

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

**Факультет комп'ютерних наук, фізики та математики**

**Кафедра інформатики, програмної інженерії та економічної кібернетики**

**РОЗРОБКА АНАЛІТИЧНОГО РЕСУРСУ ВІДСТЕЖЕННЯ  
ПРАЦЕЗДАТНОСТІ ІТ-СЕРВІСІВ УНІВЕРСИТЕТУ**

**Кваліфікаційна робота (проект)**

на здобуття ступеня вищої освіти «бакалавр»

Виконав: студент 4 курсу

Спеціальності: 122 Комп'ютерні науки

Освітньо-професійної програми:

Комп'ютерні науки

Капустніков Лев Сергійович

Керівники: кандидатка педагогічних наук,

доцентка Кушнір Наталія Олександрівна

доктор педагогічних наук, професор

Співаковський Олександр Володимирович

Рецензент: кандидат педагогічних наук, доцент

Таточенко Володимир Іванович

Херсон – 2021

## ЗМІСТ

<b>ВСТУП</b> .....	4
<b>РОЗДІЛ 1 Аналіз предметної області, приклади, порівняння та основні поняття</b> .....	6
1.1 Аналітичний ресурс відстеження працездатності.....	6
1.2 Способи відстеження працездатності веб-сервісу .....	7
1.3 Огляд існуючих веб-сервісів .....	8
1.4 Порівняння веб-сервісів, визначення найефективнішого .....	16
<b>РОЗДІЛ 2 Методологія, огляд підходів для виконання та їх порівняння</b> .....	18
2.1 Засоби розробки виду та функціональності веб-ресурс .....	18
2.2 Порівняння існуючих технологій .....	29
2.3 HTTP методи виявлення стабільної роботи ресурсу .....	31
<b>РОЗДІЛ 3 Розробка веб-ресурсу з відстеження працездатності</b> .....	35
3.1 Етапи розробки, технічні завдання.....	35
3.2 Блок-схема структури роботи ресурсу .....	35
3.3 Макет дизайну веб-ресурсу .....	37
<b>ВИСНОВКИ</b> .....	38
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	39
<b>ДОДАТОК</b> .....	42
Додаток А.....	42

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AJAX	Asynchronous Javascript and XML
API	Application Programming Interface
CPU	Central Processing Unit
DDoS	Distributed Denial of Service
DNS	Domain Name System
DNSBL	DNS blacklist
DOM	Document Object Model
FPC	Full Page Check
FTP	File Transfer Protocol
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IP	Internet Protocol
JSON	JavaScript Object Notation
PHP	Personal Home Page
PING	Packet Inter-Network Groper
RAM	Random Access Memory
RBM	Real Browser Monitoring
RUM	Real User Monitoring
SMS	Short Message Service
SNMP	SimpleNetworkManagement Protocol
SQL	Structured Query Language
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
URL	Uniform Resource Locator
XML	eXtensible Markup Language

## ВСТУП

**Актуальність:** Актуальність теми обумовлюється саме принципом проекту – моніторинг веб-ресурсів - це процес спостереження за поведінкою сайту. Він використовується, щоб контролювати роботу сайтів і інших сторінок, доступність яких вкрай важлива для користувачів.

Відповіді, отримані як результат перевірки, зберігаються і аналізуються. Зазвичай, моніторинг відбувається з різних серверів, віддалених географічно - може бути з окремих частин світу або країн, або ж може бути глобальним. Зібрана інформація може бути представлена в різній формі: звіти у вигляді електронних листів, різні графіки і залежності, створені для того, щоб клієнт міг отримати всеосяжну картину роботи своїх сайтів. Такі параметри, як час завантаження, швидкість та інші можуть допомогти оптимізувати роботу сайту.

**Мета роботи:** проектування та розробка додатку для відстеження працездатності ІТ-сервісів університету.

**Об'єкт дослідження:** система, яка дозволяє відстежити працездатність веб-ресурсу та технології, які були задіяні для створення.

**Предмет дослідження:** методи здобутку інформації з веб-ресурсу та подальша їх обробка для виявлення найнеобхідніших ознак.

**Практичне значення** полягає у представленні усім бажаючим – як студентам, так і викладачам здатності перевірити працездатність найголовніших веб-ресурсів університету.

**Завдання роботи:**

1. Зробити аналіз предметної області:

1.1 Аналіз існуючих веб-ресурсів аналогів, згідно теми роботи

- 1.2 Аналіз видів, способів спостереження за працездатністю веб-ресурсів
2. Визначити функціональні можливості аналітичного ресурсу, що розроблюється:
  - 2.1 Обґрунтувати вибір методів моніторингу
  - 2.2 Обґрунтувати вибір фреймворків
  - 2.3 Визначити функції, які будуть використовуватися
3. Розробити працюючий додаток:
  - 3.1 Розробити Front-End частину
  - 3.2 Розробити Back-End частину
  - 3.3 Виконати тестування функціональної частини проекту

**Структура роботи:** Дипломна робота складається зі змісту, списку умовних скорочень, вступу, трьох розділів, висновку, списку використаних джерел та додатків.

# РОЗДІЛ 1

## АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ, ПРИКЛАДИ, ПОРІВНЯННЯ ТА ОСНОВНІ ПОНЯТТЯ

### 1.1 Аналітичний ресурс відстеження працездатності

Моніторинг веб-сайтів - це всеохоплюючий термін для будь-якої діяльності, яка передбачає тестування веб-сайту або веб-служби на наявність, ефективність чи функціональність. Служба моніторингу веб-сайтів перевіряє та перевіряє, чи працює сайт, і що відвідувачі можуть користуватися ним, як очікувалося.

Термін «моніторинг веб-сайтів» стосується будь-якої діяльності, яка перевіряє наявність, ефективність та функціонування веб-сайту або веб-служби. Зазвичай цей термін стосується автоматизованого тестування або моніторингу реальних користувачів (RUM), але деякі сайти все ще не тестують взагалі або покладаються на періодичні перевірки, що проводяться працівниками. Тестування вручну є епізодичним і неможливим, якщо враховувати кількість змінних, які впливають на доступність, продуктивність та функції веб-сайту.

Найважливіші кроки роботи, перевірки веб-ресурсу:

- Ініціювання підключення до веб-сайту або служби.
- Перевірка повернення коду відповіді. Для базової доступності контрольно-пропускний пункт повідомляє результат і вважає тест завершеним, але для більш розширеного моніторингу контрольний пункт продовжується.
- Перевірка повернення для вказаного вмісту
- Завантажує вміст у реальний браузер (RBM).
- Записує час завантаження кожного елемента сторінки під час завантаження у браузері (моніторинг ефективності).

- Спроби увійти в систему, провести пошук, скористатися кошиком для покупок, навіть здійснити покупку (моніторинг веб-додатків).
- Повідомляє свої висновки службі моніторингу.

## 1.2 Способи відстеження працездатності веб-сервісу

### Моніторинг доступності

Доступність - це час безвідмовної роботи, або іншими словами, забезпечення веб-сайту чи служби завжди доступними та певною мірою функціональними. Доступність може включати веб-служби, домени та сторінки.

*Базовий моніторинг веб-сайтів та API* - ці базові методи перевіряють успішну відповідь або конкретну відповідь веб-сайтів та API, які підтримують протокол HTTP, і вони можуть виконувати базову автентифікацію. Основні методи доступності також можуть вимірювати час і розмір відповіді та видавати сповіщення про повільний час відгуку. Методи HTTP (S) не завантажують вміст у браузер, але служба моніторингу може перевірити відповідь на наявність або відсутність зазначених слів, фраз або регулярного виразу.

*Доступність сервера* - до тих пір, поки сервер або пристрій розпізнає протокол TCP / IP, служба моніторингу може перевірити наявність пристрою та вибраних портів. Служба моніторингу може перевіряти доступність один раз на хвилину, запобігаючи довгим простоям і втраті продуктивності в Інтернеті або за брандмауером.

*Розширена доступність* - ці спеціалізовані автоматизовані методи перевіряють записи DNS, перевіряють правильність конфігурації сертифікатів SSL, запитують бази даних, входять на сервери електронної пошти та завантажують файли з FTP-серверів.

## **Моніторинг ефективності**

Моніторинг ефективності перевіряє швидкість веб-сайту або служби. Методи продуктивності відстежують час для швидкості з'єднання (frontend and backend) та час завантаження браузера. Методи ефективності можуть використовувати технологію синтетичного моніторингу або RUM. RUM та FPC - Full Page Check забезпечують найповніший набір даних про ефективність. Повна перевірка сторінки надає докладні дані про ефективність кожного елемента на сторінці. Монітори продуктивності видають попередження про помилки сторінок, відсутність вмісту та низьку продуктивність.

## **Моніторинг функціональності**

Методи веб-додатків або методи транзакцій перевіряють функціональність сайту. Ці спеціалізовані методи використовують файли скриптів, які взаємодіють з формами, пошуком на сайті, кошиками для покупок та платіжними системами. Методи транзакцій взаємодіють із веб-додатком так само, як і звичайні відвідувачі, і вони зазвичай перевіряють передбачувані шляхи для виконання завдання. Якщо виникає помилка або продуктивність падає, система видає попередження службі підтримки. Багато різних помилок можуть заблокувати користувачам використання веб-програми, яку методи доступності та продуктивності не можуть виявити.

### **1.3 Огляд існуючих веб-сервісів**

#### **HOST-TRACKER.COM**

HOST-TRACKER - це сервіс для моніторингу працездатності сайту. Ловить, сповіщає і проводить логістику усього розмаїття помилок сайтів і веб



додатків, що спрощує діагностику, покращує Аптайм і комерційні показники ресурсу. Сервіс базується на хмарних технологіях, що підвищує його стійкість і значно знижує можливість помилкових спрацьовувань.

Перевірити можна:

- Доступність сайту по протоколу HTTP методами HEAD, GET, POST, PUT, DELETE, PATCH;
- Доступність утилітою Ping;
- Доступність портів (можуть перевірятися як стандартні порти, так і будь-які на вибір користувача);
- Домен або IP адреса на присутність в списках DNSBL;
- Наявність або відсутність на сторінці заданих ключових слів;
- Час відгуку сайту при заздалегідь визначених запитах;
- Працездатність баз даних (потрібно вказати дані для доступу до БД, можливо виконання SQL запитів);
- Мережеву інфраструктуру по протоколу SNMP;
- Стан сервера (використання CPU, RAM, диска або файлової системи, час з'єднання з портом або сервером SQL), для чого встановлюється монітор на PHP або ASP.NET, або обробляються дані в форматі JSON, отримані з зазначеного URL;
- Витікання терміну реєстрації доменного імені або SSL сертифікату.
- При необхідності виконання декількох однакових перевірок для різних доменів / IP можна додати список адрес і не редагувати кожне завдання окремо. Є можливість додавання завдань Stop для виконання перевірок за розкладом.

- Практично для всіх типів перевірок можливий вибір точок, з яких вони будуть проводитися (11 в Росії, 41 в Європі, 15 в Азії, 14 у Північній Америці, 4 у Південній Америці і по 2 в Африці та Австралії).
- Повідомлення можна отримувати по електронній пошті, через SMS або дзвінок по телефону, за допомогою Skype, Hangouts, Viber, а також через HTTP запит POST. (Рисунок 1.1)

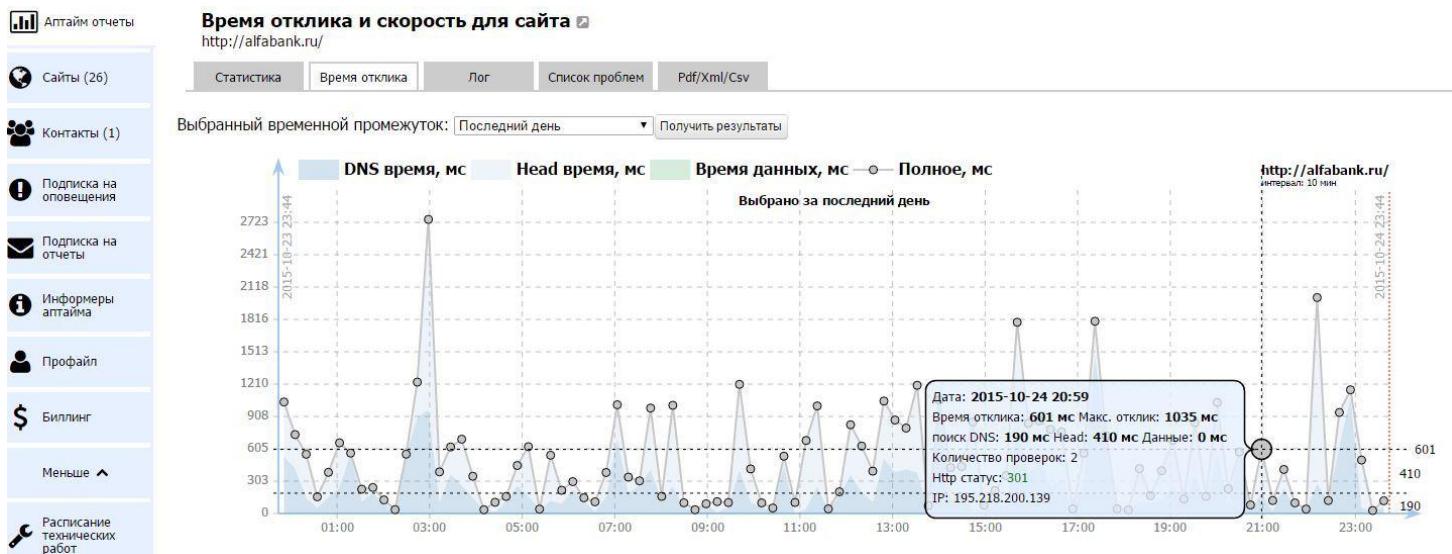


Рисунок 1.1 - Скріншот ресурсу host-tracker.com

## MONITIS.COM

Monitis - це безкоштовна програма, яка дозволяє отримати доступ до ваших рішень Monitis для веб- та хмарного моніторингу та контролювати поточний стан ваших ІТ-систем незалежно від того, де ви перебуваєте.

Відстежуйте веб-сервери, мережі, додатки, хмарні системи, веб-сайти, поведінку реальних користувачів та отримуйте миттєві сповіщення за допомогою push-сповіщень у дорозі. Monitis - це провідне хмарне рішення для

всіх ваших потреб моніторингу, якому довіряють понад 200 000 веб-професіоналів у всьому світі.

За допомогою Monitis ви можете:

- Зберігайте контроль над своїми ІТ-системами цілодобово та без вихідних
- Отримуйте миттєві сповіщення за допомогою push-сповіщень
- Отримайте вичерпні уявлення про свою інфраструктуру та додатки
- Збільште час безвідмовної роботи, доступність та досвід користувачів ваших ІТ-систем
- Перевірте стан свого веб-сайту та веб-систем у різних місцях по всьому світу
- Доступ до поточних та архівованих даних про ефективність
- Усуньте відключення, перш ніж вони зашкодять вашому бізнесу (Рисунок 1.2)

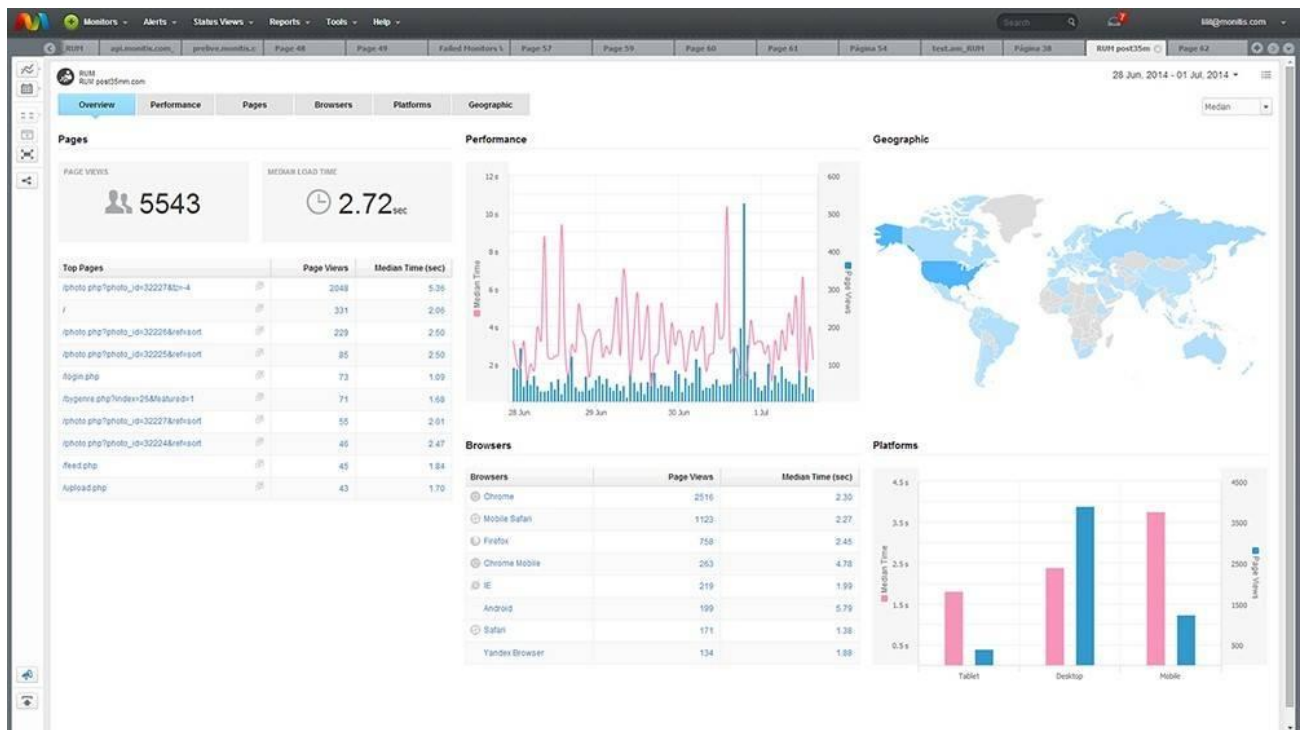


Рисунок 1.2 - Скріншот ресурсу Monitis.com

## **MONITORUS.RU**

Monitorus - це веб-сервіс, який в режимі 24 × 7 × 365 стежитиме за доступністю і безпекою сайту. У разі виникнення будь-яких проблем - негайно повідомить по E-mail або SMS і запропонує вирішення проблеми. Будьте спокійні за свій ресурс - якщо Monitorus мовчить, значить з ним все в порядку. Є можливість установки сертифіката безпеки сайту на кожній сторінці - це підвищить лояльність і довіру відвідувачів.

Сервіс допоможе зрозуміти, скільки часу сайт або сервер не працюють, якщо власник досі не підозрював про простоях.

При цьому щодня антивіруси фіксують зараження сайтів різними вірусами. Все це призводить до блокування ресурсів пошуковими системами і браузерами. Кожна хвилина простою може привести до серйозних наслідків - втрати відвідувачів та прибутку. Може сильно постраждати і імідж, якщо сайт не працюватиме в потрібний момент.

Можливості Monitorus:

- Моніторинг доступності ресурсу або сервера.
- Контроль над файлами сайту.
- Контроль антивірусів.
- Моніторинг редіректу.
- Контроль над доменом.
- Контроль над SSL сертифікатом.
- Кнопки-інформери для сайту.

- Захист від DDoS атак.
- Видалення вірусів.

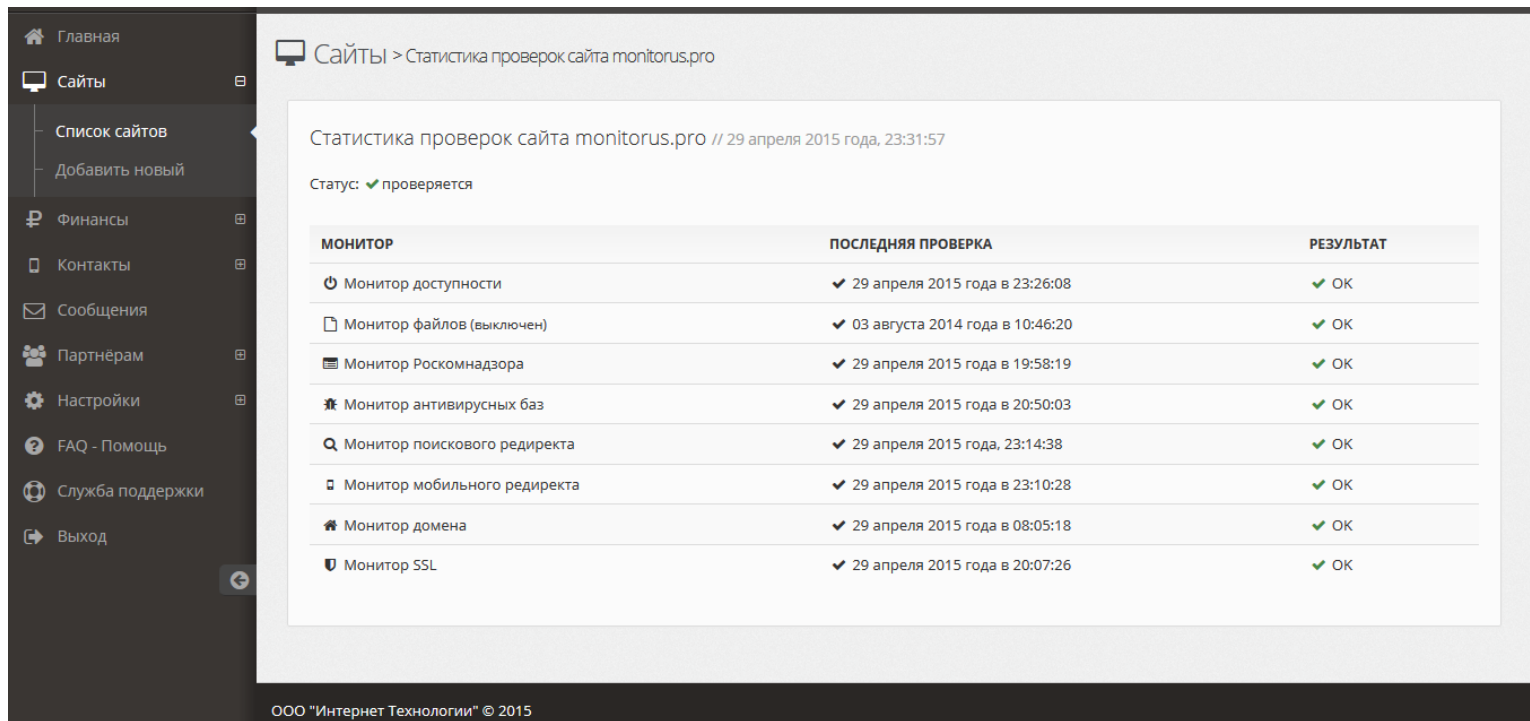


Рисунок 1.3 - Скріншот ресурсу monitorus.ru

## PINGUP.RU

Розподілений сервіс моніторингу сайтів і серверів за допомогою мережі з 54 станцій по всьому світу. Головне завдання - ліквідувати збій на ваших сайтах і серверах. PingUp працює для того, щоб ваші інформаційні системи завжди були онлайн і доступні для вас.

### Можливості PINGUP:

- Підтримка більшості протоколів: ping, HTTP, HTTPS, FTP, SSH, SMTP, DNS, POP3, IMAP, MySQL і будь-які інші протоколи, що використовують TCP / IP.
- Можливість встановити графік технічного обслуговування (разового або періодичного), для того щоб система тимчасово припиняла моніторинг.

- Можливість надавати всю інформацію про Аптайм і працездатності ваших сайтів і серверів в простій і наочній формі, у вигляді звітів і та графіків.
- Повідомлення про збій по електронній пошті, SMS і з допомогою push-повідомлень.



Рисунок 1.4 - Скріншот ресурсу pingup.ru

## **SITE24X7.COM**

Сервіс вмiє вiдправляти запити HTTP методами POST, GET, HEAD, шукати задані ключові слова, перевіряти роботу POP, SMTP, FTP, SOAP, обраний порт, записи DNS на зазначеному сервері, запускати Ping, повідомляти про закінчення терміну дії SSL сертифікату.

Для моніторингу серверів під управлінням Windows, Linux, FreeBSD, OS X пропонуються спеціальні програми.

Повідомлення про несправності можуть бути вiдправлені по електронній пошті, SMS або по телефону, в Twitter, IM.

Site24x7 Products ▾ Plans and Pricing Features Resources ▾ Tools Support Log In [SIGN UP](#)

## Monitoring Availability and End User Experience Made Simple

Performance Monitoring Solution for DevOps and IT Operations.

[PLANS, PRICING AND SIGN UP](#)  
30-DAY FREE TRIAL, SIGN UP IN 30 SECONDS

### All-in-One Monitoring Solution

- Website Monitoring
- Server Monitoring
- Cloud Monitoring
- Network Monitoring
- Application Performance
- Real User Monitoring

We are online! How can we help?

Рисунок 1.5 - Скріншот ресурсу site24x7.ru

## UPTIMEROBOT.COM

Перевіряє HTTP / HTTPS, може шукати ключове слово, може не шукати, запускає Ping, перевіряє заданий порт.

Сповіщення можна отримувати через SMS, електронну пошту, Twitter, HTTP запит в форматі JSON, Push-повідомлення, в Hipchat.

UptimeRobot [Features](#) [Integrations](#) [Status Page](#) [Pricing](#) [Log In](#) [Register for FREE](#)

### Website monitoring

Be the first who knows that your website is down. Website monitoring warns you before any significant troubles and saves you money.

### SSL certificate monitoring

Don't lose visitors because of expired SSL certificate. Get notified 30, 14 and 7 days before expiration so you have time to renew.

### Port & ping monitoring

Is your server online? Is the email service running on port 465 still? Let's check! Monitor any specific port or ping any service.

### Heartbeat monitoring (Cron)

Great way to monitor recurring jobs or internet devices connected to the internet. You send request and we check if it arrives on time.

Рисунок 1.6 - Скріншот ресурсу UPTIMEROBOT.COM

## WEBOPULSAR.RU

Перевіряються доступність по HTTP / HTTPS протоколу методами HEAD, GET, POST, записи на DNS-сервері, Ping, час завантаження сторінки в браузерах IE, Chrome, Firefox.

Повідомлення відправляються на електронну пошту, через SMS, в Jabber або на заданий URL.



Рисунок 1.7 - Скріншот ресурсу WEBOPULSAR.RU

### 1.4 Порівняння веб-сервісів, визначення найефективнішого

Для того, щоб провести процес кількісного або якісного зіставлення різних властивостей (схожості, відмінностей, переваг і недоліків) двох (і більше) об'єктів, з'ясування, який з двох (і більше) об'єктів краще в цілому («інтегральне зіставлення»), твердження, що дані об'єкти дорівнюють або подібні, прирівнюються, уподібнюються треба врахувати усі фактори(Рисунок 1.8):



	host-tracker.com	monitor.us	monitorus.ru	pingup.ru	site24x7.com	uptimerobot.com	webopulsar.ru
Перевірка							
HTTP (Head)	+		+	+	+	+	+
HTTP (Post)	+	+		+	+		+
HTTP (Get)	+	+	+	+	+		+
HTTPS	+	+	+	+	+	+	+
FTP		+	+		+		+
FTPS		+			+		+
Ping	+	+	+	+	+	+	+
Keyword	+		+	+	+	+	+
Port	+	+		+	+		
DNS		+	+		+	+	+
Проверка БД	+	+	+		+		
Мониторинг сервера	+	+			+		
Частота перевірок							
1 минута	+	+	+	+	+		+
2 минуты	+		+	+			
5 минут	+		+	+	+	+	+
10 минут	+		+	+	+	+	+
15 минут	+	+	+	+	+	+	
30 минут	+		+	+	+	+	+
1 час	+		+	+	+	+	+
2 часа	+				+	+	
3 часа					+	+	+
6 часов	+				+	+	+
12 часов	+					+	+
24 часа	+				+	+	+
Повідомлення							
E-mail	+	+	+	+	+	+	+
SMS	+	+	+	+	+	+	+
URL	+	+	+			+	+
Push				+	+	+	

Рисунок 1.8 - Таблица порівняння веб-додатків

## РОЗДІЛ 2

# МЕТОДОЛОГІЯ, ОГЛЯД ПІДХОДІВ ДЛЯ ВИКОНАННЯ ТА ЇХ ПОРІВНЯННЯ

### 2.1 Засоби розробки виду та функціональності веб-ресурсу

#### **Одна із найуніверсальніших мов програмування – JavaScript**

JavaScript - це мова сценаріїв, яка використовується для створення та управління динамічним вмістом веб-сайту, тобто будь-що, що переміщується, оновлюється чи іншим чином змінюється на екрані, не вимагаючи перезавантаження веб-сторінки вручну. Такі функції, як:

- анімована графіка
- слайд-шоу фото
- пропозиції щодо автозаповнення тексту
- інтерактивні форми

JS - це думати про певні веб-функції, якими ви користуєтеся щодня і, напевно, сприймаєте їх як належне - наприклад, коли ваша хронологія Facebook автоматично оновлюється на вашому екрані або Google пропонує пошукові терміни на основі кількох запущених вами букв друкувати. В обох випадках це JavaScript у дії.

Результати JavaScript можуть здатися простими, але є причина, чому ми навчаємо цілому сегменту JavaScript як у нашому веб-розробнику Front End, так і "Вриватися в технічні плани". Під усіма цими чудовими анімаціями та автозавершеннями відбувається щось досить захоплююче. У цьому посібнику буде розказано, як саме працює JavaScript, чому і як ним користуватися, а також найкращі способи вивчення JavaScript, якщо ви усвідомлюєте, що він вам потрібен у вашому наборі навичок.

## **JAVASCRIPT використовується для автоматизації та анімації**

Мова сценаріїв, така як JavaScript важко піднімає ситуацію, кажучи комп'ютерним програмам, як веб-сайти чи веб-програми, “робити щось”. У випадку з JavaScript це означає сказати тим динамічним функціям, які були описані раніше, робити все, що вони роблять - наприклад, наказувати зображенням анімуватись, фотографії циклічно демонструвати слайд-шоу або пропозиції автозаповнення відповідати на підказки. Це „сценарій” у JavaScript, який змушує ці речі відбуватися начебто самі по собі.

Тим часом, оскільки JavaScript є такою невід'ємною частиною веб-функціональності, усі основні веб-браузери мають вбудовані механізми, які можуть відображати JavaScript. Це означає, що команди JS можна вводити безпосередньо в документ HTML, і веб-браузери зможуть їх зрозуміти. Іншими словами, використання JavaScript не вимагає завантаження додаткових програм чи компіляторів.

Також JS використовується для :

- Додавання інтерактивності до веб-сайтів - так, якщо ви хочете, щоб веб-сайт був більше, ніж статична сторінка тексту, вам потрібно буде виконати деякі сценарії JavaScripting
- Розробка мобільних додатків. JavaScript призначений не лише для веб-сайтів ... він використовується для створення додатків, які є у вас на телефоні та планшеті
- Створення ігор на основі веб-браузера - Ви коли-небудь грали в гру безпосередньо з веб-браузера? Ймовірно, JavaScript допоміг це здійснити
- Інтернетна веб-розробка - так, JavaScript В основному використовується на передньому кінці речей, але це досить універсальна мова сценаріїв, яка також може бути використана в інфраструктурі інтерфейсу.

## Переваги JavaScript

- **Швидкість** - JavaScript має тенденцію бути дуже швидким, оскільки він часто запускається відразу в браузері клієнта. Поки він не вимагає зовнішніх ресурсів, JavaScript не сповільнюється через виклики до серверного сервера. Крім того, всі основні браузери підтримують компіляцію JIT (вчасно) для JavaScript, що означає, що немає необхідності компілювати код перед його запуском.
- **Простота** - синтаксис JavaScript був натхненний Java і його порівняно легко вивчити порівняно з іншими популярними мовами, такими як C ++.
- **Популярність** - JavaScript є скрізь в Інтернеті, і з появою Node.js все частіше використовується на серверній основі. Існує незліченна кількість ресурсів для вивчення JavaScript. Як StackOverflow, так і GitHub демонструють дедалі більшу кількість проєктів, які використовують JavaScript, і привабливість, яку вона здобула за останні роки, очікується лише збільшення.
- **Сумісність** - На відміну від PHP чи інших мов сценаріїв, JavaScript можна вставити на будь-яку веб-сторінку. JavaScript можна використовувати в багатьох різних видах програм завдяки підтримці інших мов, таких як Pearl та PHP.
- **Навантаження сервера** - JavaScript працює на стороні клієнта, тому загалом зменшує попит на сервери, а простим програмам сервер може взагалі не знадобитися.
- **Розширені інтерфейси** - JavaScript можна використовувати для створення таких функцій, як перетягування та компонентів, таких як повзунки, що все значно покращує користувацький інтерфейс та досвід роботи на сайті.
- **Розширена функціональність** - Розробники можуть розширити функціональність веб-сторінок, написавши фрагменти JavaScript для сторонніх доповнень, таких як Greasemonkey.

- **Універсальність** - Є багато способів використовувати JavaScript через сервери Node.js. Якщо ви хотіли завантажити Node.js за допомогою Express, використовувати базу даних документів, як MongoDB, і використовувати JavaScript на інтерфейсі для клієнтів, можна розробити цілу програму JavaScript спереду назад, використовуючи лише JavaScript.
- **Оновлення** - З моменту появи ECMAScript 5 (специфікація сценаріїв, на яку покладається JavaScript), ECMA International займається оновленням JavaScript щорічно. Наразі ми отримали підтримку браузера для ES6 у 2017 році і сподіваємось на підтримку ES7 у майбутньому.

## Бібліотека jQuery

jQuery - це легка бібліотека JavaScript з відкритим кодом, яка допомагає нам створювати інтерактивні веб-сторінки з анімацією, візуальними ефектами та розширеною функціональністю. Це найпопулярніша бібліотека JavaScript, якою користуються близько 70 мільйонів веб-сайтів у всьому світі.

Девізом jQuery є «писати менше, робити більше», оскільки він зменшує багато рядків необробленого коду JavaScript в один рядок за допомогою простого інтерфейсу. Основні особливості jQuery включають:

- Обробка подій
- Маніпуляція DOM
- Анімація та ефекти
- Фреймворк AJAX

Давайте подивимося, як jQuery працює для спрощення нашого коду JavaScript. Ось популярний метод маніпуляції, який ми використовуємо з ванільним JavaScript для завантаження деяких абзаців у DOM (Рисунок 1.9):

```
// Updating the text present in 4 paragraph elements
document.addEventListener("DOMContentLoaded", function() {
  paragraphs = document.getElementsByTagName("p")
  for (i = 0; i <= paragraphs.length; i++) {
    paragraphs[i].innerHTML = "This is a Paragraph";
  }
});
```

Рисунок 1.9 - Зразок коду Java Script

Подивіться, як він порівнюється з деяким jQuery, який дає ті самі результати. Бібліотека jQuery використовує JavaScript під капотом, щоб зробити код більш читабельним.(Рисунок 2.1)

```
// Updating the text present in 4 paragraph elements
$(document).ready(function() {
  $("p").html("This is a Paragraph")
});
```

Рисунок 2.1 - Зразок коду JS

Бібліотека jQuery досі є однією з найпопулярніших бібліотек, що зустрічаються у застарілому коді. Його мантра про те, щоб писати менше, щоб робити більше, робить її популярною бібліотекою, яку можна продовжувати використовувати і сьогодні.

*"Пишіть менше, робіть більше".*

jQuery - це, мабуть, найпопулярніша та розширювана бібліотека JavaScript. Він використовується великими компаніями, такими як Netflix,

Google, IBM та Microsoft. IT - зазвичай перша бібліотека, яку розробники JavaScript вивчають, оскільки:

- Його синтаксис для викликів AJAX дуже простий у порівнянні з власним XMLHttpRequest
- Він пропонує скорочені рішення загальних завдань JavaScript, таких як сортування та фільтрування масивів
- Багато інших бібліотек мають залежності jQuery
- Він працює на стороні сервера Node

Нещодавні досягнення у світі інтерфейсів за допомогою таких фреймворків JavaScript, як React та Angular та API вибору, роблять jQuery трохи довгим, але випадки використання все ще існують залежно від того, що потрібно для вашого проекту.

Поки Bootstrap і Wordpress використовують jQuery для створення своїх компонентів та тем. Багато великих технологічних компаній, хоча вони і не можуть починати зелені проекти за допомогою jQuery, все ж базуються на тих, які розпочали з jQuery.

## **Технологія AJAX**

AJAX розшифровується як асинхронний JavaScript та XML. AJAX - це нова техніка для створення кращих, швидших та інтерактивніших веб-додатків за допомогою XML, HTML, CSS та JavaScript.

- Ajax використовує XHTML для вмісту, CSS для презентації, а також об'єктну модель документа та JavaScript для динамічного відображення вмісту.

- Звичайні веб-програми передають інформацію на сервер і з нього за допомогою синхронних запитів. Це означає, що ви заповнюєте форму, натискаєте кнопку подати та отримуєте перехід на нову сторінку з новою інформацією з сервера.
- За допомогою AJAX, коли ви натискаєте кнопку подати, JavaScript зробить запит на сервер, інтерпретує результати та оновить поточний екран. У чистому розумінні користувач ніколи не дізнається, що щось навіть передається на сервер.
- XML зазвичай використовується як формат отримання даних сервера, хоча може використовуватися будь-який формат, включаючи звичайний текст.
- AJAX - це технологія веб-браузера, незалежна від програмного забезпечення веб-сервера.
- Користувач може продовжувати використовувати додаток, поки програма-клієнт запитує інформацію від сервера у фоновому режимі.
- Інтуїтивна та природна взаємодія користувача. Клацання не потрібно, рух миші є достатнім активатором події.
- На основі даних, на відміну від сторінок.



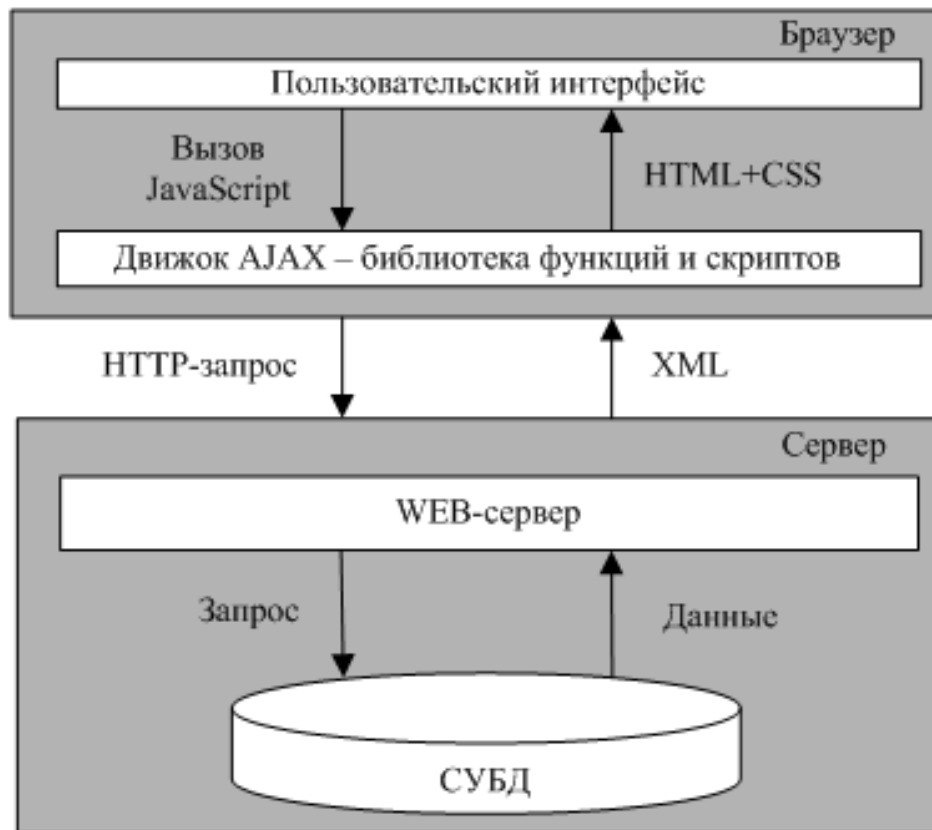


Рисунок 2.2 - Блок-схема роботи бібліотеки AJAX

## Всеоб'ємні технології застосування Інтернету

На даний момент AJAX є найбільш життєздатною технологією Rich Internet Application (RIA). Він набуває величезного промислового імпульсу, і з'являється кілька наборів інструментів та рамок. Але в той же час AJAX має несумісність браузера, і це підтримується JavaScript, який важко підтримувати та налагоджувати.

## AJAX базується на відкритих стандартах

AJAX базується на наступних відкритих стандартах -

- Презентація на основі браузера з використанням HTML та каскадних таблиць стилів (CSS).

- Дані зберігаються у форматі XML і отримуються з сервера.
- Заулісні дані отримують за допомогою об'єктів XMLHttpRequest у браузері.
- JavaScript для реалізації всього.

За словами Каскарано, JavaScript та XML поєднуються, щоб здійснити асинхронне оновлення, використовуючи щось, що називається об'єктом XMLHttpRequest. Коли користувач відвідує веб-сторінку, призначену для використання AJAX, і виникає встановлена подія (користувач завантажує сторінку, натискає кнопку, заповнює форму тощо) JavaScript створює об'єкт XMLHttpRequest, який потім передає дані в XML форматі між веб-браузером (програма, що використовується для перегляду веб-сайту) та веб-сервером (програмне чи апаратне забезпечення, де зберігаються дані веб-сайту). Об'єкт XMLHttpRequest надсилає запит на оновлені дані сторінки на веб-сервер, сервер обробляє запит, відповідь створюється на стороні сервера і надсилається назад у браузер, який потім використовує JavaScript для обробки відповіді та відображення її на екрані як оновлений вміст.

Підсумок: JavaScript автоматизує процес оновлення, запит на оновлений вміст відформатовано у форматі XML, щоб зробити його загальнозрозумілим, і JavaScript знову починає оновлювати відповідний вміст для користувача, який переглядає сторінку. Каскарано зазначає, що техніка AJAX ігнорує сторонні дані сторінки і обробляє лише запити на оновлену інформацію та саму оновлену інформацію. Це справді суть ефективності AJAX, завдяки чому веб-сайти та програми, які використовують AJAX, швидші та швидше реагують на користувачів.

## Інтерфейс бібліотека React

React JS - це бібліотека JavaScript, яка використовується у веб-розробці для створення інтерактивних елементів на веб-сайтах. Але якщо ви не знайомі з JavaScript або бібліотеками JavaScript, це не корисне визначення. Тож давайте зробимо крок назад і спочатку розберемося з цими умовами.

React - це бібліотека JavaScript, яка спеціалізується на допомозі розробникам у створенні користувальницьких інтерфейсів. Що стосується веб-сайтів та веб-додатків, інтерфейси користувача - це набір екранних меню, рядків пошуку, кнопок та всього іншого, з чим хтось взаємодіє для ВИКОРИСТАННЯ веб-сайту чи програми.

React-розробка полягає в описі того, що потрібно вивести на сторінку (а не в складанні інструкцій для браузера, присвячених тому, як це робити). Це, крім іншого, означає значне скорочення обсягів шаблонного коду[11].

До React JS розробники застрягли у створенні користувальницьких інтерфейсів вручну за допомогою «класичного JavaScript» (розробник самостійно говорить за вихідну мову JavaScript) або з менш орієнтованими на UI попередниками React, такими як jQuery. Це означало довший час розробки та безліч можливостей для помилок та помилок. У 2011 році інженер Facebook Джордан Уолк створив React JS спеціально для поліпшення розробки інтерфейсу.

Окрім забезпечення багаторазового коду бібліотеки React (економія часу на розробку та зменшення ризику помилок кодування), React має дві ключові функції, які додають привабливості розробникам JavaScript:

- JSX
- Віртуальний DOM

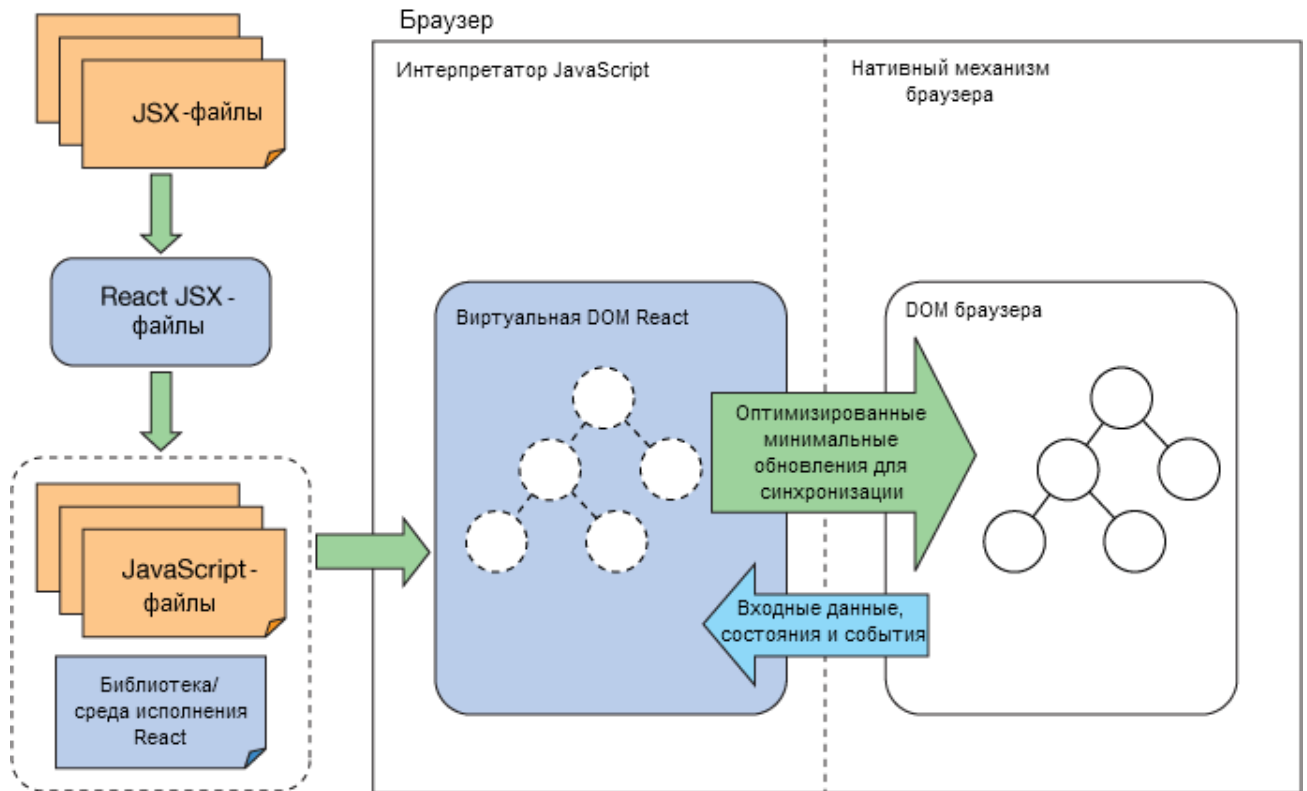


Рисунок 2.3 - Схема роботи бібліотеки React

Основні переваги ReactJS:

- Швидкість
- Гнучкість
- Продуктивність
- Юзабіліті
- Розробка мобільних додатків
- Причини, чому вибирають ReactJS
- Це легко навчитися
- Це допомагає створювати розширені користувальницькі інтерфейси
- Це дозволяє писати власні компоненти

- Це підвищує продуктивність розробників
- Він пропонує швидкий візуалізацію
- Це SEO-зручно
- Він поставляється з корисним набором інструментів для розробників
- Потужна підтримка громади
- Він пропонує кращу стабільність коду
- Він використовується багатьма компаніями з 500 компаній

## 2.2 Порівняння існуючих технологій

Ближче до кінця цієї публікації, давайте згадаємо характерні особливості кожного фреймворку, щоб спробувати відповісти на запитання: Angular vs React vs Vue: який із них вибрати?



Рисунок 2.4 - Логотип Angular

Angular - найбільш зрілий з фреймворків, має хорошу підтримку з точки зору учасників та є повним пакетом.

Однак крива навчання крута, і концепції розвитку в Angular можуть відкласти нових розробників.

Angular - хороший вибір для компаній з великими командами та розробників, які вже використовують TypeScript.

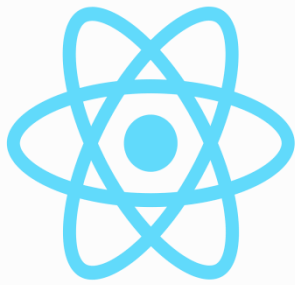


Рисунок 2.5 - Логотип React

React досить старий, щоб бути зрілим, і він має величезну кількість внесків від громади. Він отримав широке визнання. Ринок вакансій для React справді хороший, і майбутнє за цим фреймворком виглядає яскравим.

React виглядає як хороший вибір для тих, хто починає роботу з інтерфейсними фреймворками JavaScript, стартапами та розробниками, яким подобається певна гнучкість. Можливість плавної інтеграції з іншими фреймворками дає велику перевагу тим, хто бажає певної гнучкості у своєму коді.



Рисунок 2.6 - Логотип Vue

Vue є найновішим на арені без підтримки великої компанії.

Однак за останні кілька років це дуже добре зробило, що стало потужним конкурентом для Angular та React, і особливо з виходом Vue 3.0. Можливо, це відіграє роль багатьом китайським гігантам, таким як Alibaba та Baidu, які

вибирають Vue як основну фронтенд-структуру JavaScript. Vue повинен бути вашим вибором, якщо ви віддаєте перевагу простоті, але також любите гнучкість.

## 2.3 HTTP методи виявлення стабільної роботи ресурсу

### Метод GET

HTTP-метод GET дозволяє отримувати інформацію з HTTP-сервера. Інформація, що отримується від сервера, може бути будь-якою, головною, щоб вона була у формі HTTP-об'єкта, доступ до інформації при використанні методу GET здійснюється через URI. Часто відбувається так, що HTTP-метод GET повертається до какому-то коду, а не на конкретних сторінках (усі CMS генерують контент-налету), тому метод GET працює так, що ми отримуємо не вихідний код, який генерує текст і сам текст.

Метод HTTP GET отримує два види: умовний метод GET і частковий метод GET. Давайте сперва подивитися на умовний метод GET. Коли використовується використовуваний метод HTTP GET, до повідомлення HTTP додаються наступні поля заголовків: If-Modified-Since, If-Unmodified-Since, If-Match, If-None-Match або If-Range. Значення таких полів є якою-небудь умовою та, якщо це умови виконується, відбувається передача об'єкта, який зберігається за вказаним URI, якщо умови не виконуються, то і сервер не передає жодних даних. Основний метод HTTP ОТРИМАТИ призначений для зменшення навантажень на сеть.

Приклад відповіді на запит метода GET:

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT
```

ETag: "34aa387-d-1568eb00"

Vary: Authorization,Accept

Accept-Ranges: bytes

Content-Length: 88

Content-Type: text/html

Connection: Closed

## **Метод HEAD**

HTTP метод HEAD працює точно так само, як і метод GET, з тією лише різницею, що сервер у відповідь не посилає тіло HTTP повідомлення. Всі заголовки відповіді при запиті клієнта з використанням методу HEAD ідентичні тим заголовкам, які б були, якби використовувався метод GET. Зазвичай HTTP метод HEAD використовується для отримання метаданих про об'єкт без пересилання тіла HTTP повідомлення. Метод HEAD часто використовується для тестування HTTP з'єднань і досяжності вузлів і ресурсів, так як немає необхідності ганяти по мережі вміст, тестування HTTP методом HEAD виробляється набагато швидше. Сервер може кешувати свої відповіді на запити з методом HEAD. Ще одне застосування методу HEAD полягає в обговорення HTTP вмісту.

Приклад відповіді на запит метода HEAD:

```
HTTP/1.1 200 OK
```

```
Date: Mon, 27 Jul 2009 12:28:53 GMT
```

```
Server: Apache/2.2.14 (Win32)
```

```
Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT
```

```
ETag: "34aa387-d-1568eb00"
```

```
Vary: Authorization,Accept
```

```
Accept-Ranges: bytes
```

```
Content-Length: 88
```



Content-Type: text/html

Connection: Closed

## Метод POST

HTTP метод POST є другим по використанню в Інтернеті і потрібен для того, щоб відправляти дані на сервер. HTTP метод POST дозволяє відправляти дані на сервер. Розробники ввели метод POST в HTTP стандарт, щоб клієнти могли:

- залишати повідомлення на різних Інтернет-ресурсах;
- передавати інформацію про себе, заповнюючи HTML форми;

Те, як буде працювати метод POST визначається виключно на стороні сервера і зазвичай залежить від запитуваної URI. Якщо порівняти URI, якого звертається клієнт і повідомлення, яке він хоче відправити з файлової системою, то URI - це папка, а повідомлення клієнта - це файл, який лежить в папці.

В результаті виконання HTTP методу POST сервер не обов'язково в якості ресурсу видає URI, код стану сервера при використанні HTTP методу POST може бути 200 (в цьому випадку ви отримаєте будь-якої ресурс), або 204 (в цьому випадку ви не отримаєте ніякого вмісту) . Відповіді сервера на метод POST НЕ кешуються, але це можна зробити примусово, якщо використовувати поле Cache-Control або Expires в заголовку.

Приклад відповіді на запит метода POST:

```
HTTP/1.1 200 OK
```

```
Date: Mon, 27 Jul 2009 12:28:53 GMT
```

Server: Apache/2.2.14 (Win32)

Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT

ETag: "34aa387-d-1568eb00"

Vary: Authorization,Accept

Accept-Ranges: bytes

Content-Length: 88

Content-Type: text/html

Connection: Closed

```
<html>
<body>
<h1>Request Processed Successfully</h1>
</body>

</html>
```

**GET**  
**HEAD**

извлечение информации  
извлечение заголовков ресурсов

**POST**

отправляйте данные на сервер.

Рисунок 2.7 - Короткий опис HTTP запитів

## РОЗДІЛ 3

### РОЗРОБКА ВЕБ-РЕСУРСУ З ВІДСТЕЖЕННЯ ПРАЦЕЗДАТНОСТІ

#### 3.1 Етапи розробки, технічні завдання

У своїй роботі я хочу виділити виконані технічні завдання, як:

- Аналітика цільової аудиторії, конкурентів та ін.
- Стратегія, основна ідея продукту
- Прототипування
- Створення тестових дизайнів, макетів
- Front-end частина
- Back-end частина
- Тестування

Етапи розробки веб-ресурсу:

- Постановка задач
- Первинний пошук теоретичної інформації
- Підготовка технічного та програмного забезпечення для виконання
- Визначення технологій, фреймворків для розробки
- Розробка
- Тестування

#### 3.2 Блок-схема структури роботи ресурсу

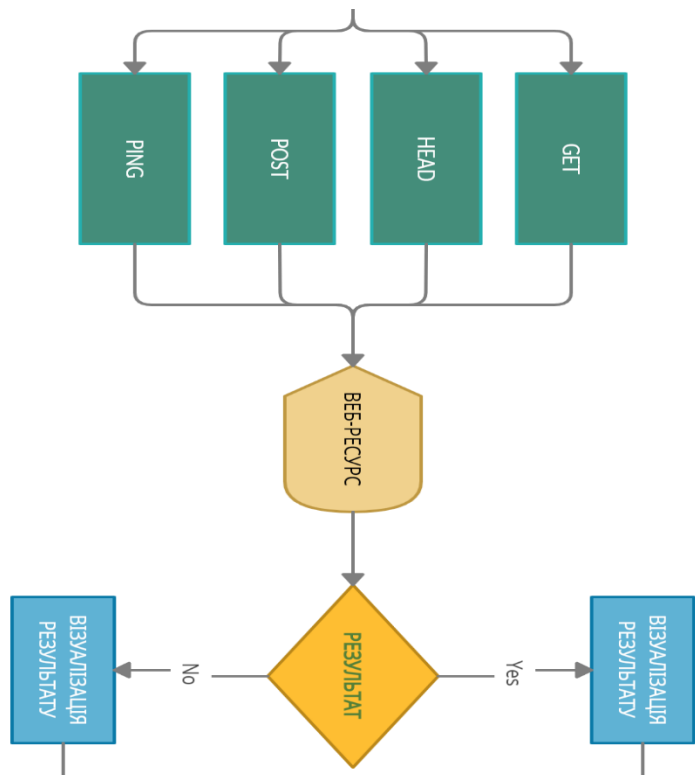
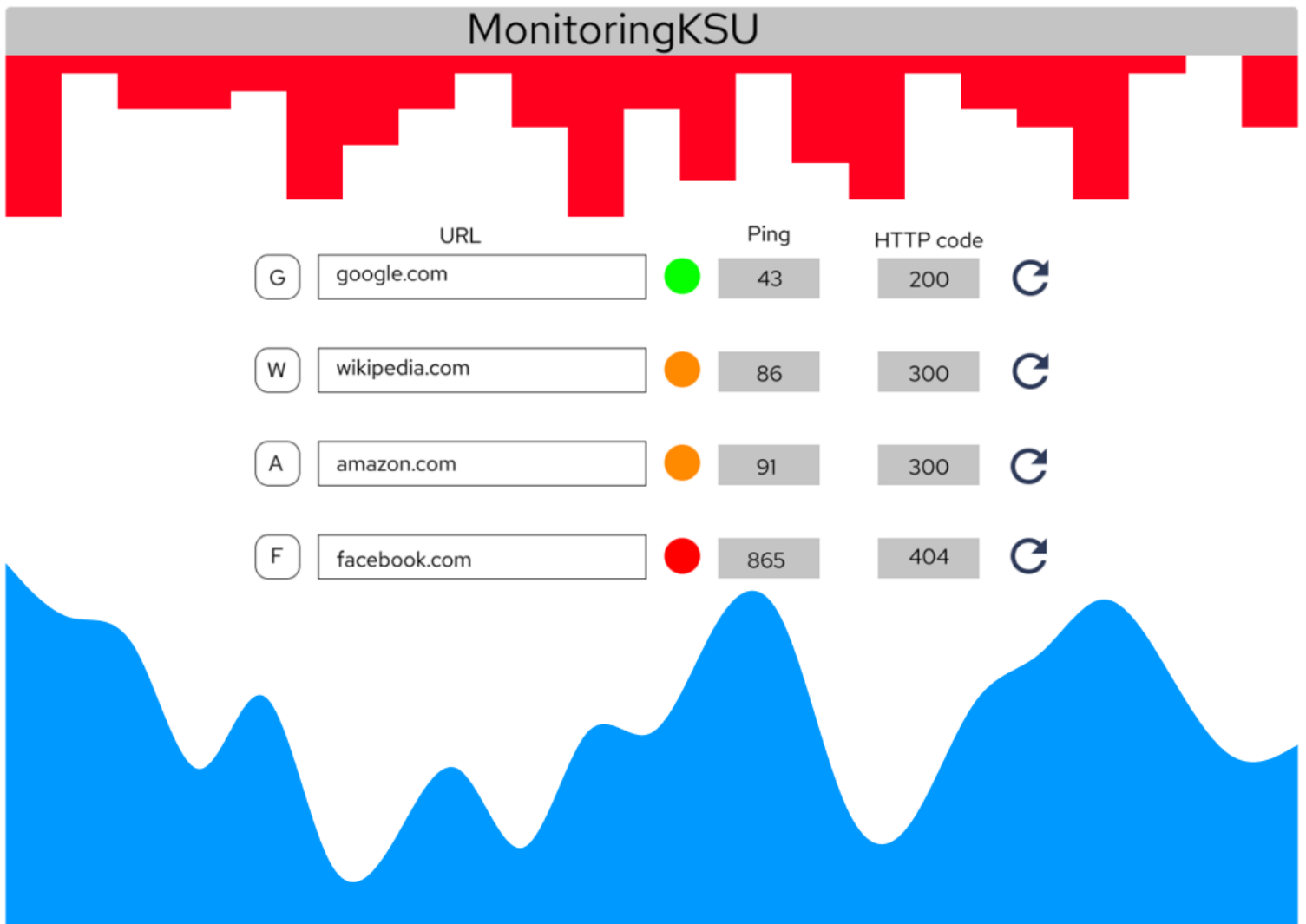


Рисунок 2.8 - Блок-схема структури роботи системи



### 3.3 Макет дизайну веб-ресурсу

Рисунок 2.9 - Макет дизайну веб-ресурсу

## **ВИСНОВКИ**

В процесі виконання кваліфікаційної роботи були поставлені конкретні цілі, завдання, які були виконані, а саме:

- Зробити аналіз предметної області;
- Визначити функціональні можливості аналітичного ресурсу, що розроблюється;
- Розробка Front-end та Back-end частини програми з відстеження працездатності сервісів університету й проведення його тестування;

Було проведено теоретично-технічний аналіз інформації стосовно фреймворків React та Ajax та виявлено і в подальшому реалізовано сукупність функцій, методів моніторингу та властивостей даного проекту з відстеження працездатності IT-сервісів університету.

Також було розроблено інтерфейс зазначеної вище програми, систему автоматизованої відправки звітів та проведено тестування даного продукту.

Моніторинг працездатності призначений для створення знімка поточного стану працездатності системи, щоб можна було перевірити правильність роботи всіх компонентів системи.

## **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

- 1) Website Monitoring Tools [Електронний ресурс] - Режим доступу:  
<https://websitesetup.org/website-monitoring-services/>
- 2) Server Health Monitoring Software [Електронний ресурс] - Режим доступу:  
<https://www.manageengine.com/network-monitoring/server-health-monitor.html>
- 3) JavaScript libraries [Електронний ресурс] - Режим доступу:  
<https://getflywheel.com/layout/best-javascript-libraries-frameworks-2020/>
- 4) An overview of HTTP [Електронний ресурс] - Режим доступу:  
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>
- 5) HTML & CSS [Електронний ресурс] - Режим доступу:  
[https://www.w3.org/standards/webdesign/htmlcss#:~:text=HTML%20\(the%20Hypertext%20Markup%20Language,for%20a%20variety%20of%20devices.](https://www.w3.org/standards/webdesign/htmlcss#:~:text=HTML%20(the%20Hypertext%20Markup%20Language,for%20a%20variety%20of%20devices.)
- 6) Typescript vs JavaScript [Електронний ресурс] - Режим доступу:  
<https://www.guru99.com/typescript-vs-javascript.html>
- 7) jQuery.ajax[Електронний ресурс] - Режим доступу:  
<https://api.jquery.com/jquery.ajax/>
- 8) Response from an asynchronous call [Електронний ресурс] - Режим доступу: <https://stackoverflow.com/questions/14220321/how-do-i-return-the-response-from-an-asynchronous-call>
- 9) Response From Asynchronous Calls in JavaScript [Електронний ресурс] - Режим доступу: <https://betterprogramming.pub/how-to-return-a-response-from-asynchronous-calls-in-javascript-d20e6f49651b>

- 10) The Good and the Bad of ReactJS and React Native [Электронный ресурс] - Режим доступа: <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-reactjs-and-react-native/>
- 11) React.js: понятное руководство [Электронный ресурс] - Режим доступа: <https://habr.com/ru/company/ruvds/blog/428077/>
- 12) Is React.js Fast? [Электронный ресурс] - Режим доступа: <https://jelvix.com/blog/is-react-js-fast>
- 13) What And Why React.js [Электронный ресурс] - Режим доступа: <https://www.c-sharpcorner.com/article/what-and-why-reactjs/>
- 14) Method Definitions [Электронный ресурс] - Режим доступа: <https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>
- 15) Using HTTP Methods for RESTful Services [Электронный ресурс] - Режим доступа: <https://www.restapitutorial.com/lessons/httpmethods.html>
- 16) HTTP Request Methods [Электронный ресурс] - Режим доступа: [https://www.w3schools.com/tags/ref\\_httpmethods.asp](https://www.w3schools.com/tags/ref_httpmethods.asp)
- 17) Методы HTTP запросов [Электронный ресурс] - Режим доступа: <https://zametkinapolyah.ru/servera-i-protokoly/tema-7-opredelenie-metodov-http-http-method-definitions-metody-http-zaprosov.html>
- 18) HTTP HEAD Request [Электронный ресурс] - Режим доступа: <https://reqbin.com/Article/HttpHead>
- 19) http HEAD vs GET performance [Электронный ресурс] - Режим доступа: <https://stackoverflow.com/questions/16539269/http-head-vs-get-performance>
- 20) HEAD [Электронный ресурс] - Режим доступа: <https://developer.cdn.mozilla.net/en-US/docs/Web/HTTP/Methods/HEAD>
- 21) POST [Электронный ресурс] - Режим доступа: <https://developer.mozilla.org/ru/docs/Web/HTTP/Methods/POST>



- 22) Web Course [Электронный ресурс] - Режим доступа:  
<https://webkysr.info/post/http-zapros-metodom-post>
- 23) HTML 5 CSS [Электронный ресурс] - Режим доступа:  
[https://html5css.ru/tags/ref\\_httpmethods.php](https://html5css.ru/tags/ref_httpmethods.php)
- 24) Simple error handling in React with react-error-boundary [Электронный ресурс] - Режим доступа: <https://blog.logrocket.com/error-handling-react-error-boundary/>

## ДОДАТОК

Додаток 1

### КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ХЕРСОНЬСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ

Я, Капустніков Лев Сергійович  
учасник(ця) освітнього процесу Херсонського державного університету, **УСВІДОМЛЮЮ**, що академічна доброчесність – це фундаментальна етична цінність усієї академічної спільноти світу.

**ЗАЯВЛЯЮ**, що у своїй освітній і науковій діяльності **ЗОБОВ'ЯЗУЮСЯ**:

- дотримуватися:
  - вимог законодавства України та внутрішніх нормативних документів університету, зокрема Статуту Університету;
  - принципів та правил академічної доброчесності;
  - нульової толерантності до академічного плагіату;
  - моральних норм та правил етичної поведінки;
  - толерантного ставлення до інших;
  - дотримуватися високого рівня культури спілкування;
- надавати згоду на:
  - безпосередню перевірку курсових, кваліфікаційних робіт тощо на ознаки наявності академічного плагіату за допомогою спеціалізованих програмних продуктів;
  - оброблення, збереження й розміщення кваліфікаційних робіт у відкритому доступі в інституційному репозитарії;
  - використання робіт для перевірки на ознаки наявності академічного плагіату в інших роботах виключно з метою виявлення можливих ознак академічного плагіату;
- самостійно виконувати навчальні завдання, завдання поточного й підсумкового контролю результатів навчання;
  - надавати достовірну інформацію щодо результатів власної навчальної (наукової, творчої) діяльності, використаних методик досліджень та джерел інформації;
  - не використовувати результати досліджень інших авторів без використання покликань на їхню роботу;
  - своєю діяльністю сприяти збереженню та примноженню традицій університету, формуванню його позитивного іміджу;
  - не чинити правопорушень і не сприяти їхньому скоєнню іншими особами;
  - підтримувати атмосферу довіри, взаємної відповідальності та співпраці в освітньому середовищі;
  - поважати честь, гідність та особисту недоторканність особи, незважаючи на її стать, вік, матеріальний стан, соціальне становище, расову належність, релігійні й політичні переконання;
  - не дискримінувати людей на підставі академічного статусу, а також за національною, расовою, статевою чи іншою належністю;
  - відповідально ставитися до своїх обов'язків, вчасно та сумлінно виконувати необхідні навчальні та науково-дослідницькі завдання;
  - запобігати виникненню у своїй діяльності конфлікту інтересів, зокрема не використовувати службових і родинних зв'язків з метою отримання нечесної переваги в навчальній, науковій і трудовій діяльності;
  - не брати участі в будь-якій діяльності, пов'язаній із обманом, нечесністю, списуванням, фабрикацією;
  - не підроблювати документи;
  - не поширювати неправдиву та компрометуючу інформацію про інших здобувачів вищої освіти, викладачів і співробітників;
  - не отримувати і не пропонувати винагород за несправедливе отримання будь-яких переваг або здійснення впливу на зміну отриманої академічної оцінки;
  - не залякувати й не проявляти агресії та насильства проти інших, сексуальні домагання;
  - не завдавати шкоди матеріальним цінностям, матеріально-технічній базі університету та особистій власності інших студентів та/або працівників;
  - не використовувати без дозволу ректорату (деканату) символіки університету в заходах, не пов'язаних з діяльністю університету;
  - не здійснювати і не заохочувати будь-яких спроб, спрямованих на те, щоб за допомогою нечесних і негідних методів досягати власних корисних цілей;
  - не завдавати загрози власному здоров'ю або безпеці іншим студентам та/або працівникам.

**УСВІДОМЛЮЮ**, що відповідно до чинного законодавства у разі недотримання Кодексу академічної доброчесності буду нести академічну та/або інші види відповідальності й до мене можуть бути застосовані заходи дисциплінарного характеру за порушення принципів академічної доброчесності.