# 2.2. Базові технології фронтенду та бекенду

## Сучасний Веб - це постійно зростаюча кількість сторінок і веб-додатків, що пов'язані між собою посиланнями. Він сповнений відеороликів, фотографій і інтерактивного контенту. Однак, взаємодія веб-технологій, завдяки яким все це так злагоджено працює, залишається прихованою від очей звичайного користувача.

Веб-технології стрімко розвиваються і розробники мають широкі можливості створювати веб-вміст нового покоління. Сьогоднішній Інтернет є результатом безперервних зусиль відкритої веб-спільноти, яка розробляє новітні технології і домагається їх підтримки всіма браузерами.

Будь-який сучасний сайт чи додаток має дві частини, що тісно взаємодіють між собою. Ці частини зазвичай вимагають окремої розробки. Розробляти клієнтську та серверну частину може як один розробник, так і велика команда фахівців. Це залежить від складності веб-проекту та бюджету.

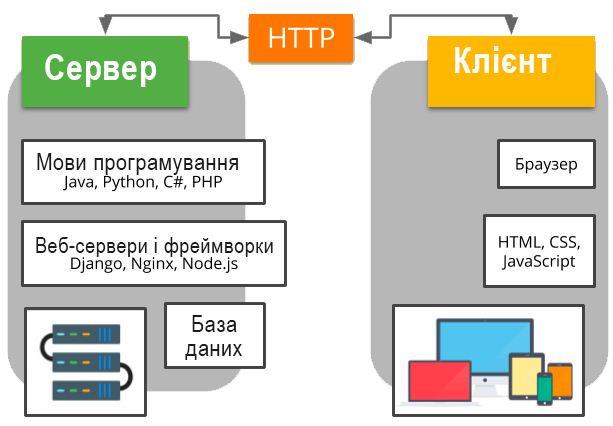


Рис.5. Взаємодія серверної та клієнтської частин веб-програми

### Фронтенд розробка

Для написання клієнтського коду зазвичай використовують комбінацію HTML (відповідає за структурування змісту сторінки), CSS (надає веб-сторінці певного вигляду), JavaScript (клієнтська мова програмування). Браузер розуміє код цих мов, тому, для відтворення сайту (додатку) не потрібно жодних налаштувань.

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

### Серверна частина (Backend)

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

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).

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.

## Базові інструменти фронтенд-розробки

### HTML – мова розмітки тексту

Мова гіпертекстової розмітки HTML (HyperText Markup Language) є основним будівельним засобом для веб-сторінок, використовується для створення та візуального представлення веб-сторінок. Визначає структуру і описує зміст веб-сторінки в структурованій формі.

HTML-сторінка є звичайним текстовим документом, в якому використано спеціальні оператори – **теги** (*tag*) чи інша назва **дескриптори** (*descriptor*) для позначення, в якому вигляді буде виведено текстовий чи інший елемент у вікні браузера. Прикладами таких операторів є <img>, <title>, <p>, <div>, <table> тощо.

HTML дозволяє формувати на сторінці сайту текстові блоки, додавати до них зображення, організовувати таблиці, додавати до дизайну сайту звуковий супровід, організовувати гіперпосилання з переходом до інших розділів сайту або ресурсів Інтернету і компонувати всі ці елементи між собою. За допомогою HTML можна створити як статичний так і динамічний сайт. Сторінки, які створено лише засобами HTML мають розширення .html.

**Гіперпосилання** (Hyperlink) — це базовий функціональний елемент HTML-документу, який реалізовує зв'язок певного об'єкту веб-сторінки з іншим об’єктом. Для гіперпосилання може використовуватися як фрагмент тексту, так і графічний об'єкт, а сам зв'язок можна встановлювати як між об’єктами одного сайту, так і між об'єктами, що розміщені на різних сайтах Інтернету.

HTML є мовою, що лише інтерпретується, тому, для виконання коду, його не потрібно компілювати. Інтерпретатор мови втілено в браузер і він «компілює» код безпосередньо під час відкривання документа. Якщо в коді сторінки виявлено помилку, інтерпретатор, зазвичай, не видає відповідного попередження, а просто ігнорує помилку, що може зіпсувати зовнішній вигляд завантаженої сторінки. Для запобігання цього розробникам слід бути уважними під час складання HTML-коду і ретельно тестувати результати своєї роботи.

На сьогодні актуальною є версія HTML5/

Елементи, що знаходяться всередині елементу <html>, утворюють дерево документа, так звану об'єктну модель документа DOM (Document Object Model).

### DOM (Document Object Model)

Об'єктна модель документа – це модель документа як об'єкта, створюється веб-браузером у пам'яті пристрою на підставі коду HTML, що отримано від сервера.

1. Сервер створює HTML код сторінки або віддає його (якщо це простий статичний сайт).
2. Браузер отримує код, аналізує його та розбирає на елементи дерева.
3. За необхідності підключається JavaScript, якщо він використовується, щоб змінити поведінку тегів та їх вмісту залежно від впливу користувача.
4. DOM-дерево відображається у вкладці браузера у вигляді, який задуманий розробниками.

HTML-код, який пишуть програмісти – це лише текстовий файл певного формату, а DOM – результат дій браузера, який створює об'єкти при розборі текстових файлів.

Завдяки W3-консорціуму вироблено єдиний стандарт побудови та аналізу вмісту веб-сторінки. До цього різні браузери діяли по-своєму, що створювало багато незручностей для розробників. Ознайомитись з актуальним DOM-стандартом можна на офіційному сайті.

DOM представляє документ у вигляді дерева, де кожен елемент сторінки представлений як вузол дерева. Ці вузли поєднуються в ієрархічну структуру елементів веб-сторінки, між якими задіяні "родинні стосунки". Відносини між множинними вкладеними елементами підрозділяються на батьківські, дочірні та сіблінгові (від англ. sibling - діти, що мають спільних батьків).

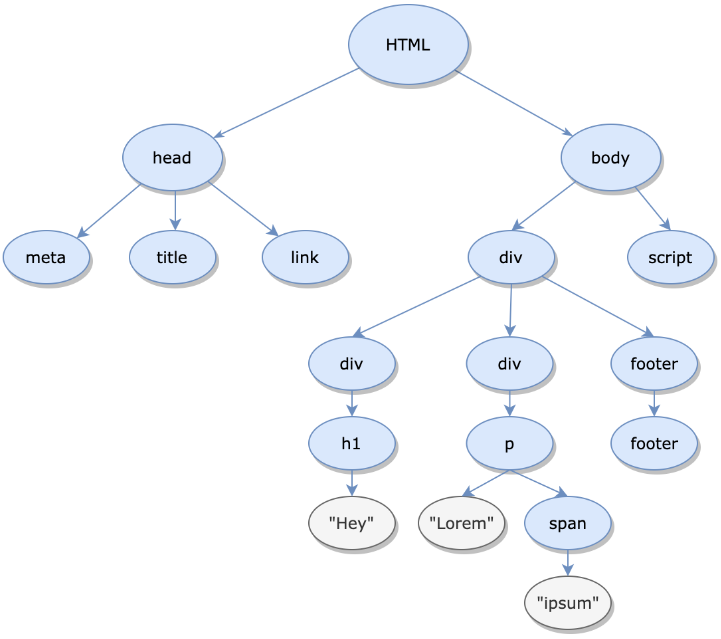


Рис. 2. Найпростіша DOM структура веб-сторінки

Основні структури, які використовуються в об'єктній моделі документа:

* **Дерево.** Веб-сторінка представляється як ієрархічне перевернуте дерево, що починається з головного елемента <html> і розширюється до низу. У кожного об'єкта дерева є батько і нащадки, якщо він знаходиться не в самому низу.
* **Батьківський елемент** - елемент, що пов'язаний з іншими елементами нижчого рівня і знаходиться на дереві вище за них. На рис.2 <html> є батьківським лише для <head> і <body>. Елемент <body> є батьківським для всіх елементів, що містяться в ньому: <h1>, <p>, <span>, <nav> і т.д. Тег <p> є батьківським лише для <span>.
* **Дочірній елемент** - елемент, що розташований всередині іншого елементу. На рис.2 елементи <h1>, <h2>, <p> і <nav> є дочірніми по відношенню до <body>.
* **Сіблінговий елемент** - елемент, що має спільний батьківський елемент з даним, це елементи одного рівня. На рис.2 <head> і <body> є елементами одного рівня, так само як і елементи <h1>, <h2> і <p> є між собою сіблінговими.

Модель DOM є потужним інструментом для створення інтерактивних та динамічних веб-сторінок. Він дозволяє розробникам створювати складні користувацькі інтерфейси, керувати даними і забезпечувати взаємодію користувача з веб-сайтом. Однією з найпоширеніших мов програмування, які використовуються для роботи з DOM у веб-розробці є JavaScript.

DOM надає багато способів доступу до елементів веб-сторінки. Можна отримати доступ до елементів за їх тегом, класом, ідентифікатором або іншими атрибутами. За допомогою DOM можна змінювати вміст та атрибути елементів веб-сторінки. Це дозволяє динамічно оновлювати вміст сторінки без перезавантаження. DOM підтримує обробку подій, таких як клацання мишею, натискання клавіш та інші дії користувача.

### CSS – каскадна таблиця стилів

CSS (Cascading Style Sheets) — це технологія опису зовнішнього вигляду документа, що створено засобами HTML.

CSS використовується для привласнення певних особливостей для елементів HTML-сторінки: колір, шрифт, розташування на сторінці тощо. До появи CSS оформлення елементів вказувалося безпосередньо в HTML-коді сторінки. Проте, з появою CSS стало можливим принципове розділення структури і опису документа. За рахунок такого розподілу стало можливим легке застосування єдиного стилю оформлення для кількох сторінок сайту, а також швидка зміна цього оформлення.

#### Переваги CSS:

* Застосування кількох варіантів дизайну сторінки для перегляду на різних пристроїв, наприклад, для комп’ютера, планшету чи телефону.
* Зменшення часу завантаження сторінок сайту за рахунок перенесення правил опису даних до окремого CSS-файлу. В цьому випадку браузер завантажує лише структуру документа і дані, що містяться в HTML коді. CSS-файл з правилами опису цих даних завантажується браузером лише один раз і зберігається в кеші браузера.
* Простота подальшої зміни дизайну. Не потрібно виправляти кожну сторінку, достатньо лише змінити кілька правил у CSS-файлі.
* Додаткові можливості оформлення. Наприклад, за допомогою CSS-правил можна застосувати обтікання певного блоку текстом або зробити так, щоб меню фіксовано знаходилося в певному місці при перегортанні сторінки.

На сьогодні актуальною є версія CSS3.

### JavaScript – мова сценаріїв

JavaScript – це фрагменти програмного коду (скрипти), що надають динаміки для певних елементів сторінки. Програмний код на JavaScript додається до HTML-сторінки і інтерпретується браузером в міру завантаження цього документа. За допомогою JavaScript можна динамічно реагувати на події, які пов'язані з діями відвідувача або змінами стану сторінки чи вікна.

Важливою особливістю JavaScript є об'єктна орієнтованість. Програмісту є доступними численні об'єкти, такі, як документи, гіперпосилання, форми, фрейми тощо. Об'єкти характеризуються описовою інформацією (властивостями) і можливими діями (методами).

## Бекенд розробка. Програмування серверної частини

Веб-додатки і більшість веб-сайтів використовують програмування серверної частини, щоб за потребою динамічно відображати різні дані, які в основному витягуються з бази даних.

**Бекенд веб-сайту** — це серверна частина сайту, прихована від очей користувача. Це закладання певної логіки та написання серверних сценаріїв, що керують функціями і процесами сайту. Бекенд розробляється з використанням іншого стеку технологій, ніж у фронтенді.

Коди, задіяні у серверній частині та клієнтській частині значно відрізняються:

* Вони мають різні цілі та призначення.
* Як правило, вони не використовують однакові мови програмування. Виняток складає JavaScript, який можна використовувати на стороні сервера та клієнта.
* Вони виконуються у різних середовищах операційної системи.

Код, який виконується в браузері, відомий як код клієнтської частини, перш за все пов'язаний з зовнішнім виглядом і поведінкою веб-сторінки. Це включає вибір і стилізацію компонентів інтерфейсу користувача, створення макетів, навігацію, перевірку форм тощо.

Код клієнтської частини написаний з використанням HTML, CSS та JavaScript. Він запускається у браузері і практично не має доступу до базової операційної системи (включаючи обмежений доступ до файлової системи).

Програмування на стороні сервера в основному включає вибір вмісту, який повертається браузеру у відповідь на запити. Код на стороні сервера обробляє такі завдання, як перевірка надісланих даних та запитів, використання баз даних для зберігання та витягування даних та надсилання правильних даних до клієнта.

Код серверної частини може бути написаний на будь-якій мові програмування (PHP, Python, Ruby, C# і NodeJS). Код серверної частини має повний доступ до операційної системи сервера.

### Дії у серверній частині

Програмування серверної частини дозволяє ефективно доставляти актуальну інформацію, складену для індивідуальних користувачів.

### Ефективне зберігання та доставка інформації

Програмування серверної частини дозволяє зберігати інформацію в базі даних і динамічно створювати та повертати HTML чи інші типи файлів (наприклад, PDF, зображення тощо).

Сервер не обмежений у надсиланні інформації з баз даних і повертає результат виконання певних сценаріїв. Контент може бути цільовим щодо пристрою клієнта, який його отримує.

### Налаштований контент для користувача

Сервери можуть зберігати та використовувати інформацію про клієнтів, щоб постачати індивідуальний контент. Глибокий аналіз звичок користувача може бути використаний для прогнозування їхніх інтересів та подальших налаштувань відповідей та повідомлень, наприклад, надання списку раніше відвіданих популярних місць на карті.

### Контрольований доступ до контенту

Авторизація - центральна частина взаємодії користувача з веб-додатком. Програмування серверної частини дозволяє сайтам обмежувати доступ не авторизованим користувачам та надавати лише ту інформацію, яку користувачеві дозволено бачити.

Реальні приклади:

* Соціальні мережі, такі як Facebook, дозволяють користувачам повністю контролювати свої дані, але друзям дозволяти переглядати чи коментувати. Користувач визначає, хто може переглядати його дані та чиї дані з'являються на його стіні.
* Блог контролює доступ до контенту: статті видно всім, але тільки авторизовані користувачі можуть коментувати чи редагувати контент.
* На сайті онлайн-банкінгу доступ до контенту контролюється. Без авторизації на ньому практично нічого не можна зробити, окрім перегляду загальної та довідкової інформації. Після авторизації стають доступними всі дозволені фінансові операції.

### Зберігання інформації про сесію/стан

Програмування серверної частини дозволяє розробникам використовувати сесії. Це механізм, що дозволяє серверу зберігати інформацію про поточного користувача сайту та надсилати різні відповіді на основі цієї інформації.

Наприклад, якщо користувач був попередньо авторизований, то виводити історію замовлень або зберегти прогрес простої гри, так щоб користувач міг повернутися на сайт продовжити з того місця, де він закінчив.

### Повідомлення та засоби зв'язку

Через програмування серверної частини можуть надсилатися повідомлення електронною поштою, смс, миттєві повідомлення, відеозв'язок або інші засоби зв'язку.

* Facebook або Twitter надсилає повідомлення електронною поштою та смс-повідомлення, щоб повідомити про нові розмови.
* Amazon регулярно надсилає листи на електронну пошту, що пропонують товари, схожі на ті, які вже були куплені або переглядалися.
* Веб-сервер може надсилати повідомлення адміністратору сайту, попереджаючи його про те, що на сервері закінчується пам'ять або про підозрілу активність користувача.
* Найпоширеніший вид повідомлень – це підтвердження реєстрації. Якщо створити новий обліковий запис на великому порталі використовуючи адресу електронної пошти, то на пошту надходить лист, який підтверджує факт реєстрації або містить інформацію про необхідність активувати обліковий запис.

### Аналіз даних

Веб-додаток може збирати багато даних про своїх користувачів: що вони шукають, що купують, що рекомендують, як довго залишаються на кожній сторінці. Програмування серверної частини може бути використане, щоб вдосконалити відповіді, що базуються на аналізі цих даних.

Наприклад, Google рекламує товари на підставі попередніх пошуків. Facebook формує стрічку новин і рекламу, орієнтуючись на користувача чи його друзів. Rozetka буде пам’ятати список переглянутих, вподобаних і куплених товарів.

# Сучасні засоби веб-технологій

Сучасні розробники зазвичай пишуть свій код, використовуючи бібліотеки та веб-фреймворки, які для розробки клієнтської і серверної частин є різними.

**Бібліотеки** — це набори попередньо написаних фрагментів коду, які можна легко інтегрувати до існуючого коду проекту. Таким чином, бібліотека є спеціалізованим інструментом для конкретних вузьких потреб, а не універсальною машиною для підготовки всього проекту.

Бібліотека, зазвичай, скорочує час розробки приблизно на 20%, дозволяючи не турбуватися про дрібниці, втім, слід врахувати певні особливості:

* Баг в реалізації бібліотеки може викликати складності в його знаходженні та способи усунення.
* Немає гарантії, що розробники бібліотеки оперативно випустять оновлення або виправлять недоліки.
* Оновлення може змінити доступ до API бібліотеки, що може спричинити значні зміни у коді.

Більшість популярних бібліотек поширюються як Open Source. Популярних комерційних бібліотек досить мало, крім складних вузькоспеціалізованих напрямів.

**Фреймворки** є шаблонами для створення веб-сайту або веб-додатку. Вони забезпечують структуру, де можна розмістити весь проект. Шаблони фреймворку створюють структуру з певними виділеними областями для вбудовування коду. Отже, фреймворки JavaScript - це повні набори інструментів для формування та налаштування веб-сайту або додатків.

Фреймворк знаходиться на більш високому рівні абстракції у порівнянні з бібліотекою і дозволить без зайвих зусиль розробляти близько 80% застосування. Але слід врахувати:

* Останні 20% можуть викликати чималі труднощі через обмеження, що накладаються фреймворком.
* Оновлення фреймворку може не мати підтримки попередніх версій.
* Основний код і концепції рідко задовольняють розробників в своєму первісному вигляді. Вони завжди знайдуть «кращий» спосіб зробити що-небудь.

### Інструменти

Інструмент - це допоміжний засіб розробки, але він не є невід'ємною частиною проекту. Інструменти містять системи збирання, компілятори, транслятори, механізми розгортання, тестування та інше.

Інструменти спрощують процес розробки. Наприклад, використання препроцесору Sass замість чистого CSS, оскільки він надає можливість використовувати цикли, функції, локальні змінні і багато іншого. Браузери не розуміють Sass/SCSS синтаксис, тому код переводиться в CSS.

### Open Source

Open Source – це програмне забезпечення, яке постачається для кінцевого користувача з відкритим вихідним кодом. Тобто програму можна доопрацювати під власні завдання без порушення авторських прав розробників вихідного ПЗ. Рішення розповсюджується під ліцензіями GNU/Linux, MIT та інші.

Саме поняття виникло як альтернатива пропрієтарного ПЗ, коли комерційні компанії закривали доступ до вихідного коду.

Програмне забезпечення, яке розповсюджується як Open Source продукт, має ряд особливостей:

* Програмні продукти постачаються абсолютно безкоштовно до кінцевого споживача.
* Розробляються різними фахівцями, які уважно відстежують код програми, оперативно усуваючи помилки чи вразливості, що можуть спричинити критичні наслідки.
* Більшість продуктів Open Source сумісні з різними сімействами операційних систем, тобто вони кросплатформні.
* Спільнота розробників вільного програмного забезпечення відкрита для зворотного зв'язку від користувачів. Кожен з них може зробити пропозицію щодо покращення або додавання нової функції.
* Оновлення Open Source рішень відбувається значно частіше, ніж комерційні продукти. Кінцевий користувач отримує актуальні виправлення миттєво, як тільки вони фіксуються.
* Активний розвиток вільного ПЗ пожвавлює конкуренцію серед комерційних організацій, що сприяє підвищенню якості програм.
* Як комерційну основу розробники використовують у своїх рішеннях опцію добровільних пожертв. Користувачі можуть за власним бажанням підтримати програмістів фінансовими коштами шляхом переведення на електронний гаманець.
* Більшість Open Source рішень створюються під операційні системи сімейства Unix/Linux, що зменшує ризик зараження комп'ютера чи сервера вірусом. Шкідливе ПЗ переважно пишеться під Windows.

Для розміщення відкритих проектів з контролем версій популярною є онлайн платформа Github з використанням Git. Git — розподілена система контроля версій, яка надає можливість розробникам відслідковувати зміни в файлах и колективно працювати над одним проектом. Хоча Github більше відомий як платформа для розробки open source проектів, ресурс також надає можливість використання приватних репозиторіїв.

### CSS-бібліотека Bootstrap

Bootstrap є найпопулярнішою бібліотекою, що вільно поширюється через офіційний сайт getbootstrap.com. Він містить готові стилі і скрипти, для застосування яких достатньо прописати необхідні класи і атрибути html-елементів. За допомогою Bootstrap-сітки легко адаптувати будь-який сайт і добре відображати його на будь-яких пристроях.

* Файли Bootstrap з готовим написаним кодом HTML I CSS під’єднуються до сайту в елементі head, після чого стають доступними їх можливості.
* Bootstrap ідеально підходить при роботі в команді. Верстка на Bootstrap при належному розумінні відбувається в 3-5 разів швидше, а однаковість коду дозволить іншому розробнику вносити правки.
* В Bootstrap закладено багато компонентів, все, що може знадобитися при розробці типових сайтів, наприклад, випадне меню, кнопки, таби, індикатори стану, хлібні крихти, списки, заголовки тощо. Присутній іконковий шрифт, вставляти іконки на веб-сторінки просто, опис є в офіційній документації.
* Bootstrap має велику спільноту прихильників і наявність хорошої документації. Завдяки такій поширеності Bootstrap з'явилося багато шаблонів, де вже перероблено дизайн всіх основних елементів. На основі таких шаблонів можна робити сайти, лише незначно щось змінюючи.

### JS-фреймворки

JS-фреймворки - це інструменти для побудови динамічних додатків на Javascript. Розробники використовують JS-фреймворки там, де неможливо/складно/довго виконувати завдання звичайними засобами. У переважній більшості випадків, фреймворки використовуються для написання односторінкових додатків.

Односторінковий додаток (Single Page Application, SPA) - це веб-додаток чи веб-сайт, в якому необхідний код - HTML, JavaScript, та CSS - завантажується разом із сторінкою або динамічно довантажується за потребою, зазвичай, у відповідь на дії користувача. Сторінка не оновлюється і не перескеровує користувача на іншу сторінку в процесі роботи з нею. Взаємодія з односторінковим додатком часто здійснюється через динамічний зв'язок з веб-сервером.

#### Переваги побудови програми на JS-фреймворку:

* Ефективність. Проекти, які раніше розроблялися місяцями і мали сотні рядків коду, зараз можуть бути реалізовані значно швидше з добре структурованими готовими шаблонами і функціями. Коду стає помітно менше і він чистіше, що позитивно відбивається на швидкості розробки, а також підтримки та усунення помилок в коді програми.
* Безпека. Кращі JavaScript фреймворки мають фірмову систему безпеки і підтримуються великою спільнотою.
* Витрати. Більшість фреймворків є з відкритим кодом і безкоштовними. Оскільки вони допомагають програмістам швидше розробляти власні рішення, підсумкова ціна веб додатку буде нижчою. Наявність структури передбачає модульність додатку, а це надає можливість простіше працювати над додатком кільком розробникам одночасно.
* Можливість швидко створити мобільний або настільний кросплатформний додаток з веб-версії.

Додатків на js-фреймворках створено багато і цей сегмент швидко розвивається.

### Популярні JavaScript фреймворки та бібліотеки

* **Angular** – це JavaScript-фреймворк від Google, сумісний з більшістю поширених редакторів коду. Angular призначений для створення динамічних односторінкових веб-додатків (SPA) та прогресивних веб-додатків (PWA). Angular є одним із найпопулярніших фронтенд-фреймворків.
* **Vue.js** —фреймворк із відкритим вихідним кодом для односторінкових додатків. Він використовує модель розробки на основі компонентів та дозволяє приєднувати компоненти до проекту. Vue.js — приклад бібліотеки, більше схожої на фреймворк. Він пропонує багато шаблонів та патернів, які застосовуються при розробці.
* **Ember.js** – це фреймворк для розробки односторінкових, мобільних та десктопних додатків. Інструменти Ember дозволяють проектувати середовище розробки, а його командний рядок надає інструменти для автоматизації сценаріїв.
* **React** — бібліотека з відкритим вихідним кодом для створення динамічних інтерфейсів, що розроблена Facebook. Застосовується для створення веб-додатків з множинними динамічними компонентами. React заснований на JavaScript та JSX і дозволяє створювати HTML-елементи для багаторазового використання. React також включає React Native, спеціальне кросплатформве середовище для розробки мобільних додатків.
* **Node.js** – це серверна платформа з відкритим вихідним кодом, що створена на основі Google Chrome JavaScript Engine. Це одне з найбільш завантажених кросплатформових середовищ для виконання коду JavaScript. Node.js - це асинхронна, однопоточна, неблокуюча модель введення/виводу, яка робить її легкою та ефективною.
* **jQuery** призначений для управління HTML-документами. Бібліотека має простий API для управління подіями та розробки анімації у браузерах. jQuery застосовується для управління об'єктною моделлю документа (DOM), а також є інструментом розробки плагінів. Вона також поставляється з легшою крос-браузерною бібліотекою jQuery UI для побудови графічного інтерфейсу.

## Контрольні питання

1. Для чого призначена мова розмітки HTML?
2. Перелічити переваги, що надає використання стилів CSS
3. Які функції покладено на мову клієнтського програмування JavaScript?
4. Для чого використовують мови серверного програмування?
5. Перелічити переваги від використання мови HTML5.
6. Які функції покладено на різні рівні специфікацій CSS?
7. Який набір функціоналу може містити бібліотека?
8. Що собою представляє JS-фреймворк?
9. Перелічити популярні JS-фреймворки.
10. Назвати переваги від використання бібліотеки jQuery?

## Використані джерела

1. Огляд мови розмітки HTML для початківців <https://timeweb.com/ru/community/articles/chto-takoe-html>
2. HTML-довідник [https://schoolsw3.com/html/](https://schoolsw3.com/html/index.php)
3. CSS-довідник [https://schoolsw3.com/css/](https://schoolsw3.com/css/index.php)
4. JS-довідник [https://schoolsw3.com/js/](https://schoolsw3.com/js/index.php)
5. HTML Book <https://html5book.ru/html-html5/>
6. Керівництво по оформленню коду HTML/CSS <https://proglib.io/p/obuchenie-veb-razrabotke-rukovodstvo-po-oformleniyu-koda-html-css-2020-12-06>
7. Мови програмування для серверної веб-розробки <https://andreyex.ru/programmirovanie/7-luchshih-yazykov-programmirovaniya-dlya-servernoj-veb-razrabotki/>
8. Селектори CSS. Види, групування та специфічність <https://itchief.ru/html-and-css/selectors>
9. Інструменти та веб-ресурси для веб-розробки <https://techblog.sdstudio.top/blog/instrumenty-i-veb-resursy-dlia-veb-razrabotki-100-plus>
10. Кращі фреймворки CSS <https://proglib.io/p/9-luchshih-freymvorkov-css-aktualnyh-v-2021-godu-2021-03-10>
11. Кращі JS-фреймворки та тенденції веб-розробки <https://medium.com/nuances-of-programming/лучшие-javascript-фреймворки-и-тенденции-веб-разработки-в-2021-году-2a35e348a12>