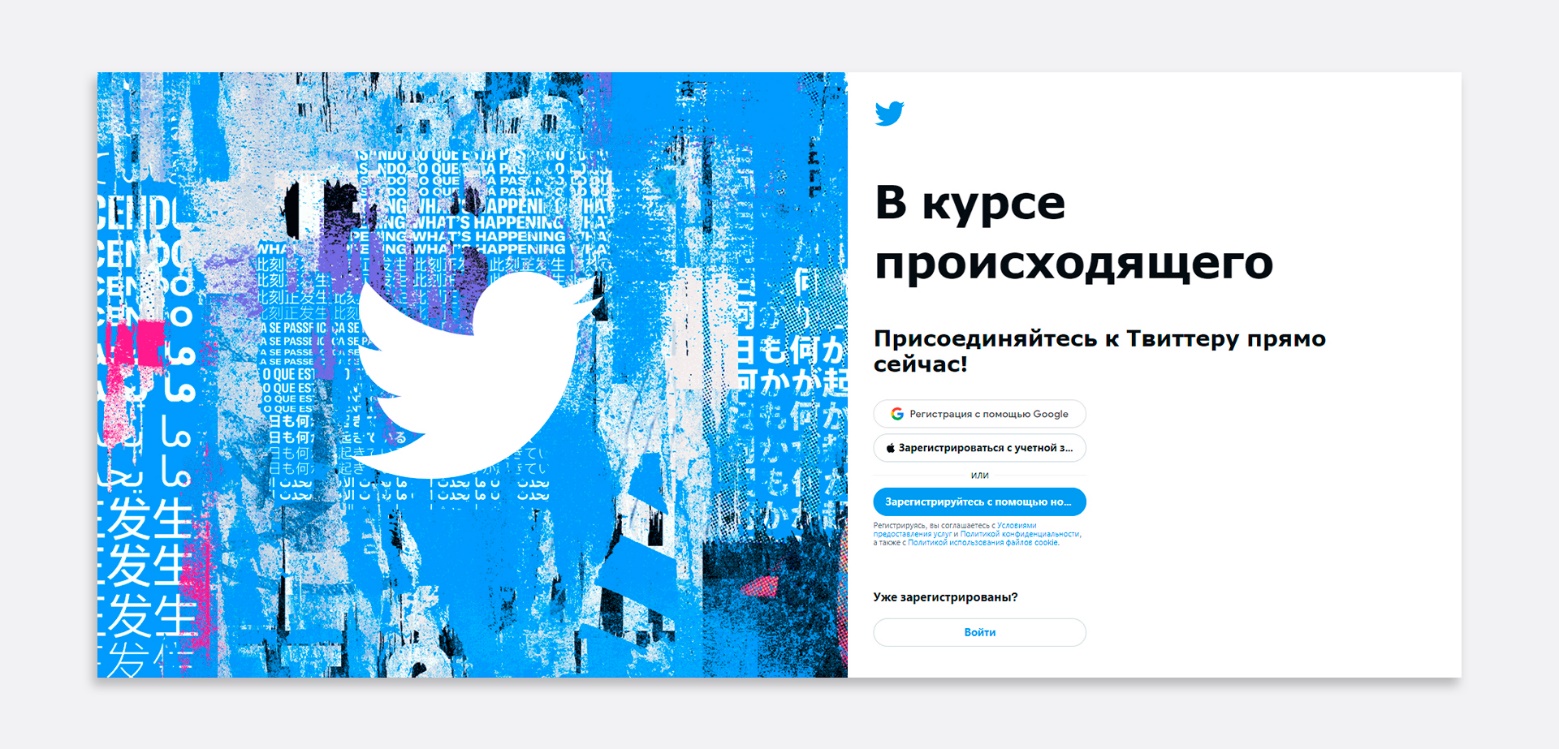
Спочатку трохи історії, щоби зрозуміти, як і навіщо сайти еволюціонували до вебдодатків:

1. Мобільна версія. Років 15 тому, за часів перших моделей айфонів та інших смартфонів, коли трава була зеленіша, а на мобільний трафік припадало не 55 % від усього світового обсягу, а менш як 4 %, почали створювати перші мобільні версії сайтів. Легкі, з меншою кількістю функцій та безліччю проблем вони вирішували найголовнішу — давали користувачам мобільних пристроїв доступ до сервісів. Зараз, коли в багатьох «телефони», потужніші за ПК, мобільні версії вже ніхто не робить, але це був початок.
2. Чуйний сайт. Потім з’явилися сайти з чуйним дизайном, які стискалися або розтягувалися, залежно від розміру вікна браузера. Тоді з’явився принцип mobile-first, згідно з яким сайт спочатку робився під перегляд із мобільних, і в другу чергу для браузера. Такі шаблонні рішення на основі різних CMS використовуються й зараз, наприклад, у маленьких сайтах-візитках та лендингах.
3. Адаптивна верстка. З 2011 всі «просто» сайти, що заслуговують на увагу, за замовчуванням створюються з адаптивним дизайном. У них є кілька варіантів верстки для різних пристроїв, які забезпечують хороший досвід користувача. Зараз це базове рішення для всіх проєктів із невеликим та середнім навантаженням. Тих самих корпоративних та новинних сайтів, а також середніх інтернет-магазинів.

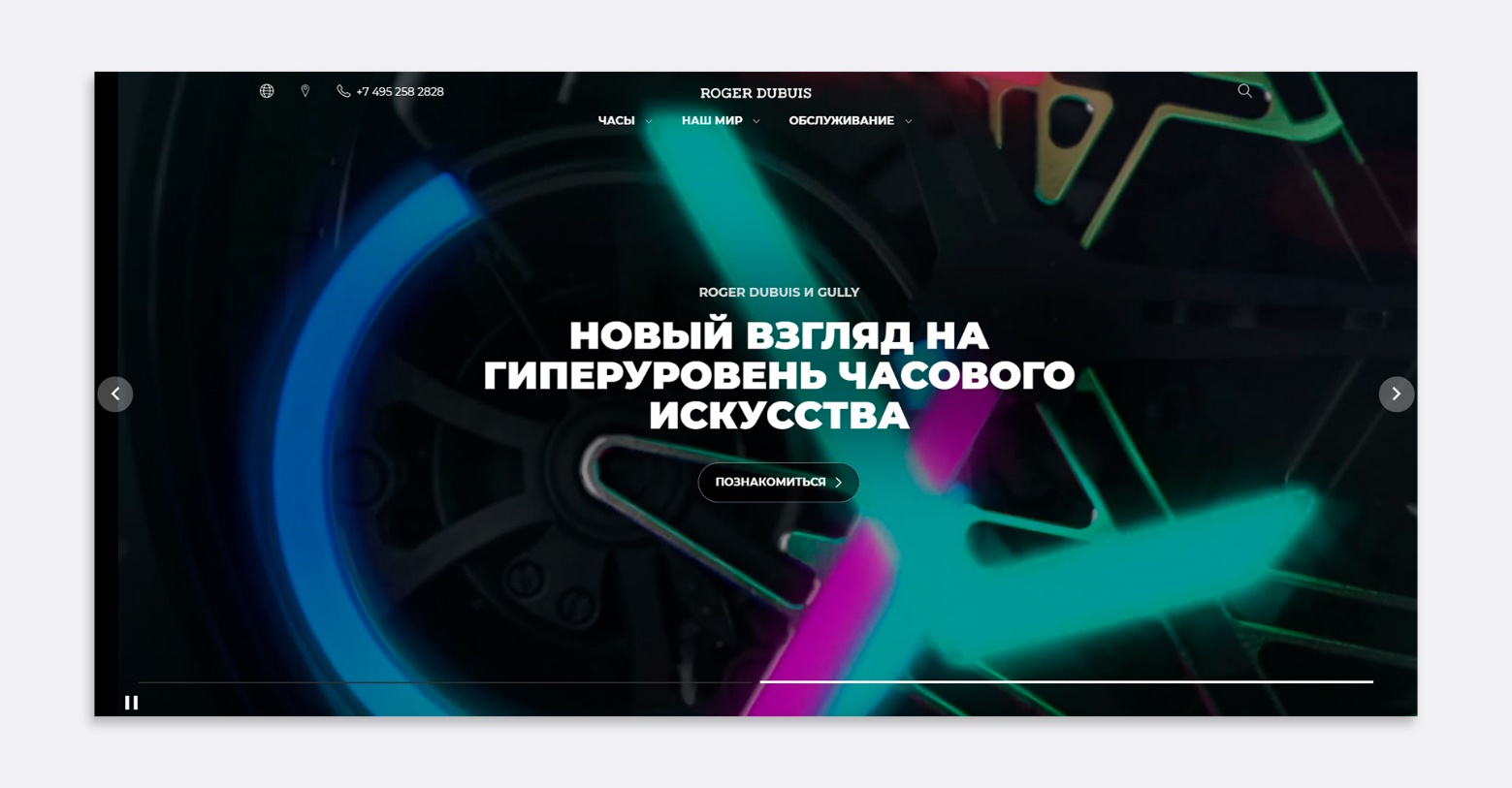
### PWA, HTML-5 і SPA вебдодатки

Сучасні вебпрограми відрізняються від усього, описаного вище й бувають трьох видів.

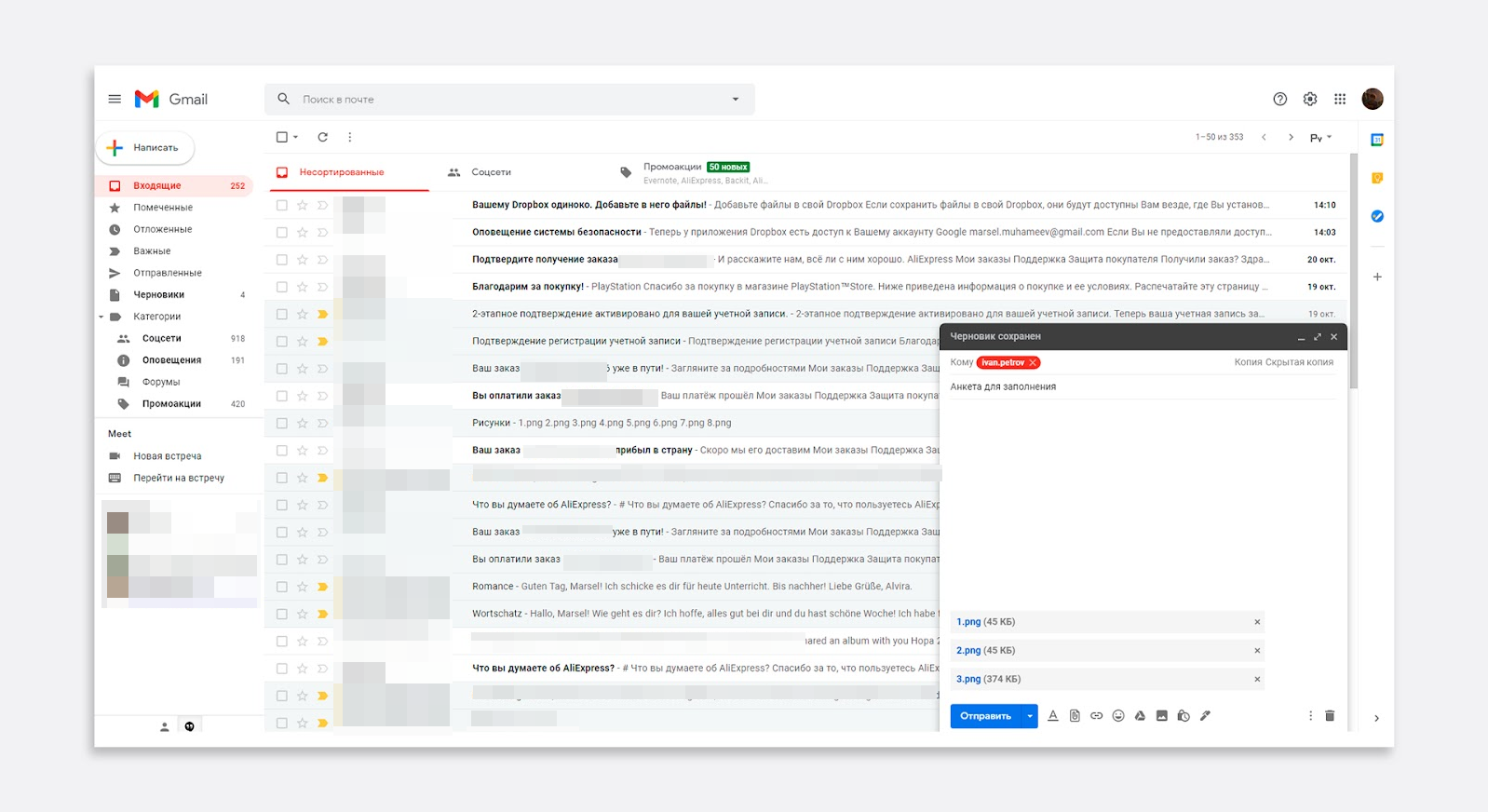
1. PWA. Прогресивний вебдодаток — майже як нативний мобільний додаток. Працює скрізь, але що новіша версія ОС, то краще, оскільки може поліпшуватися пропорційно технологіям. Зручний інтерфейс користувача, як у повноцінної програми. Одночасно індексується в пошукових мережах та дозволяє зберегти на екран іконку швидкого доступу. Вимагає підключення за захищеним протоколом HTTPS, але може частково працювати без інтернету.

[](https://brander.ua/assets/2022-02/v-chem-otlichie-veb-prilozhenij-ot-mob-prilozhenij-11.jpg)

1. HTML-5. Вебсторінка, що імітує роботу програми, також доступна в будь-яких браузерах. Не запускається без інтернету, не підходить для суперскладних проєктів, зате ідеальна, коли треба швидко створити якісний вебдодаток, котрий відповідає користувальницьким очікуванням.

[](https://brander.ua/assets/2022-02/v-chem-otlichie-veb-prilozhenij-ot-mob-prilozhenij-12.jpg)

1. SPA. Single Page Application — односторінковий вебдодаток із динамічним оновленням. Статична основа залишається незмінною, змінюються ті дані, із якими взаємодіє користувач. SPA-технологію використовують, як для адаптивних сайтів, так і для PWA вебзастосунків.

[](https://brander.ua/assets/2022-02/v-chem-otlichie-veb-prilozhenij-ot-mob-prilozhenij-13.png)

## **Що таке мобільні програми**

З мобільними програмами все набагато простіше. Вони — те саме, що комп’ютерні програми, тільки для смартфонів та планшетів. «Хакери 80 рівня», а також фанати китайських або кастомних ОС навіть можуть завантажити їх не тільки з фірмових магазинів Apple і Android, а з будь-якого вільного джерела. Є безліч технологій, за допомогою яких можна створити мобільний додаток, але вони діляться на дві великі категорії:

1. Нативні. Програми написані рідними мовами мобільної платформи — Java та Kotlin для Android та Swift та objective-C для iOS/iPadOS. Легко отримують доступ до технічної частини вашого смартфона, наприклад, мікрофону та камери. Гарантують хороший, звичний досвід користувача.
2. Гібридні. Універсальні програми, не прив’язані до платформи, у розробці використовуються одночасно нативні та вебтехнології. Дозволяють заощадити й не платити за розробку двох версій для iOS та Android, але водночас не підходять для дуже складних проєктів і забезпечують не такий гарний досвід користувача.

Хоча цикл розробки у веб-додатків та мобільних дуже схожий, а підсумкові проєкти можуть бути зовні майже ідентичними, створенням програм займаються абсолютно інші програмісти, які володіють іншими мовами. Щобільше, усі інші фахівці, задіяні в розробці, від дизайнерів, до техпідтримки, і навіть маркетологів, теж будуть вирішувати свої завдання іншими способами, не такими, як для вебдодатку.

## **Що вибрати вебдодаток або мобільний**

Ще раз про головне: вебдодаток нічим не кращий за мобільний і навпаки. Це просто різні рішення для різних цілей і, найголовніше, дуже часто великим проєктам потрібно й те, й інше.

Щоби зрозуміти, що потрібно саме вашому проєкту, проводиться аналітика. Як і при ухваленні будь-яких інших бізнес-рішень, потрібно насамперед орієнтуватися на потреби потенційних користувачів. А ці потреби серйозно різняться залежно від напрямів бізнесу та особливостей ЦА.

[](https://brander.ua/assets/2022-02/v-chem-otlichie-veb-prilozhenij-ot-mob-prilozhenij-14.jpg)

Крім даних про особливості своїх споживачів, рекомендуємо врахувати ще три моменти:

1. Технічні можливості смартфона. Критичний фактор вибору між мобільним та вебдодатком. PWA вебпрограми можуть надсилати push-сповіщення та визначати геолокацію, якщо користувач дозволить. Якщо ваш проєкт потребує доступу не тільки до геопозиціювання, але також до камери смартфона, розпізнавання обличчя або відбитка пальця, Bluetooth, NFC і так далі, значить, вам потрібна мобільна програма. Без варіантів.
2. Розкрутка. Вебдодаток можна й потрібно просувати за всіма законами SEO. Але й запустити рекламу ніхто не заважає. З програмами з магазинів усе складніше. Потрібно працювати з їх ранжуванням у магазині та намагатися просувати в пошукових мережах. Програми в будь-якому випадку обов’язково залучать новий трафік на основний сайт, якщо він є. І навпаки. А ось, якщо сайту або вебдодатку немає, то просування мобільного додатка обійдеться вам дорожче й займе більше часу.
3. Ціна та терміни. Мобільний додаток набагато складніше розробити, ніж його вебверсію. Особливо, якщо ви створюватимете дві версії. Розробка складного вебдодатку — теж нешвидкий і непростий процес, але в середньому випуск вебверсії вимагатиме менше часу та інвестицій.

## **Висновок**

[Звичайні сайти](https://brander.ua/what-we-offer/web-development) з адаптивним дизайном та HTML-5 вебдодатки чудово справляються з усіма базовими та просунутими завданнями e-commerce. Якщо у вас є більш складна бізнес-концепція, вам може підійти PWA вебдодаток. Головна його відмінність від мобільного — воно працюватиме в браузері. І замість того, щоби завантажувати програму в магазині, користувачам потрібно буде зареєструватися. Якщо вам потрібен доступ до технічних компонентів смартфона, наприклад, камери, NFC, FaceID тощо — вам точно потрібна нативна програма. Гібридні програми варто вибирати з метою економії й тільки якщо вам точно потрібно саме мобільний, а не вебдодаток або як доповнення до основного сайту. У будь-якому випадку приймати рішення щодо свого проєкту потрібно лише після всебічної аналітики.