

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

## Кваліфікаційна робота магістра

на тему: «Інформаційна технологія керування  
швидкістю завантаження веб-сторінок»

Завідувач

випускаючої кафедри

Довбиш А.С.

Керівник роботи

Проценко О.Б.

Студент групи ІН-м91н

Ковалевський С.О.

Суми – 2021

## **Зміст**

Вступ .....	3
1 ІНФОРМАЦІЙНИЙ ОГЛЯД .....	5
1.1 Літературний огляд .....	5
1.2 Огляд технологій та будови веб-сайту .....	6
1.3 Огляд існуючих веб-сайтів.....	9
1.4 Постановка задачі .....	12
2 ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ.....	14
2.1 Технічні засоби .....	14
2.2 Мови програмування .....	17
2.3 Хостинг .....	17
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ.....	19
3.1 Інформаційна модель .....	19
3.2 Програмна реалізація .....	23
3.3 Результати тестування.....	37
ВИСНОВКИ .....	38
СПИСОК ЛІТЕРАТУРИ .....	39
Додаток – Лістинг файлів .....	41

## ВСТУП

Мережа Інтернет використовується на повсякденній основі більше ніж половиною населення Землі. Вона спрощує повсякденну діяльність, дозволяючи обмінюватись інформацією на відстані, і тому знайшла своє місце у сферах розваг, спілкування, публічного сповіщення та маркетингу. Остання почала набирати популярність не так давно із появою великої кількості онлайн-сервісів, що дозволяють здійснити замовлення довільного товару та оплатити його, не виходячи з дому.

Власники веб-сайтів, що надають подібні послуги, знаходяться у стані постійної боротьби за увагу потенційних конкурентів – кожен з них прагне створити веб-ресурс, який не лише викликає довіру у відвідувача, а й має бути конкурентоспроможним, тобто не гіршим за свої аналоги. Фактично це стосується як електронних магазинів, так і інших сайтів, що містять будь-яку форму реклами – себе, своєї фірми або партнерів, бо кожен відвідувач, що побачив та зацікавився рекламою – це потенційний заробіток для власника. Для досягнення веб-ресурсом популярності багато уваги приділяється його дизайну, зручності у користуванні та атмосфері, яку він викликає. Усі ці фактори спираються на те, як сайт виглядає, тому важливим завданням для розробника є подбати про вибір стилю та реалізувати візуальне оформлення сайту.

Бажання зробити власний додаток привабливішим за інші породжує іншу проблему – сторінку, яку власник бажає показати користувачу, користувач має спершу завантажити на свій комп'ютер, на що потребується деякий час. Поки здійснюється процес завантаження сторінки, браузер користувача буде відображати або її частину, або білий екран, тому відвідувачу доведеться чекати, перш ніж він зможе повноцінно користуватись сайтом. Якщо завантажування триває занадто довго, це дратуватиме та відлякуватиме цільову аудиторію. Щоб уникнути цього, потребується оптимізація сторінки.

У випадку з веб-сайтами під оптимізацією зазвичай розуміється збільшення швидкодії сторінки. Метою оптимізації є досягнути такого стану, при якому сайт одночасно добре виглядає та швидко працює – для цього на сторінці послідовно позбуваються елементів, які сповільнюють її, але не є критично важливими для нормальної роботи. Цей процес виконується програмістом один раз на стадії розробки та визначає те, яким буде кінцевий вигляд сайту – при цьому оптимальність результату не можна гарантувати, бо прийняті заходи є суб'єктивними, а отже можуть задовольнити не усіх користувачів. Вирішити цю проблему дозволяють мови програмування, які формують вміст сайту та дозволяють змінювати його залежно від ситуації.

# 1 ІНФОРМАЦІЙНИЙ ОГЛЯД

## 1.1 Літературний огляд

Питання оптимізації роботи веб-сторінок було досліджено в наступних роботах.

Jatinder Manhas [7] описав поняття продуктивності веб-сторінки, проаналізував результати досягнення високої продуктивності, навів статистику середнього об'єму даних на сайтах. В його роботі надаються поради до підвищення продуктивності, що можна узагальнити, яка зменшення об'єму сторінки та використання кешу.

Oliver Hohlfeld, Dennis Guse та співавтори [8] визначили вплив низької продуктивності на сприйняття сторінки користувачем та наслідки виникнення помилки завантаження окремих елементів.

В роботі І.Н. Єгорової та В.Є. Ригіної [1] вказано, що структура сторінки та візуальне оформлення перш за все залежить від виду та функціональності сайту (наприклад, сітка з однією колонкою для сайту-візитки та 2-3 колонками для корпоративного сайту). Розміри елементів мають бути кратними та визначаються за правилом золотого перетину.

В роботі П.І. Банокіна та В.Н. Вічугова [2] описано метод динамічної генерації веб-сайту на основі семантичного аналізу дій користувача – кожного разу, коли користувач проявляє інтерес до якоїсь теми, ця тема записується до його семантичного профіля, впливаючи на результати, які він отримує. Сам профіль зберігається на стороні клієнта. Ця робота містить обґрунтування необхідності підлаштовувати вміст сторінки під потреби користувача, а не пошукових робіт.

Деарт В. Ю. та Кожухов І. С. [3] описали залежність між якістю досвіду (QoE) та якістю обслуговування (QoS) користувача та перелічили механізми браузера, спрямовані на оптимізацію сторінок.

Ботигін І.О. та Калікін К.О. [4] використовують середній час відгуку сервера, об'єм даних, що передаються мережею та необхідну для виконання пам'ять сервера як критерій продуктивності WEB-додатку.

Ash T., Page R та Ginty M. [5] у своєму керівництві з оптимізації вказують, що потрібно виставити пріоритети, основані на функціях сайту та позбутись елементів, які їм не відповідають.

Naoyu Wang, Yao Guo, Mengxin Liu та Xiangqun Chen [16] описали можливість використання кешу для оптимізації подібних між собою сторінок.

Узагальнення основних засобів оптимізації:

- 1) Медіа-файли максимально стискаються для економії трафіку.
- 2) Некритичні дані завантажуються в останню чергу.
- 3) Найоб'ємніші елементи виносяться з видимої області.
- 4) Наявні дані використовуються повторно.

## 1.2 Огляд технологій та будови веб-сайту

Веб-сайт можливо побудувати як з нуля, так і з використанням фреймворків. Фреймворки представляють собою шаблони з готовою структурою сайту, які можуть бути наповнені довільним змістом, дозволяючи пропустити один з кроків розробки – усі зміни можуть бути зроблені через підготовлений для адміністратора інтерфейс, а самостійно задаються лише потрібні текстові дані.

Розробка вручну передбачає об'єднання наступних технологій.

**HTML** – мова, якою здійснюється розмітка сторінки. Текст розташовується між спеціальними помітками – тегами. Вони визначають структуру сторінки, положення елементів відносно один одного, їх рівень вкладеності. Без використання розмітки вміст сторінки виглядатиме як один абзац простого тексту, це також унеможлиблює поєднання з іншими технологіями. Важливі елементи html-сторінки:

Розділ `head` – містить мета-інформацію, тобто дані про саму сторінку, які не відображаються для користувача. Наприклад: її заголовок (назву вкладки), кодування, мову, підключені файли (стили та скрипти).

Розділ `body` – містить усю інформацію, яка має бути відображена для користувача: текст, зображення, інші теги.

Тег `script` – містить програмний код, зазвичай мовою `javascript`, який виконується після завантаження

Тег `style` – містить фрагмент каскадної таблиці стилів, що впливає на вигляд сторінки

Теги `div`, `table` – блоки або таблиці, знаходяться у `<body>` та задають структуру сторінки.

**PHP** – мова програмування, скрипти якої виконуються на сервері, в той час як користувач отримує лише їх результат. Вона дозволяє отримувати від користувача запит, який обробляється сервером, після чого надсилається відповідь у вигляді готової `html`-сторінки, яку користувач може переглянути. Вміст безпосередньо `php`-скрипта не може бути переглянутий зі сторони відвідувача, тому він дозволяє безпечно обробляти персональні дані, такі як пароль або `ip`-адресу. Обробка даних здійснюється з використанням умовних операторів:

`if` – перевіряє, чи є ствердження вірним і виконує або не виконує задану дію;

`for` – виконує дію циклічно (декілька разів) з періодом, заданим в умові;

`while` – циклічно виконує дію до тих пір, поки виконується умова;

`switch..case` – виконує одну із заданих дій, перевіряючи декілька можливих тверджень.

Використання цих операторів дозволяє завантажувати на сторінку різну інформацію на основі даних, попередньо надісланих користувачем, що використовується для обробки форм або завантажування файлів.

**CSS** – правила візуального оформлення елементів сторінки. Представлені у вигляді каскадної таблиці стилів – переліку стилів, які застосовуються у каскадному порядку – зверху до низу, а отже можуть перекривати один іншого. Не може використовуватись без html.

Найчастіше використовувані стилі:

color – колір текстової інформації;

width, height – висота/ширина елемента/секції;

margin/padding – зовнішні/внутрішні відступи секції;

font – властивості шрифту текстової інформації

Наявність цих стилів перш за все потрібна, щоб перетворити суцільний на статтю зі структурованим виглядом, яку буде зручно читати, а також дозволяє оздоблювати оточуючі її елементи різними шляхами.

**JavaScript** – мова програмування, що надає сторінці інтерактивності, дозволяючи реагувати на дії користувача виконанням відповідних скриптів. Від php відрізняється тим, що виконується не на сервері, а безпосередньо на комп'ютері користувача, а тому не підходить для обміну даними. Скрипти цієї мови дозволяють анімувати об'єкти, ховати та показувати їх, змінювати властивості за певною умовою, наприклад, коли користувач навів курсор на елемент або клікнув по ньому.

Загальний вигляд сценарію звертання до сайту наступний:

- користувач надсилає HTTP запит на завантаження певної сторінки (за замовчуванням – index.php);
- сервер отримує запит та звертається до html-сторінки або php скрипта з відповідною назвою;
- якщо звернення відбулось до файлу php, скрипт у ньому виконується (при цьому з директивою require він може звертатись до інших скриптів), і після завершення його роботи як результат повертається html-сторінка;
- отримавши відповідь від сервера, браузер користувача починає завантажувати сторінку, послідовно зчитуючи її код.



Деталі останнього пункту є важливими для процесу оптимізації, бо різні елементи сторінки в цьому випадку мають різну поведінку. Теги html та текст мають невелику вагу, тому з'являються на екрані майже миттєво. Коментарі завантажуються, як і звичайний текст, але не відображаються. Якщо ж у коді сторінки є набір стилів css або скрипт, то її завантаження зупиняється до тих пір, поки цей фрагмент коду не буде виконаний повністю. Тобто, якщо сторінка починається із JS-коду, користувач бачитиме лише білий екран до тих пір, поки код не буде повністю виконаний, і лише після цього почнуть з'являтися інші елементи. З іншого боку, якщо стилі сторінки завантажити після безпосередньо її вмісту, то її зовнішній вигляд змінюватиметься у користувача прямо на очах, що викликає потенційні проблеми через зміщення елементів з їх початкових місць.

### 1.3 Огляд існуючих веб-сайтів

Як зразок досконалої оптимізації може слугувати сайт пошукової системи Google

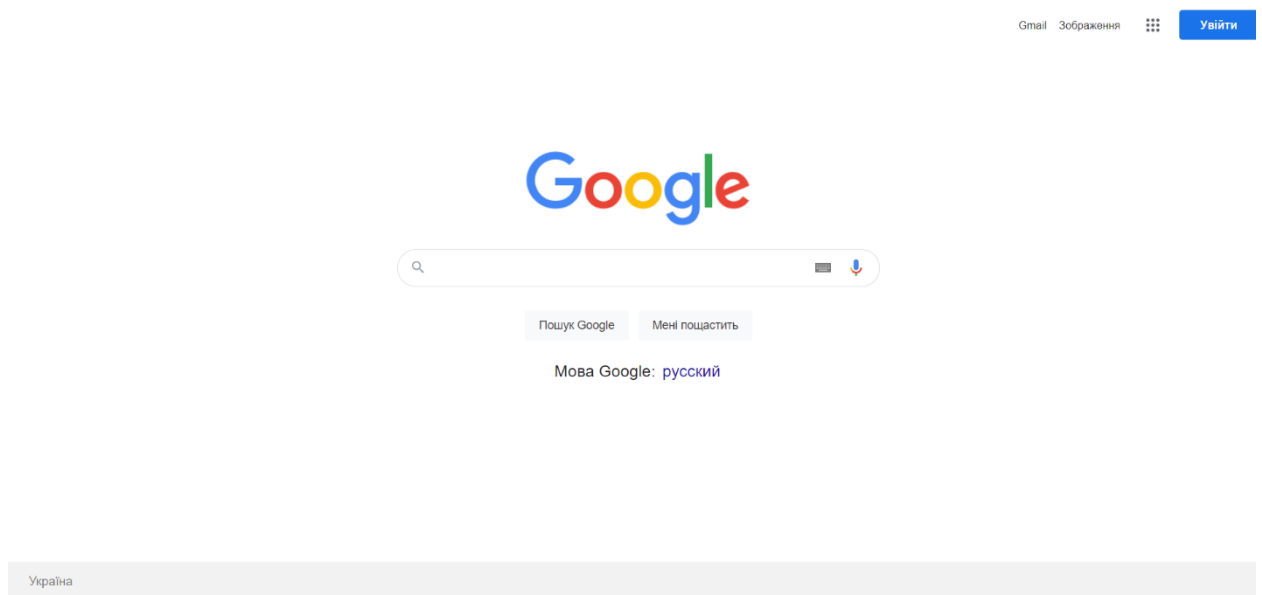


Рисунок 1.1 – Вигляд сторінки Google

Видно, що ця сторінка зроблена настільки простою, наскільки це можливо – присутні два основних елементи: логотип, за яким користувач розуміє, що потрапив у потрібне місце, та пошуковий рядок, яким він

користується. Важливі посилання винесені на бокову лінію, в центрі екрану немає зайвих елементів, які сповільнюють завантаження або відволікають користувача. Для порівняння, сайт-аналог Yahoo! виглядає наступним чином:

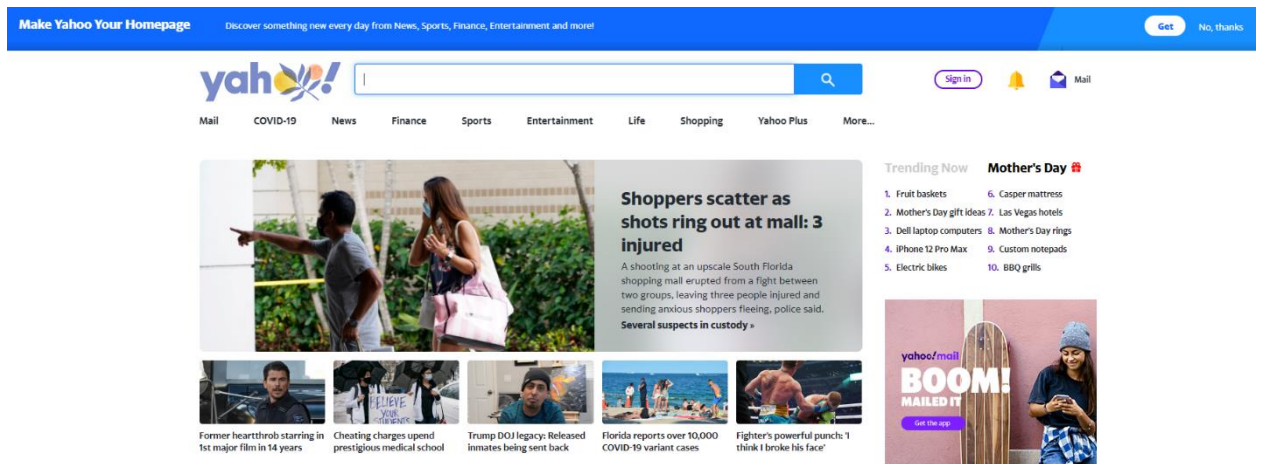


Рисунок 1.2 – Вигляд сторінки Yahoo!

Велика кількість реклами дає додаткові можливості для заробітку, але якщо взяти відвідуваність сайту як критерій його успішності, Google значно випереджає і на даний момент є світовим лідером серед пошукових систем [9].

- Google — 86,30 %
- Yahoo — 5,30 %
- Bing — 3,13 %
- Baidu — 3,02 %
- Yandex — 1,7 %
- Ask — 0,67 %
- AOL — 0,44 %
- Bing — 0,08 %
- AltaVista — 0,07 %
- Excite — 0,03 %
- Lycos — 0,02 %
- AlltheWeb — 0,01 %

Рисунок 1.3 – Відсоток використання пошукових систем на Заході

Одними з причин такої популярності є швидкість роботи та відсутність нав'язливої реклами, що є важливим для пошукової системи, адже користувач не хоче затримуватись на сторінці, його мета – знайти та перейти на інший

сайт, а те, як швидко йому це вдалося, впливатиме на вірогідність його повернення.

З іншого боку бувають ситуації, коли швидкість не є критичним показником і можна зосередитись на зовнішньому вигляді і працездатності. Це можна побачити на прикладі двох комерційних сайтів зі спільною тематикою: “продаж автоматичних воріт”.

Перший сайт – budlux.in.ua

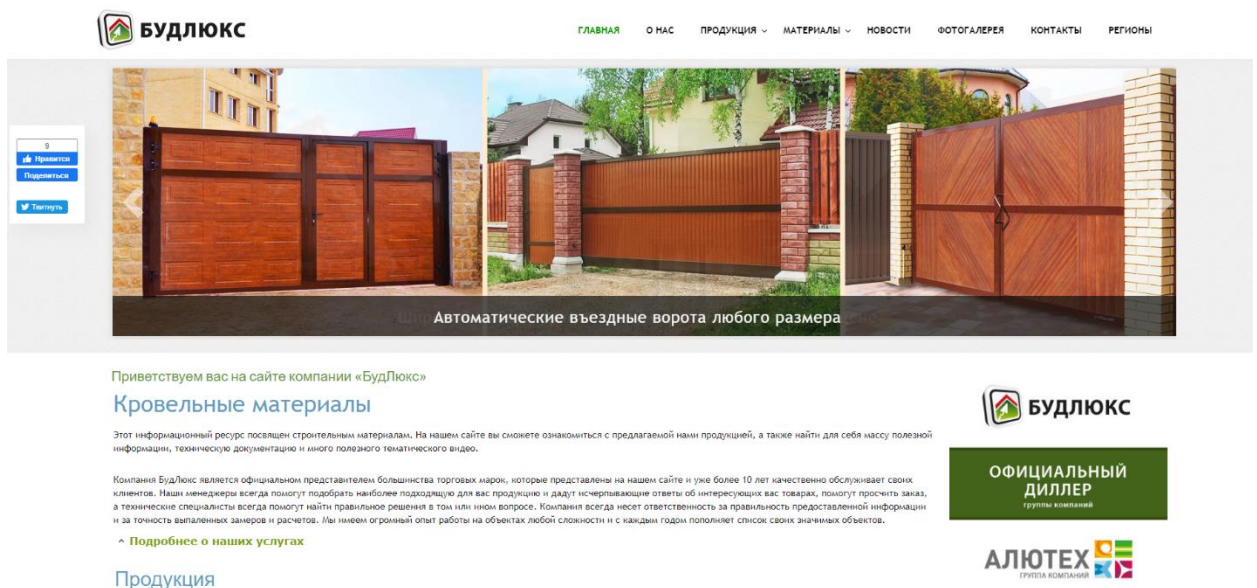


Рисунок 1.4 – Головна сторінка budlux.in.ua

Користувач відвідує цю сторінку з поставленою метою дізнатись про або придбати автоматичні ворота. Вже у першій частині сторінки він бачить багато важливої для себе інформації: що це за сайт, що продає власник, як виглядає товар (зображення на слайдері), посилання на інші розділи, на сторінки бізнес-партнерів та соціальні мережі – розрахунок йде на те, щоб швидко наділити користувача інформацією, яку він шукає. Зовнішнє оформлення при цьому страждає – відсутній фон, текст статті обривається, а у процесі завантаження з’їжджає вниз, щоб звільнити місце для фотографій. Технічно цей сайт був максимально оптимізований – швидкодія на комп’ютерах та телефонах є головним пріоритетом.

Інший підхід використовується на сайті mir-vorot.com.ua

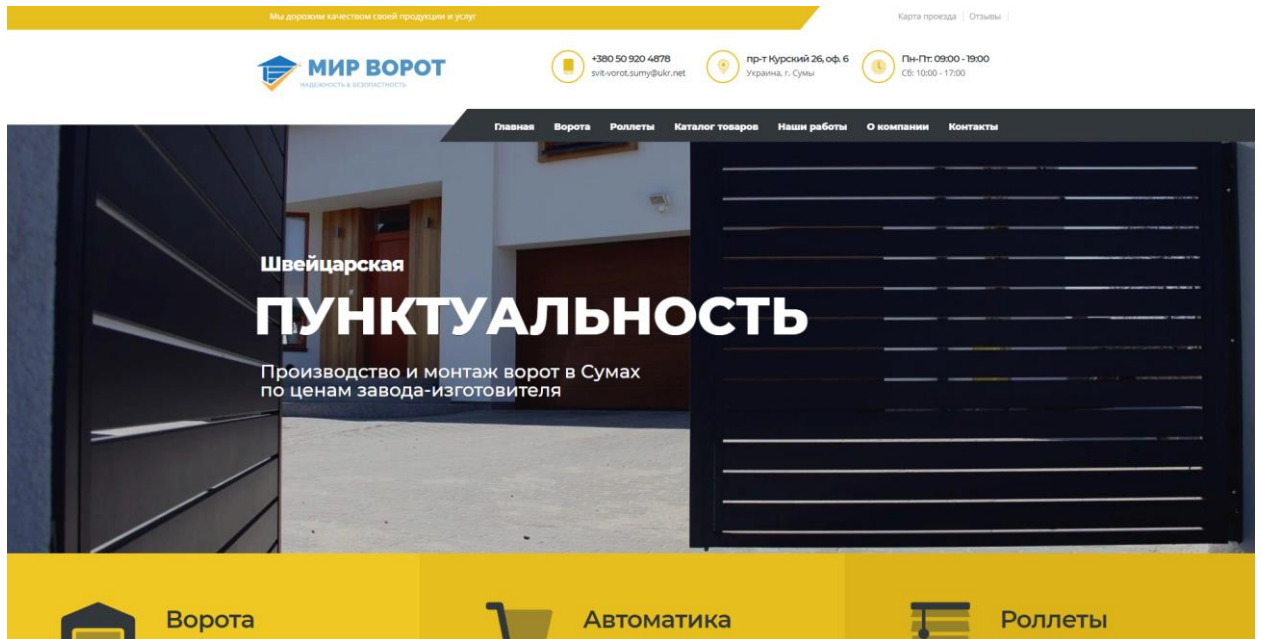


Рисунок 1.5 – Головна сторінка mir-vorot.com.ua

Дизайн цього сайту виглядає дружнім до користувача, перш за все помітне зображення товару та контактні дані власника – щоб знайти іншу інформацію, доведеться прокручувати сторінку, у випадку якщо користувач зацікавлений. Багато уваги приділяється стилям, підбору кольорової гами, а також візуальній межі між різними секціями – головним пріоритетом є перше враження. Те, що цей підхід виявився більш вдалим, можна стверджувати, бо сайт mir-vorot.com.ua є третім у пошуковій системі за запитом “автоматичні ворота”. Це свідчить про те, що під час вибору між якістю та швидкістю потрібно враховувати мету, яку переслідує веб-додаток.

#### 1.4 Постановка задачі

Мета дипломної роботи – розробити сайт із фото-галереєю, що дозволяє зберігати, відображати та додавати нові зображення. Веб-додаток повинен мати властивість змінювати свій рівень оптимізації відповідно до потреб відвідувача.

Розробка додатку включає в себе наступні підзадачі:

- формування структури сайту
- кодування та розмітка окремих сторінок

- об'єднання сторінок між собою через посилання
- додавання механізму збору інформації про користувача
- додавання алгоритму аналізу інформації та видозміни сторінок
- тестування сайту та формування висновків з його результату

Для тестування обраного підходу кожній з функцій сайту буде присвячена окрема сторінка, що дозволить провести тестування без впливу зовнішніх чинників.

## 2 ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ

### 2.1 Технічні засоби

#### Локальний сервер

Розробка та відлагоджування сайту відбувається за допомогою локального сервера OpenServer Panel

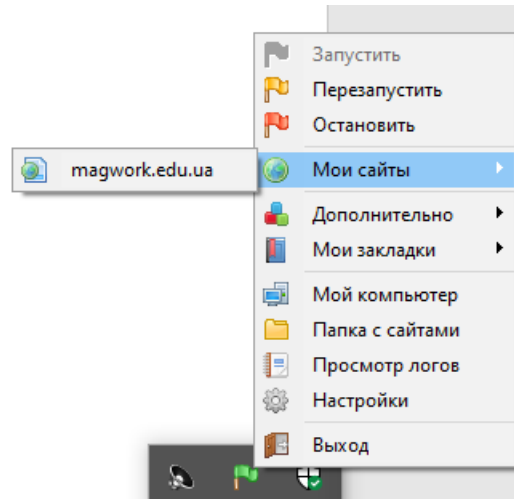


Рисунок 2.1 – Робоча область OS Panel

Альтернативами до нього є набори Denwer та XAMPP, що не були обрані з наступних причин:

- Denwer був оновлений до останньої версії у 2013 році і більше не супроводжується. При цьому він має ряд проблем. Однією з них є практична відсутність інтерфейсу – керування сервером здійснюється через прямий запуск командних файлів. Окрім цього, після закриття системними засобами деякі процеси залишаються активними і витрачають оперативну пам'ять, а також не дозволяють виконати запуск повторно.
- Перевагами XAMPP [Вікіпедія] є його багатоплатформовість, відкритий код, а також те, що серед своїх аналогів він найбільше схожий на справжній сервер – в даній роботі вони не зіграють ролі, бо розробка здійснювалась на платформі Windows x64, роботу з якою підтримує

OpenServer. При цьому XAMPP також є найскладнішим у налаштуванні. Для відображення кирилиці потребується налаштування MySQL.

## Текстовий редактор

Кодування здійснюється у текстовому редакторі Sublime Text – додаток є пропрієтарним, але ознайомитись з його роботою можна безкоштовно.

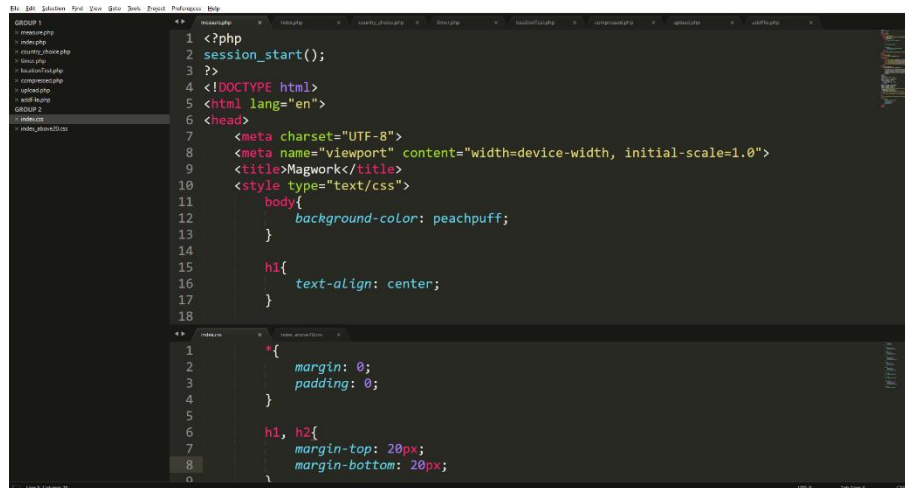


Рисунок 2.2 – Робоча область Sublime Text

Цей редактор має допоміжні функції, що стають у нагоді під час програмування: різні кольорові теми, перевірка та підсвічування синтаксису, перегляд файлів в окремих панелях, поділ їх на групи, перегляд директорій. За ключовою фразою можна здійснювати пошук як в межах одного файлу, так і у цілій директорії зі швидкою заміною. Є навіть можливість компілювати код на мові Python. Також можливе додавання нових функцій шляхом пошуку та установки розширень.

## Браузер

Для перегляду розробленого додатку зі сторони клієнта обрано два браузери – Google Chrome та Opera GX. Причини наступні:

- Для тестування потрібно як мінімум два браузери, бо деякі браузери мають властивість додавати до сторінки власні стилі, а тому один і той

же самий вміст може виглядати по різному за перегляду в різних браузерах.

- Google Chrome на даний момент є найпопулярнішим браузером у світі, тому якщо додаток гарно працює у Chrome – це гарантує, що проблем з браузером не виникне у 63,63% користувачів [Вікіпедія]. При відлагоджуванні додатка дозволяє скористуватись панеллю розробника, що дає доступ до завантажених ресурсів, збережених cookie-файлів, відображає код сторінки та консольні попередження.

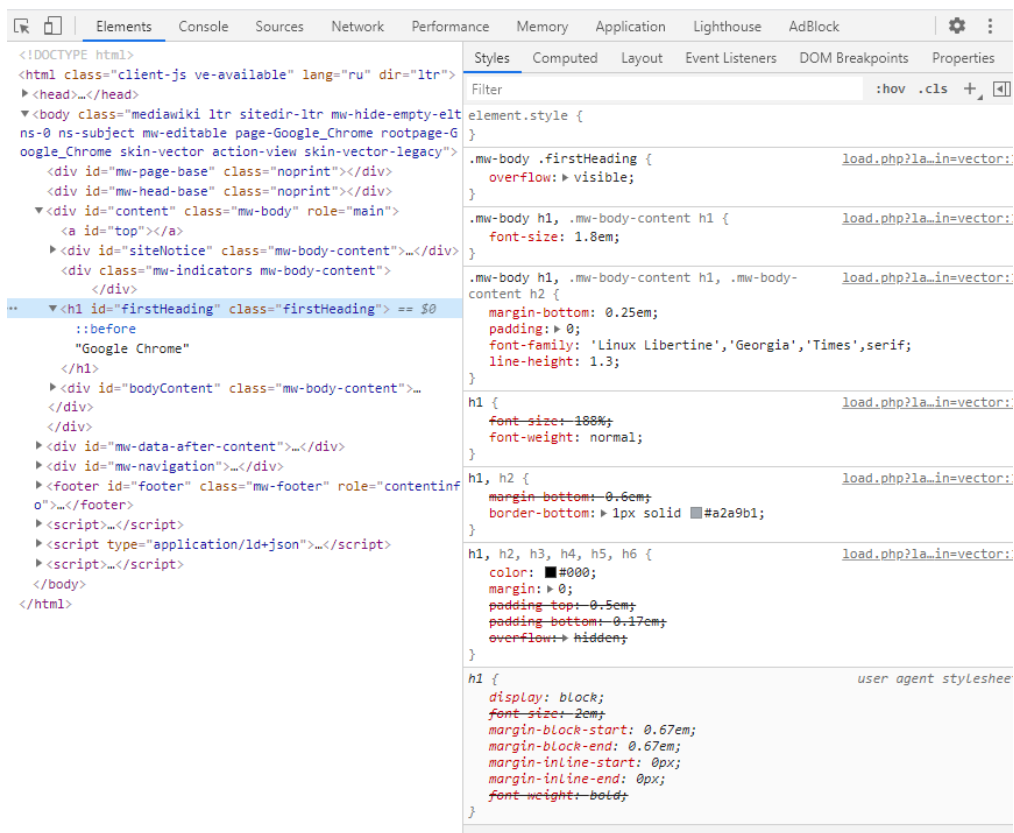


Рисунок 2.3 – Панель розробника Google Chrome

- Opera GX – це оновлена версія іншого популярного браузера Opera. Він має вбудовані інструменти, що допомагають при тестуванні веб-додатків, зокрема VPN із вибором локації та обмеження виділених ресурсів процесора.



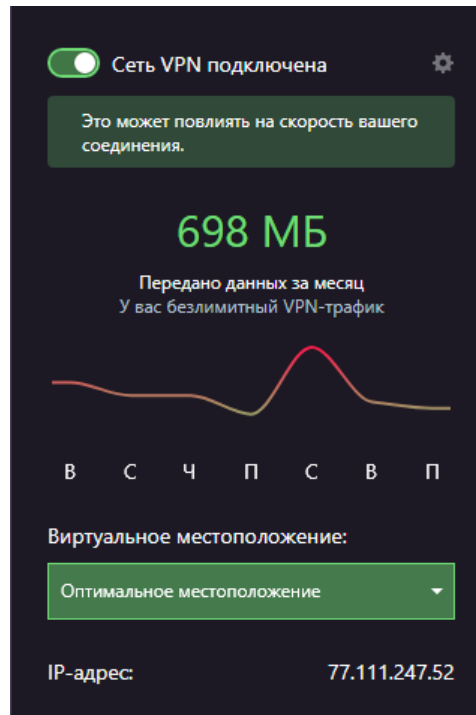


Рисунок 2.4 – Вбудований VPN в Opera GX

## 2.2 Мови програмування

Використано усі стандартні веб-технології. Фреймворки та шаблони відсутні, сторінки розмічені вручну із використанням вставок `php` для додаткового контролю над вмістом.

Стили `css` винесені в окремі файли та підключаються зовнішньо із допомогою `php` операторів.

JavaScript використовується в незначній мірі для реалізації автоматичного надсилання запитів користувачем.

## 2.3 Хостинг

В ході тестування програми потребується робота з різними IP-адресами - цього неможливо досягти на локальному сервері, який змушує комп'ютер звертатись сам до себе, тобто за адресою `127.0.0.1`. У зв'язку з цим для повноцінної перевірки працездатності додатку він розміщений на хостинг як реальний сайт і може бути відвіданий з довільного пристрою.

Обрати хостинг можна із великої кількості альтернатив, ось деякі з них:

- [ukraine.com.ua](http://ukraine.com.ua) – пропрієтарний сайт, на якому можна зареєструвати власний домен та використовувати хостинг із виділеними серверами, панеллю управління, файловим менеджером, налаштованою базою даних, захистом від вірусів та хакерських атак, онлайн підтримкою.
- [zzz.com.ua](http://zzz.com.ua) – надає доступ до файлового менеджера та бази даних із пробним періодом в 1 рік, за який можна ознайомитись з усіма функціями сайту безкоштовно, після чого обов’язково потрібно обрати інший тариф. Можливий перенос сайту з іншого хостингу.
- [sprinthost.ru](http://sprinthost.ru) – надає кожному користувачу хостинг для 1 сайту безкоштовно на необмежений час. Серед недоліків – відсутність технічної підтримки, обов’язкове регулярне відвідування панелі управління та обмежений об’єм файлової системи, які виправляються переходом на інший тариф. Обраний для використання в дипломному проєкті, бо має простий та швидкий доступ із прив’язкою облікового запису до пошти.

## 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

### 3.1 Інформаційна модель

Використання мови php дозволяє вносити зміни у структуру додатку вже під час його виконання, але попередньо потрібно визначити критерії, за якими перевіряється необхідність у змінах. Під час використання сайту відвідувачем виникає наступна проблема: веб-сайт один і той самий, а користувачі різні, отже їх потреби також можуть відрізнятись, тому баланс між швидкістю та якістю не може бути ідеальним для усіх одразу, за умови що він фіксований. Надсилаючи запит на перегляд сторінки відвідувач надає про себе інформацію, яка включає його IP-адресу. Найбільш важливим параметром є його швидкість інтернет, яка не вказується напрямую, але може бути визначена практично. Порівнюючи якість зв'язку, яку має користувач, з тією, яку очікує від нього сервер, можна виділити чотири загальних сценарії:

- висока якість, високі вимоги – користувач швидко отримує сторінку в повному об'ємі
- низька якість, низькі вимоги – користувач швидко отримує найважливіші частини, не дивлячись на поганий зв'язок
- висока якість, низькі вимоги – користувач отримує урізану версію сторінки, на яку очікував
- низька якість, високі вимоги – завантаження сторінки виконується довгий час або зупиняється через помилку

Два останніх випадки є небажаними, але зі зростанням аудиторії веб-сайту деяка частина користувачів неминуче зустрінеться з ними. Щоб уникнути цього, потрібно знати якість зв'язку із користувачем наперед. Користувачу надсилається невеликий об'єм текстової інформації (її обсяг повинен бути відомий) та вимірюється час, за який це було виконано. Їх співвідношення становитиме швидкість інтернет на даний момент. Таким підходом

користуються сторінки, присвячені повідомити відвідувача про його швидкість інтернет, наприклад [www.speedtest.net](http://www.speedtest.net) або [www.speedtestnet.io](http://www.speedtestnet.io).

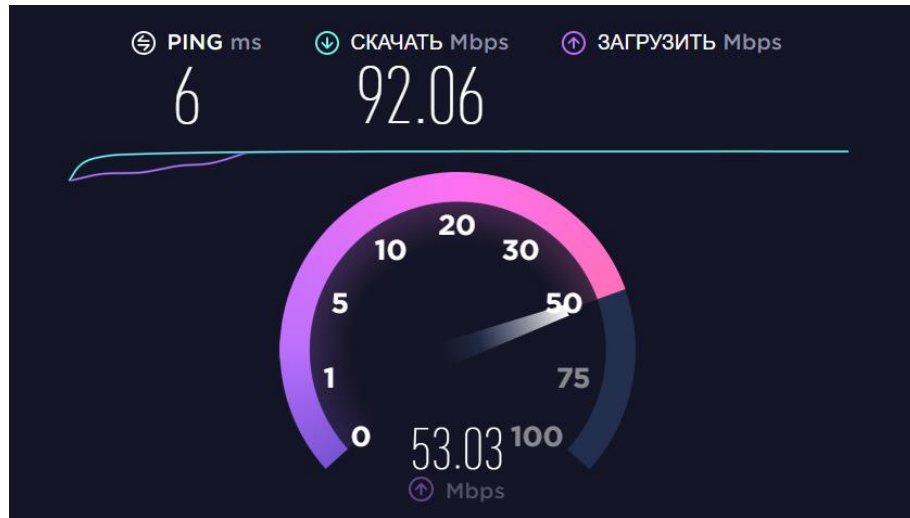


Рисунок 3.1 – Вимірювання швидкості завантажування на speedtest.net

Для досягнення точного результату такі тести проводяться протягом деякого часу і показують середній результат, оскільки стан підключення є мінливим. В додатку, що розробляється, ця функція є вторинною і затримує користувача. Виміряти величину потрібно швидко, тому передається незначний об'єм інформації (до декількох мегабайтів) – це призводить до ненадійності результату. Щоб отримати більш стабільний показник, використовується середнє значення між фактично виміряною та середньою регіональною (далі - очікуваною) швидкістю. Визначити країну користувача можна, скориставшись API зі спеціалізованих сайтів, наприклад, 2ip.ua або ipinfo.io. Зазвичай після звернення такі сервіси повертають відповідь у форматі JSON, в якій надається інформація про розташування відвідувача та провайдера, послугами якого він користується.

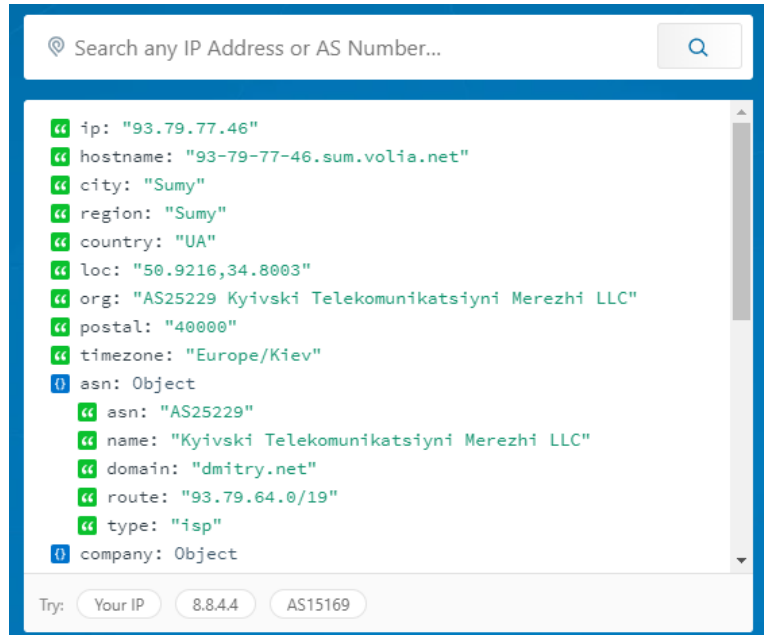


Рисунок 3.2 – Приклад відповіді від геолокаційного API

Найбільш цікавим для визначення очікуваної швидкості є пункт “country”. Згадані сайти, що вимірюють швидкість з’єднання, надають публічну статистику, яка демонструє середню – а отже найбільш вірогідну швидкість для кожної країни. Ці результати подані у табличному вигляді [10].

Rank ↕	Country/Territory ↕	Average connection speed (Mbit/s) ↕
1	Singapore	241.10
2	Bahrain	222.92
3	Thailand	213.14
4	Monaco	206.66
5	Switzerland	192.13
6	Denmark	191.40
7	Romania	188.08
8	Andorra	182.17
9	France	177.25
10	Hungary	175.01
11	United States	170.88
12	South Korea	168.26
13	Spain	166.62

Рисунок 3.3 – Список країн, відсортованих за швидкістю з’єднання на комп’ютері за версією Speedtest.net

Бездротові мережі (Wi-Fi або мобільний інтернет) мають гірші показники через послаблення сигналу з відстанню та перешкодами, тому наявні окремі табличні дані для комп'ютерних та для бездротових мереж. Перші використовуються на персональних комп'ютерах, що мають 8P8C роз'єми і утворюють мережі Ethernet[11]. Wi-Fi та супутниковий зв'язок використовуються здебільшого на мобільних телефонах та інших пристроях, які не мають можливості прямого підключення. Знаючи це, можна передбачити яким із двох типів мереж користується відвідувач. Тип пристрою [12] також не передається до сервера напряму, але його можна визначити за назвою версії використаного браузеру – для чого потрібно було б передбачити використання усіх популярних браузерів у світі. Інший спосіб – визначити ширину та довжину пристрою, порівнявши їх з відомими розмірами пристроїв. Більшість телефонів має viewport до 850 пікселів у ширину, отже пристрій з шириною меншою за 900 можна вважати мобільним, а більшою за 1000 – комп'ютером. Відповідно для визначення швидкості зв'язку, яку очікуватиме сервер, використовуються три параметри: практично виміряна швидкість, географічне розташування та тип пристрою.

Після вимірювання очікуваної швидкості чисельний результат має бути збережений для можливості використовувати його повторно на поточній та інших сторінках сайту. Є декілька можливих варіантів: cookie, session і local storage.

- cookie – файли з довільною інформацією, які відправляються сервером на комп'ютер користувача і зберігаються на ньому деякий час (якщо він вказаний). При повторному зверненні сайт може використовувати інформацію з такого файлу для того, щоб “згадати” свого відвідувача. Зараз використання cookie формально потребує згоди відвідувачів та виходить із вжитку через небезпеку викрадання інформації з них. Ще одним недоліком є те, що відправляються вони разом із заголовками сторінки, а тому не можуть бути відправлені після того, як сторінка вже

почала відображатись – значення все ще можна змінити, але воно не буде видимим глобально для усього сайту.

- `session` – масив змінних мови `php`, на відміну від `cookie` зберігається не на комп'ютері користувача, а на сервері (і тому не може бути прочитаний зловмисником), його значення можуть бути змінені у процесі роботи, видимі на усьому домені (якщо не вказано інакше). Сесія триває 15 хвилин або поки користувач не закриє свій браузер повністю, може бути програмно розпочата та завершена – завдяки такій гнучкості обрана для використання в додатку.
- `local storage` – масив мови `javascript`, як і у випадку з `cookie`, зберігає дані у сховищі браузера. Для обміну даними із сервером потребує додаткової інтеграції з `php`, окрім того підтримується не усіма браузерами.

Після того, як параметри збережені тим чи іншим чином, на їх основі до кожного елемента сторінки з відносно високою вагою (зображення, аудіо, відео, скрипти, стилі) виставляється умова: швидкість має бути вища за число `n`, інакше елемент не відобразатиметься. Значення `n` є індивідуальним для різних підключень/блоків та визначається програмістом.

## 3.2 Програмна реалізація

### Головна сторінка

Формується файлом `index.php` з метою показати користувачу галерею та надати посилання для доступу до інших відділів сайту, а також проінформувати його про аналітичний метод, що використовується на сайті. Він починається з наступного фрагменту коду:

```
<?php
session_start();
?>
```

Цей код запускає сесію, даючи можливість зберігати дані та передавати між сторінками.

Далі йде розділ `head`, в якому вказується кодування (для відображення символів кирилиці), відношення розмірів вікна до ширини пристрою та заголовок сторінки, який буде написаний у вкладинці браузера.

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<title>Magwork</title>
```

За ними слідує рядок підключення зовнішнього файлу зі стилями.

```
<link rel="stylesheet" type="text/css" href="./css/index.css">
```

Ці стилі допомагають сторінці зберігати свою форму і є потрібним мінімумом для коректної роботи сторінки, тому вони підключаються безумовно та в першу чергу, щоб уникнути зміщення макету.

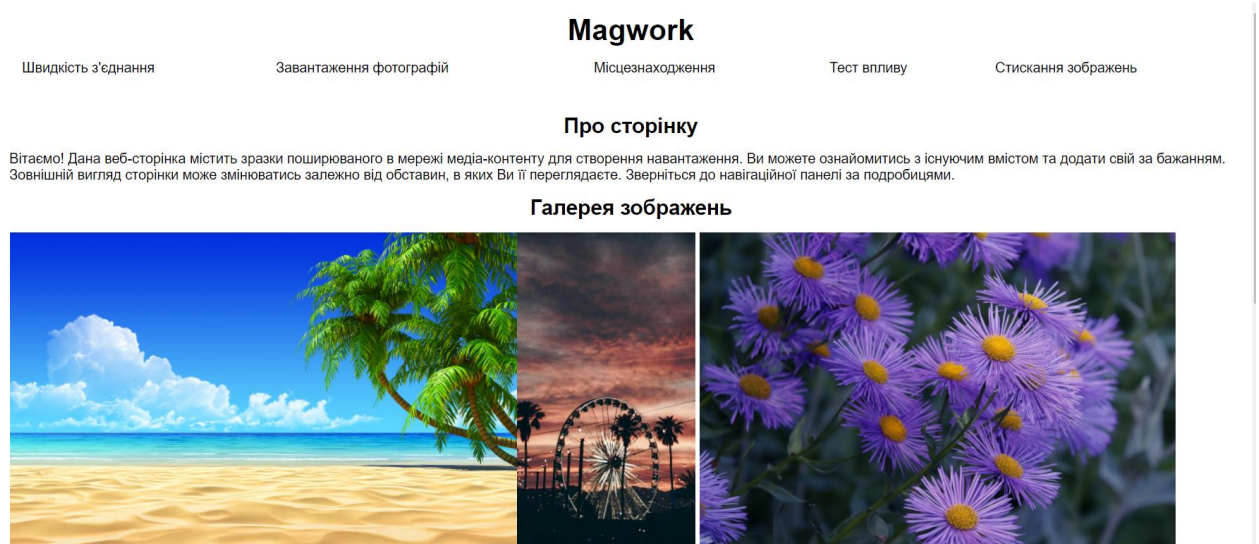


Рисунок 3.4 – Мінімізований дизайн сторінки

Наступний код [15] запам'ятовує системний час та циклічно друкує крапки до тих пір, поки загальний їх обсяг не становитиме 1024 кілобайти.



Крапки поміщені між дужками коментаря, тому не відображаються та не впливають на сторінку.

```
<?php
$size=1024;
echo "<!--";
flush();
$clock = explode(" ",microtime());
$beginning = $clock[0] + $clock[1];
for($x=0;$x<$size;$x++){
    echo str_pad('', 1024, '.');
    flush();
}
```

Після цього час вимірюється повторно і обсяг даних ділиться на різницю між часом початку та кінця.

```
$clock = explode(" ",microtime());
$end = $clock[0] + $clock[1];
$difference = $end - $beginning;
$speed=round($size / $difference, 3);
echo "->";
```

Отримане число зберігається у наперед відкритій сесії

```
$_SESSION['pagespeed']=$speed;
```

Після цього з ним можна працювати в довільний момент, наприклад підключити ще один файл стилів.

Наступний код перевіряє, чи задана у сесії змінна “очікуваної швидкості”. Це потрібно на випадок, коли країна користувача не була розпізнана (наприклад, коли використовується локальний сервер і справжня адреса невідома). Якщо змінна існує, відбувається перевірка, чи її значення перевищує певний поріг (визначений програмістом для конкретно цього файлу згідно за обсягом і важливістю). Якщо ж змінна відсутня, то замість неї використовується фактична швидкість напряму.

```

if($_SESSION['expected_speed']>20000||$_SESSION['pagespeed']>200
00){
echo "<link rel='stylesheet' href='/css/index_above20.css'>";
}

```

Ці стилі є вторинними за важливістю, але будуть підключені, якщо виміряна швидкість перевищує задане число, в даному випадку – 20000 Кбайт в секунду або приблизно 20Мб/с. Обраний поріг може бути змінений залежно від вмісту файлу та його важливості. Таким чином 3 рядки коду дозволяють у потрібний момент зберегти ресурси сторінки, відключивши файл стилів, який на практиці може досягати обсягу в 20 і більше тисяч рядків.



Рисунок 3.5 – Вигляд сторінки із додатковими стилями

Далі приведений код, який грає участь у вимірюванні очікуваної швидкості. Перш за все перевіряється, чи є у сесії задана ширина екрану – якщо так, то подальші дії не є необхідними

```

if(isset($_SESSION['screen_width'])) {
}

```

Якщо в сесії цей параметр не заданий, але він переданий у запиті користувача, то значення зберігається до сесії.

```
else if(isset($_REQUEST['width'])) {
    $_SESSION['screen_width'] = $_REQUEST['width'];
}
```

І нарешті якщо параметру немає ні в сесії, ні в запиті, то підключається фрагмент коду мовою javascript, який додає потрібні значення до запиту.

```
else {
    echo '<script type="text/javascript">window.location = "' .
$_SERVER['PHP_SELF'] .
'?width='+screen.width+'&height='+screen.height;</script>';
}
```

Ця операція потрібна, бо сервер не може знати, з якого монітору був відправлений запит на підключення, а отже користувач має повідомити це сам, що можна автоматизувати із допомогою javascript.

У розділі body знаходиться інформація для відображення користувачу, така як блок header із назвою сайту у вигляді стилізованого заголовку. Він також слугує посиланням, щоб повернутись до головної сторінки з будь-якого місця на сайті.

```
<div class="header">
    <h1><a href="/">Magwork</a></h1>
</div>
```

Список посилань на інші сторінки, витягнутий в один рядок та позбавлений нумерації для того, щоб його пункти виглядали як окремі блоки.

```

<div class="nav">
  <ul>
    <li><a href="measure.php">Швидкість з'єднання</a></li>
    <li><a href="addFile.php">Завантаження
фотографій</a></li>
    <li><a href="locationTest.php">Місцезнаходження</a></li>
    <li><a href="timer.php">Тест впливу</a></li>
    <li><a href="compressed.php">Стискання
зображень</a></li>
  </ul>
</div>

```

Ці елементи складають навігацію, і тому присутні на кожній сторінці, щоб не втрачати можливості довільно просуватись по сайту. Блок із класом `container` містить дані виключно для поточної сторінки, такі як привітання для відвідувача.

```

<h2 class="center section_title">Про сторінку</h2>
  <div class="section white">
    Вітаємо! Дана веб-сторінка містить зразки поширюваного
в мережі медіа-контенту для створення навантаження. Ви можете
ознайомитись з існуючим вмістом та додати свій за бажанням.
Зовнішній вигляд сторінки може змінюватись залежно від обставин,
в яких Ви її переглядаєте. Зверніться до навігаційної панелі за
подробницями.
  </div>

```

Фрагмент `php`-коду, який відкриває задану директорію і зчитує назву кожного файлу у ній

```

<?php
    $folder = "./images";
    $allowed=array("jpg", "png", "gif");
    $file_parts = array();
    $extension="";
    $title="";
    $i=0;
    $handler = @opendir($folder) or die("Не вдалось відкрити
папку");
    while ($file_name = readdir($handler))
    {
        if($file_name=="." || $file_name == "..") continue;
        $file_parts = explode(".", $file_name);
        $extension = strtolower(array_pop($file_parts));

        Перевіряється, чи належить розширення файлу до списку дозволених
(формати фотографій), і якщо файл дійсно є зображенням, посилання до нього
вноситься в окремий тег.
        if(in_array($extension,$allowed))
        {echo '';
            $i++;}
    }

```

Таким чином усі фотографії у директорії відображаються на сторінці, їх кількість довільна. Наведений далі код потрібен для демонстрації повноцінної роботи сторінки. Швидкість у кілобайтах переводиться до потрібної одиниці вимірювання та друкується за умови, якщо вона була визначена.

```

if(isset($_SESSION['pagespeed'])){ echo "Виміряна швидкість:
".($_SESSION['pagespeed']/1000).'Mb/s</p>';
}else{
    echo "Швидкість невідома</p>";
}

```

Таким же чином виводиться країна перебування відвідувача.

```

if(isset($_SESSION['country'])){
    echo "<br/>Виявлена країна:
".$_SESSION['country'].'</p>';
}else{
    echo "<br/>Країна невідома</p>";
}

```

Після друку ширини екрану до змінних сесії додається нове значення – тип пристрою: комп'ютер або телефон, залежно від того, чи перевищує ширина мінімальне для монітора значення.

```

if(isset($_SESSION['screen_width'])){
echo "<br/>Ширина екрану: ".$_SESSION['screen_width'].'</p>';
if($_SESSION['screen_width']<900){
    $_SESSION['device']="phone";
}else{
    $_SESSION['device']="PC";
}}

```

Воно виводиться разом з іншими даними

```

if(isset($_SESSION['device'])){
    echo "<br/>Розпізнаний пристрій: ";
    if($_SESSION['device']=="phone"){
        echo "телефон";
    }else{
        echo "комп'ютер";
    }
}

```

## Розділ вимірювання

Має на меті повідомити користувача, що його з'єднання перевіряється та продемонструвати результат тестування. Як і інші, має навігацію у вигляді заголовка та списку посилань. Відрізняється привітанням для користувача.

```
<h1>Вимірювання швидкості інтернету</h1>
```

```
<p>
```

```
    Ми визначимо час, за який Ваш комп'ютер здатний
    завантажити 1024 кілобайти текстової інформації. Результат буде
    збережений у сесії та використаний для визначення оптимальних
    параметрів інших сторінок. Зверніть увагу, що вимірний показник
    є актуальним на момент вимірювання та постійно зазнає змін до
    деякого ступеня.
```

```
</p>
```

```
    Файли стилів, що підключаються, також інші.
```

```
    В цьому випадку базових стилів немає і йде вибір між двома
    підключеннями для значення вище або нижче заданого порогу.
```

```
if($_SESSION['pagespeed']>20000){
    echo "<link rel='stylesheet' href='css/above20.css'>";
}else{
    echo "<link rel='stylesheet' href='css/below20.css'>";
}
```

Вони містять протилежні стилі для окремого блоку, який приймає зелений колір, коли швидкість з'єднання перевищує задану, або червоний у іншому випадку.



Рисунок 3.6-3.7 – Вигляд секції при високому та низькому результаті

## Місцезнаходження

Ця сторінка має мету визначити адресу та географічні дані, виводячи частину з них на екран. Адреса користувача може бути передана на сервер у різному вигляді. Функція [14] по чергово перевіряє три змінні та формує адресу з першого непорожнього значення, повертаючи її як результат.

```
function learnIp() {
    $variant= [
        'HTTP_CLIENT_IP',
        'HTTP_X_FORWARDED_FOR',
        'REMOTE_ADDR'
    ];
    foreach ($variant as $k) {
        if (!empty($_SERVER[$k])) {
            $address = trim(end(explode(',', $_SERVER[$k])));
            if (filter_var($address, FILTER_VALIDATE_IP)) {
                return $address;
            }
        }
    }
}
```

Ця адреса передається у запиті до сайту [ipinfo.io](http://ipinfo.io), щоб отримати звіт з даних про вказаний хост



```
function ip_details($ip) {
    $json = file_get_contents("http://ipinfo.io/{$ip}");
    $details = json_decode($json);
    return $details;
}
```

Після того, як дані звіту (або повідомлення про їх відсутність) виведені на екран, до файлу підключається окремий скрипт, що відповідає за вибір країни.

```
$details = ip_details($ip);
if(isset($details->city)){
    echo '<p>Місто:<b>'.$details->city.'</b></p>';
    echo '<p>Країна:<b>'.$details->country.'</b></p>';
    echo '<p>Провайдер:<b>'.$details->org.'</b></p>';
    echo '<p>Назва хоста:<b>'.$details->hostname.'</b></p>';
    $_SESSION['country']=$details->country;
    /*налаштування для різних країн*/
    require 'country_choice.php';
}
```

Він містить два “словники” – масиви попарних значень. Перший ставить середню швидкість країни (на комп’ютерах) у відповідність до її скороченої назви .

```
$pc_speeds = array(
    'UA' => 63000,
    'CA' => 157800,
    'RO' => 188000,
    'NL' => 133000,
    'JP' => 148010,
    'SE' => 192130
);
```

Другий робить те саме, але для мобільних пристроїв, яким відповідають менші значення.

```

$mobile_speeds = array(
    'UA' => 14000,
    'CA' => 59600,
    'RO' => 21400,
    'NL' => 54800,
    'JP' => 49300,
    'SE' => 42800
);

```

Обидва списки можуть бути доповнені, щоб містити у собі усі країни, користувачі з яких очікуються на сайті. Важливо, щоб кожна країна була в обох списках.

За словниками йде код, який бере значення швидкості з одного із словників, залежно від того, чи визначений пристрій відвідувача як телефон або комп'ютер. Обране значення додається до виміряного фактично і їх середнє значення (половина суми) зберігається як швидкість, яку сервер очікуватиме від даного клієнта. Це значення використовується у попередніх скриптах за рахунок того, що зберігається на інших сторінках протягом сесії.

```

if(isset($_SESSION['device']) && $_SESSION['device']=='mobile'){
    $_SESSION['expected_speed']=
    ($_SESSION['pagespeed']+$mobile_speeds[$_details->country])/2;
}else{
    $_SESSION['expected_speed']=(($_SESSION['pagespeed']+$pc_speeds[$_
details->country])/2;
}

```

Оператор switch-case може виконати додаткові дії на основі того, яка країна була розпізнана. Наступний фрагмент зберігає у сесії значення національних (та контрастних до них) кольорів, щоб стилізувати сторінку

```

switch($details->country) {
    case 'UA':
        $_SESSION['country_color1']="#136bcc";
        $_SESSION['country_color2']="#fed501";
        $_SESSION['country_color3']="";
        $_SESSION['back_font']="white";
        $_SESSION['front_font']="black";
        break;

```

...



Рисунок 3.8-3.9 – Вигляд сторінки для адреси з Нідерландів або невідомої

### Тест впливу

До цього розділу не застосовані ніякі заходи оптимізації. В ньому демонструється вплив кількості даних на час завантаження. Як і на головній сторінці, код починається із запам'ятовування часу, після чого слідує велика кількість підключень одного й того самого фото. Чим їх більше – тим сильнішим буде вплив .

```

<?php
    $start=microtime(true);
    ?>

    <p>Перегляньте низ сторінки, щоб дізнатись час завантаження її
вмісту.</p>
    
    
    
    
    ...

```

За показниками таймера можна помітити, що стискання зображень вже на сторінці шляхом задання фіксованих розмірів не дає ніякого прискорення, бо перед додаванням стилів зображення все одно завантажуються в оригінальному розмірі, а отже додаткові стилі навпаки лише сповільнюють процес.

```

<style type="text/css">
    body img{
        /*max-width: 400px;
        max-height: 200px;*/
    }
</style>

```

### **Завантаження фотографій**

Цей розділ використовується для наповнення галереї. Усі присутні в ній файли завантажуються через публічно доступну форму, що дозволяє швидко наповнити її після появи перших клієнтів.

```
<div class="section white">
```

```
    Вітаємо! Виберіть файл зі свого комп'ютера та відправте його для того, щоб додати до існуючої галереї. Завантажувати можна лише зображення розміром до 1024x768. Якщо Ви передумали, натисніть на заголовок, щоб повернутись до головної сторінки.
```

```
    <br><br>
```

```
    <form          action="upload.php"          method="post"
enctype="multipart/form-data">
```

```
        <input type="file" name="image">
```

```
        <button type="submit">Завантажити</button>
```

```
    </form>
```

```
</div>
```

### 3.3 Результати тестування

Розроблений додаток був завантажений на хостинг та перевірений шляхом повторних відвідувань з використанням різних ір-адрес. Сайт успішно виконує операції з відображення існуючих та додавання нових даних. Встановлено, що інформація про зв'язок та знаходження користувача збирається, аналізується та відображається на сторінках згідно очікувань. При зниженні швидкості зв'язку (наприклад, через використання VPN) сторінки сайту змінюють свій вигляд, зупиняючи використання ресурсів, відмічених як додаткові. Також коректно обробляються ситуації, коли частину даних про відвідувача встановити неможливо.

## ВИСНОВКИ

В ході виконання дипломної роботи було розроблено веб-додаток - фото-галерею, що дозволяє зберігати, відображати та додавати нові зображення. Додаток має властивість змінювати свій рівень оптимізації відповідно до потреб відвідувача.

Під час розробки виконано наступні підзадачі:

- формування структури сайту
- кодування та розмітка окремих сторінок
- об'єднання сторінок між собою через посилання
- додавання механізму збору інформації про користувача
- додавання алгоритму аналізу інформації та видозміни сторінок
- тестування сайту та формування висновків з його результату

Для тестування обраного підходу кожній з функцій сайту було присвячено окрему сторінку, в результаті чого тестування було проведено без впливу зовнішніх чинників.

## СПИСОК ЛІТЕРАТУРИ

1. Егорова И. Н., Рыгина В. Е. Исследование методов оптимизации структуры web-страниц // ВЕЖПТ. 2010. №2 (47). URL: <https://cyberleninka.ru/article/n/issledovanie-metodov-optimizatsii-struktury-web-stranits>
2. Баночкин Павел Иванович, Вичугов Владимир Николаевич Система семантической оптимизации содержимого веб-сайтов на основе пользовательских предпочтений // Известия ТПУ. 2012. №5. URL: <https://cyberleninka.ru/article/n/sistema-semanticheskoy-optimizatsii-soderzhimogo-veb-saytov-na-osnove-polzovatel'skih-predpochteniy>
3. Деарт В. Ю., Кожухов И. С. Влияние современных способов оптимизации загрузки Интернет страниц в браузерах на параметры качества обслуживания HTTP-трафика // Т-Comm. 2012. №7. URL: <https://cyberleninka.ru/article/n/vliyanie-sovremennyh-sposobov-optimizatsii-zagruzki-internet-stranits-v-brauzerah-na-parametry-kachestva-obsluzhivaniya-http-trafika>
4. Ботыгин Игорь Александрович, Каликин Кирилл Александрович Исследование методов увеличения производительности WEB-приложений // Известия ТПУ. 2008. №5. URL: <https://cyberleninka.ru/article/n/issledovanie-metodov-uvelicheniya-proizvoditelnosti-web-prilozheniy>
5. Ash T., Ginty M., Page R. Landing page optimization: The definitive guide to testing and tuning for conversions. – John Wiley & Sons, 2012.
6. [Check Your Internet Speed with Speed Test](#)". *speedtestnet.io*. Retrieved 12 March 2021.
7. Manhas J. A study of factors affecting websites page loading speed for efficient web performance //International Journal of Computer Sciences and Engineering. – 2013. – Т. 1. – №. 3. – С. 32-35.
8. D. Guse, S. Schuck, O. Hohlfeld, A. Raake and S. Möller, "Subjective quality of webpage loading: The impact of delayed and missing elements on quality ratings and task completion time," 2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX), 2015, pp. 1-6, doi: 10.1109/QoMEX.2015.7148094.

9. Пошукова система - Вікіпедія [Електронний ресурс] – Режим доступу за посиланням:

[https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D1%88%D1%83%D0%BA%D0%BE%D0%B2%D0%B0\\_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0](https://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D1%88%D1%83%D0%BA%D0%BE%D0%B2%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0)

10. List of countries by Internet connection speeds – Вікіпедія [Електронний ресурс] – Режим доступу за посиланням:

[https://en.wikipedia.org/wiki/List\\_of\\_countries\\_by\\_Internet\\_connection\\_speeds](https://en.wikipedia.org/wiki/List_of_countries_by_Internet_connection_speeds)

11. 802.3-2015 IEEE Standard for Internet – URL:

[https://standards.ieee.org/standard/802\\_3-2015.html](https://standards.ieee.org/standard/802_3-2015.html)

12. Editors PC Magazine. "Definition of: tablet computer". PC Magazine. Archived from the original on July 16, 2010

13. How to upload image in Laravel – [Електронний ресурс] – URL:

<https://helperbyte.com/questions/470841/how-to-upload-image-in-laravel>

14. Getting the screen resolution using PHP - [Електронний ресурс] – URL:

<https://stackoverflow.com/questions/1504459/getting-the-screen-resolution-using-php>

15. How to detect internet speed in PHP – [Електронний ресурс] – URL:

<https://askpython.dev/questions/174483/how-to-detect-internet-speed-in-php/>

16. Haoyu Wang, Mengxin Liu, Yao Guo, and Xiangqun Chen. 2014. Similarity-based web browser optimization. In Proceedings of the 23rd international conference on World wide web (WWW '14). Association for Computing Machinery, New York, NY, USA, 575–584. DOI:<https://doi.org/10.1145/2566486.2567971>



## ДОДАТОК – ЛІСТИНГ ФАЙЛІВ

### **index.php**

```
<?php
session_start();
//$mark=microtime(true);
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Magwork</title>

    <!-- Базові стилі -->
    <link                rel="stylesheet"                type="text/css"
href="./css/index.css">

    <!-- Вимірювання швидкості -->
<?php
$size=1024;
echo "<!--";
flush();
$clock = explode(" ",microtime());
$beginning = $clock[0] + $clock[1];
for($x=0;$x<$size;$x++){
    echo str_pad('', 1024, '.');
    flush();
}

$clock = explode(" ",microtime());
```

```

$end = $clock[0] + $clock[1];
$difference = $end - $beginning;
$speed=round($size / $difference, 3);
echo "->";
$_SESSION['pagespeed']=$speed;
if($_SESSION['expected_speed']>18000||$_SESSION['pagespeed']>180
00){
echo "<link rel='stylesheet' href='/css/index_above20.css'>";
}
?>
<!-- ^ Якщо результат високий, підключаються додаткові стилі -->

<!-- Перевірка/передача ширини та висоти -->
<?php
if(isset($_SESSION['screen_width'])                                AND
isset($_SESSION['screen_height']))){

}          else          if(isset($_REQUEST['width'])            AND
isset($_REQUEST['height'])) {
    $_SESSION['screen_width'] = $_REQUEST['width'];
    $_SESSION['screen_height'] = $_REQUEST['height'];
    header('Location: ' . $_SERVER['PHP_SELF']);
} else {
    echo '<script type="text/javascript">window.location = "' .
$_SERVER['PHP_SELF']
'?width="'+screen.width+"&height="'+screen.height;</script>';
}
?>

</head>
<body>
<div class="header">

```

```

        <h1><a href="/">Magwork</a></h1>
</div>
<div class="nav">
    <ul>
        <li><a href="measure.php">Швидкість з'єднання</a></li>
        <li><a href="addFile.php">Завантаження
фотографій</a></li>
        <li><a href="locationTest.php">Місцезнаходження</a></li>
        <li><a href="timer.php">Тест впливу</a></li>
        <li><a href="compressed.php">Стискання
зображень</a></li>
    </ul>
</div>

<div class="container">

    <h2 class="center section_title">Про сторінку</h2>
    <div class="section white">
        Вітаємо! Дана веб-сторінка містить зразки поширюваного
в мережі медіа-контенту для створення навантаження. Ви можете
ознайомитись з існуючим вмістом та додати свій за бажанням.
Зовнішній вигляд сторінки може змінюватись залежно від обставин,
в яких Ви її переглядаєте. Зверніться до навігаційної панелі за
подробностями.
    </div>

    <div class="gallery">
        <h2 class="center section_title">Галерея зображень</h2>
        <?php
$directory = "./images";
$allowed_types=array("jpg", "png", "gif");

```

```

$file_parts = array();
    $ext="";
    $title="";
    $i=0;
    $dir_handle = @opendir($directory) or die("Ошибка при
открытии папки !!!");
while ($file = readdir($dir_handle))
    {
    if($file=="." || $file == "..") continue;
    $file_parts = explode(".", $file);
    $ext = strtolower(array_pop($file_parts));

    if(in_array($ext,$allowed_types))
    {
    echo '';
    $i++;
    }

    }
closedir($dir_handle);
?>
</div>

<h2 class="center section_title">Про користувача</h2>
<div class="section white">
    <p>
        <!-- Вивід показників -->
    <?php
        if(isset($_SESSION['pagespeed'])) {

```

```

        echo                "Виміряна                швидкість:
".($_SESSION['pagespeed']/1000).'Mb/s</p>';
    }else{
        echo "Швидкість невідома</p>";
    }

    if(isset($_SESSION['country'])){
        echo                "<br/>Виявлена                країна:
".$_SESSION['country'].'</p>';
    }else{
        echo "<br/>Країна невідома</p>";
    }

    if(isset($_SESSION['screen_width'])){
        echo                "<br/>Ширина                екрану:
".$_SESSION['screen_width'].'</p>';

        if($_SESSION['screen_width']<900){
            $_SESSION['device']="phone";
        }else{
            $_SESSION['device']="PC";
        }
    }

    if(isset($_SESSION['device'])){
        echo "<br/>Розпізнаний пристрій: ";
        if($_SESSION['device']=="phone"){
            echo "телефон";
        }else{
            echo "комп'ютер";
        }
    }

```

```
    }

    ?>
  </div>

</div>
<div class="footer">
  <br><br>
</div>
<style type="text/css">
@media (max-width: 900px){
    .ping{
        max-width: 100%;
        height: auto;
    }

    body{
        background-image: none;
    }

    html{
        background-color: skyblue;
    }

    .section_title{
        width: 90%;
        margin-left: 0%;
        margin-right: 0%;
```

```
        }
    }
</style>
<?php
//echo '<br/>Час виконання скрипта: '.round((microtime(true) -
$mark)*1000, 10).' мс.';
?>
</body>
</html>
```

### **measure.php**

```
<?php
session_start();
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Magwork</title>
    <style type="text/css">
        body{
            background-color: peachpuff;
        }

        h1{
            text-align: center;
        }

        .container{
            background-color: blanchedalmond;
```

```
        min-height: 700px;
    }
</style>
<link          rel="stylesheet"          type="text/css"
href="./css/index.css">
</head>
<body>
<div class="header">
    <h1><a href="/">Magwork</a></h1>
</div>
<div class="nav">
    <ul>
        <li><a href="measure.php">Швидкість з'єднання</a></li>
        <li><a
            href="addFile.php">Завантаження
фотографій</a></li>
        <li><a
href="locationTest.php">Місцезнаходження</a></li>
        <li><a href="timer.php">Тест впливу</a></li>
        <li><a
            href="compressed.php">Стискання
зображень</a></li>
    </ul>
</div>
<div class="container">
    <h1>Вимірювання швидкості інтернету</h1>
    <p>
        Ми визначимо час, за який Ваш комп'ютер здатний
завантажити 1024 кілобайти текстової інформації. Результат буде
збережений у сесії та використаний для визначення оптимальних
параметрів інших сторінок. Зверніть увагу, що вимірний показник
є актуальним на момент вимірювання та постійно зазнає змін до
деякого ступеня.
    </p>
```



```

<?php
$kb=1024;
echo "<br><hr><br>Відправка $kb Кб...<!--";
flush();
$time = explode(" ",microtime());
$start = $time[0] + $time[1];
for($x=0;$x<$kb;$x++){
    echo str_pad('', 1024, '.');
    flush();
}
$time = explode(" ",microtime());
$finish = $time[0] + $time[1];
$deltat = $finish - $start;
$speed=round($kb / $deltat, 3);
echo "-> Вимірювання завершено за $deltat секунди. Результат:".
$speed."Kb/s<br><br><hr><br>";
$_SESSION['pagespeed']=$speed;
if($_SESSION['pagespeed']>20000){
    echo "<link rel='stylesheet' href='css/above20.css'>";
}else{
    echo "<link rel='stylesheet' href='css/below20.css'>";
}
?>
<p>
    Зелений - вище норми
</p>
<p>
    Червоний - нижче норми
</p>
<br>

```

```
<div class="colored_block" style="width: 200px; height:
200px;"></div>
</div>
<style>
    .container{
        border: 3px solid black;
        margin-left: 5%;
        margin-right: 5%;
        padding: 15px;
    }
</style>
</body>
</html>
```

### **country\_choice.php**

```
<?php

$pc_speeds = array(
    'UA' => 63000,
    'CA' => 157800,
    'RO' => 188000,
    'NL' => 133000,
    'JP' => 148010,
    'SE' => 192130
);

$mobile_speeds = array(
    'UA' => 14000,
    'CA' => 59600,
    'RO' => 21400,
    'NL' => 54800,
```

```

    'JP' => 49300,
    'SE' => 42800
);

if(isset($_SESSION['device']) && $_SESSION['device']=='mobile'){
    $_SESSION['expected_speed'] =
    ($_SESSION['pagespeed']+$_mobile_speeds[$_details->country])/2;
}else{
    $_SESSION['expected_speed'] =
    ($_SESSION['pagespeed']+$_pc_speeds[$_details->country])/2;
}

switch($_details->country) {
    case 'UA':
        $_SESSION['country_color1']="#136bcc";
        $_SESSION['country_color2']="#fed501";
        $_SESSION['country_color3']="";
        $_SESSION['back_font']="white";
        $_SESSION['front_font']="black";
        break;
    case 'SE':
        $_SESSION['country_color1']="#358CFF";
        $_SESSION['country_color2']="#fed501";
        $_SESSION['country_color3']="";
        $_SESSION['back_font']="white";
        $_SESSION['front_font']="black";
        break;
    case 'CA':
        $_SESSION['country_color1']="#CC0000";
        $_SESSION['country_color2']="#FFFFFF";
        $_SESSION['country_color3']="";

```

```

    $_SESSION['back_font']="white";
    $_SESSION['front_font']="black";
    break;
case 'RO':
    $_SESSION['country_color1']="#025abb";
    $_SESSION['country_color2']="#fed501";
    $_SESSION['country_color3']="#CC0000";
    $_SESSION['back_font']="white";
    $_SESSION['front_font']="black";
    break;
case 'NL':
    $_SESSION['country_color1']="#CC0000";
    $_SESSION['country_color2']="#FFFFFF";
    $_SESSION['country_color3']="#025abb";
    $_SESSION['back_font']="white";
    $_SESSION['front_font']="black";
    break;
default:
    $_SESSION['expected_speed']=$_SESSION['pagespeed'];
    $_SESSION['country_color1']="";
    $_SESSION['country_color2']="";
    $_SESSION['country_color3']="";
    $_SESSION['back_font']="";
    $_SESSION['front_font']="";
    break;
}
?>

```

### **timer.php**

```

<!DOCTYPE html>
<html lang="en">

```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Magwork</title>
<style type="text/css">
    body img{
        /*max-width: 400px;
        max-height: 200px;*/
    }
</style>
<link rel="stylesheet" type="text/css" href="./css/index.css">
</head>
<body>
    <div class="header">
        <h1><a href="/">Magwork</a></h1>
    </div>
    <div class="nav">
    <ul>
        <li><a href="measure.php">Швидкість з'єднання</a></li>
        <li><a href="addFile.php">Завантаження фотографій</a></li>
        <li><a href="locationTest.php">Місцезнаходження</a></li>
        <li><a href="timer.php">Тест впливу</a></li>
        <li><a href="compressed.php">Стискання зображень</a></li>
    </ul>
</div>
    <?php
    $start=microtime(true);
    ?>
    <p>Перегляньте низ сторінки, щоб дізнатись час завантаження її
вмісту.</p>
```



```


















```

```
<?php
```

```
    echo '<br/>Час виконання скрипта: '.round((microtime(true) -
    $start)*1000, 10).' мс.';
```

```
    ?>
```

```
</body>
```

```
</html>
```

### **locationTest.php**

```
<?php
```

```
session_start();
```

```
?>
```

```
<html>
```

```
<head>
```

```
<title>Magwork</title>
<meta content="text/html; charset=UTF-8" http-equiv="Content-
Type">
<link rel="stylesheet" type="text/css" href="/css/index.css">
<style type="text/css">
body{
    background-color: peachpuff;
}

h1{
    text-align: center;
}

.container{
    background-color: blanchedalmond;
    min-height: 700px;
}
</style>

</head>
<body>
    <div class="header">
        <h1><a href="/">Magwork</a></h1>
    </div>
    <div class="nav">
        <ul>
            <li><a href="measure.php">Швидкість з'єднання</a></li>
            <li><a href="addFile.php">Завантаження фотографій</a></li>
            <li><a href="locationTest.php">Місцезнаходження</a></li>
            <li><a href="timer.php">Тест впливу</a></li>
```



```

        <li><a href="compressed.php">Стискання зображень</a></li>
    </ul>
</div>
<h1>Тест географічного розміщення</h1>
<div class="container">
    <?php
function getIp() {
    $keys = [
        'HTTP_CLIENT_IP',
        'HTTP_X_FORWARDED_FOR',
        'REMOTE_ADDR'
    ];
    foreach ($keys as $key) {
        if (!empty($_SERVER[$key])) {
            $ip = trim(end(explode(',', $_SERVER[$key])));
            if (filter_var($ip, FILTER_VALIDATE_IP)) {
                return $ip;
            }
        }
    }
}

$ip = getIp();
// виведем IP клиента на экран

echo '<p>Ваш ip: <b>' . $ip.'</b></p>';

function ip_details($ip) {
    $json = file_get_contents("http://ipinfo.io/{$ip}");
    $details = json_decode($json);
    return $details;
}

```

```
}

$details = ip_details($ip);
if(isset($details->city)){
    echo '<p>Місто:<b>'.$details->city.'</b></p>';
    echo '<p>Країна:<b>'.$details->country.'</b></p>';
    echo '<p>Провайдер:<b>'.$details->org.'</b></p>';
    echo '<p>Назва хоста:<b>'.$details->hostname.'</b></p>';
    $_SESSION['country']=$details->country;

    /*налаштування для різних країн*/
    require 'country_choice.php';

}
else{
    echo '<p>Інформація про розташування відсутня. Можливо ви
користуєтесь локальним сервером?</p>';
}
?>
</div>

<style type="text/css">
    h1{
        text-align: center;
    }
    .container{
        border: 3px solid black;
        margin-left: 5%;
        margin-right: 5%;
        padding: 15px;
    }
}
```

```

<?php
    if(isset($_SESSION['country_color1'])){
        echo 'body{ background-color:
'. $_SESSION['country_color1'].';}';
    }
    if(isset($_SESSION['country_color2'])){
        echo '.container{ background-color:
'. $_SESSION['country_color2'].';}';
    }
    if(isset($_SESSION['country_color3'])){
        echo '.container{ border-color:
'. $_SESSION['country_color3'].';}';
    }
    if(isset($_SESSION['back_font'])){
        echo 'body{ color: '. $_SESSION['back_font'].';}';
    }
    if(isset($_SESSION['front_font'])){
        echo '.container{ color:
'. $_SESSION['front_font'].';}';
    }
?>
</style>
</body>
</html>

```

### **compressed.php**

```

<html>
<head>
    <link rel="stylesheet" type="text/css"
href="./css/index.css">
</head>
<body>

```

```

<?php
$image = imagecreatefromjpeg("./compress/water.jpg");
$res = imagejpeg($image, "./compress/water2.jpg", 10);
?>
<div class="header">
    <h1><a href="/">Magwork</a></h1>
</div>
<div class="nav">
    <ul>
        <li><a href="measure.php">Швидкість з'єднання</a></li>
        <li><a href="addFile.php">Завантаження
фотографій</a></li>
        <li><a href="locationTest.php">Місцезнаходження</a></li>
        <li><a href="timer.php">Тест впливу</a></li>
        <li><a href="compressed.php">Стискання
зображень</a></li>
    </ul>
</div>
<p>
    Перед стисканням:
</p>

<p>
    Після стискання:
</p>

</body>
</html>

```