

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»**



МЕХАНІКО-МАШИНОБУДІВНИЙ ФАКУЛЬТЕТ
Кафедра конструювання, технічної естетики і дизайну

Методичні вказівки
для самостійної роботи студентів при виконанні лабораторних робіт
з дисципліни
«Сучасні WEB технології»
для студентів усіх спеціальностей

Дніпро
2020

Методичні рекомендації для самостійної роботи студентів при виконанні лабораторних робіт з дисципліни «Сучасні WEB технології» для студентів усіх спеціальностей / І.В. Вернер, Т.О. Письменкова, В.Е. Дитюк – Дніпро: НТУ «ДП», 2020. – 68 с.

Упорядники:

І.В. Вернер, ас.

Т.О. Письменкова, канд.пед.наук

В.Е. Дитюк

Затвержено до видання редакційною радою НТУ «ДП» (протокол № 00 від 09.10.2020) за поданням кафедри КТЕД (протокол № 00_від 20.09.2020).

Методичні рекомендації для самостійної роботи при виконанні лабораторних робіт з дисципліни «Сучасні WEB технології» для студентів усіх спеціальностей допоможуть студентам у освоєнні сучасних принципів і технологій створення веб додатків та елементів їх дизайну.

Відповідальний за випуск завідувач кафедри конструювання, технічної естетики і дизайну к.т.н., доц. К.А. Зіборов.

ЗМІСТ

Вступ.....	4
Основні поняття, терміни і визначення.....	5
Лабораторна робота № 1	7
Лабораторна робота № 2	18
Лабораторна робота № 3	21
Лабораторна робота № 4	31
Лабораторна робота № 5	36
Лабораторна робота № 6	40
Лабораторна робота № 7	44
Лабораторна робота № 8	51
Лабораторна робота № 9	59
Література.....	60

ВСТУП

Стрімкий розвиток суспільства зумовлений зростанням обсягу інформації та удосконаленням технологій, призвів до появи ряду нових професій. В кожній галузі діяльності вимоги до фахівців зростають, висувуються ряд додаткових компетентностей які потребують активної взаємодії багатьох учасників та об'єктів процесу.

Творчість та іноваційність є важливою складовою будь-якого процесу. Конструювання не є виключенням, а навіть має в основі діяльності творчий процес, що спонукає до іноваційності підходів, дій та результатів. Так підготовка фахівців вимагає суттєвих змін.

Невід'ємною частиною діяльності будь-якого сучасного підприємства є аналіз ринку товарів і послуг, а також поширення серед покупців інформації про новинки, анонсування нової продукції і послуг. На даний момент істотна частка рекламних акцій, а також продаж здійснюється онлайн. Отже, вміння працювати з інформаційними системами комерційної спрямованості є обов'язковим навиком фахівця будь-якої і технічній галузі, зокрема.

У освітніх стандартах багатьох спеціальностей до фахівців з вищою освітою на ряду з фаховими компетентностями висувається ряд додаткових компетентностей які необхідні у будь-якій сфері діяльності, як професійній так і побутовій.

Представлені методичні рекомендації створені з метою допомогти студентам у виробленні умінь створювати графічні елементи сучасних веб орієнтованих систем і вибори самих систем відповідно до потреб. З метою створення набуття навичок до розповсюдження інформації о нових виробках студенти мають змогу опанувати методи створювання сучасних електронних ресурсів та налаштовувати рекламні розсилки використовуючи системи керування контентом із відкритим вихідним кодом.

Основні поняття, терміни і визначення

Інтерфейс – у перекладі з англійської мови означає «зовнішній вигляд» програмного продукту (панелі інструментів, головне меню, його робоча область и т.д.).

Веб-сервер - сервер, який приймає HTTP-запити від клієнтів, зазвичай веб-браузерів, і видає їм HTTP-відповіді, як правило, разом з HTML-сторінкою, зображенням, файлом, медіа-потокком або іншими даними.

HTTP (англ. HyperText Transfer Protocol - «протокол передачі гіпертексту») – протокол прикладного рівня передачі даних, спочатку – у вигляді гіпертекстових документів в форматі «HTML», зараз використовується для передачі довільних даних.

HTML(англ. HyperText Markup Language – «мова гіпертекстової розмітки») – стандартизований мову розмітки документів у всесвітній павутині. Більшість веб-сторінок містять опис розмітки на мові HTML (або XHTML).

CSS (англ. Cascading Style Sheets - каскадні таблиці стилів) – формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки.

MySQL – це одна з найпопулярніших і найпоширеніших СУБД (система управління базами даних) в інтернеті. Вона не призначена для роботи з великими обсягами інформації, але її застосування ідеально для інтернет сайтів, як невеликих, так і досить великих.

SQL (Sequence query language) – язык (запитів) праці із базами даних.

CMS (Content management system) – це система управління контентом, набір скриптів для створення, редагування і управління контентом сайту. На професійному жаргоні CMS також називають «движок». Прикладами CMS є WordPress, PrestaShop, OpenCart та ін.

Блог ([англ.](#) blog, від web log - [інтернет](#)-журнал подій, інтернет-щоденник, онлайн-щоденник) – веб-[сайт](#), Основний зміст якого - регулярно додаються записи, що містять текст, зображення або [мультимедіа](#).

Інтернет магазин ([англ.](#) online shop або e-shop) – [сайт](#), що [торгує](#) товарами за допомогою мережі [Інтернет](#). Дозволяє користувачам [онлайн](#), в своєму [браузері](#) або через [мобільний додаток](#), сформувані замовлення на [покупку](#), вибрати спосіб [оплати](#) і доставки замовлення, оплатити замовлення.

Плагін (Англ. Plug-in, від plug in «підключати») – незалежний компільований програмний модуль, що динамічно підключається до основної програми і призначений для розширення і / або використання її можливостей.

Віджет – автономний елемент сторінки, який виконує різні завдання. Має призначений для користувача інтерфейс і контролер (клас віджету), який готує і обробляє необхідні дані.

Скрипт – це невелика програма, написана, як правило, для потреб інтернет-сайту або програми.

Framework (англ. Framework - «каркас, структура») – в веб-програмуванні це спеціальна програмна платформа або комплекс компонентів і моделей, які полегшують процес веб-розробки. Хоча фреймворки відомі і в інших областях

програмування, саме в веб-розробці, в останні роки вони отримали інтенсивний розвиток.

Тег – це оператор мови HTML, за допомогою якого виконується розмітка вихідного тексту веб-сторінки.

FTP клієнт – програмне забезпечення з допомогою якого відбувається завантаження та зміна документів на веб сервері використовуючи протокол передачі файлів.

FTP або File Transfer Protocol – це один з базових протоколів передачі файлів, призначений для передачі файлів в мережі між комп'ютерами.

Сайт, або веб-сайт ([англ.](#) website: web - «павутина, мережа» і site - «місце», буквально «місце, сегмент, частина в мережі») – одна або кілька логічно пов'язаних між собою [веб-сторінок](#); також місце розташування контенту [сервера](#). Зазвичай сайт в [Інтернеті](#) являє собою масив пов'язаних даних, що має унікальну адресу і сприймається користувачем як єдине ціле. [Веб-сайти](#) називаються так, тому що доступ до них відбувається по протоколу [HTTP](#).

Гіперпосилання ([англ.](#) Hyperlink) – частина гіпертекстового документа, що посилається на інший елемент (команда, текст, заголовок, примітка, зображення) в самому документі, на інший об'єкт (файл, каталог, додаток), розташований на локальному диску або в комп'ютерній мережі, або на елементи цього об'єкта.

Гарнітура – повний комплект графем символів алфавіту, виконаних в одному стилі.

Накреслення – варіація малюнка символів тексту за рахунок зміни товщини їх штрихів і нахилу.

Піксель – це мінімальна одиниця виміру, що обчислюється щодо пристрою відображення.

Властивість «за умовчанням» – властивість якого-небудь параметра, встановлена на етапі розробки програмного продукту. Ця властивість може бути змінена користувачем, після чого вона переходить у розряд поточних.

Лабораторна робота 1 Базові принципи роботи із HTML.

Мета роботи: ознайомитись з основними принципами верстки та оформлення інтернет сторінок з використанням мови розмітки гіпертексту HTML.

Перед початком роботи рекомендується встановити безкоштовний редактор тексту Notepad ++.

Мова HTML інтерпретується браузером і відображається у вигляді документа в зручній для людини формі.

Браузер інтерпретує код HTML для відображення його на комп'ютері, планшеті або телефоні. Мова HTML була розроблена британцем Тімом Бернерс-Лі, в ЦЕРНі. В першу чергу мова HTML призначався для обміну науковими документами. Верстка документів відбувається за допомогою спеціальних операторів мови – дескрипторів (але найчастіше їх називають **тегами**).

Було безліч версій HTML, в процесі доопрацювання з'являлися все нові і нові версії. На даний момент остання версія – **HTML5**. Перша бета версія **HTML5** з'явилася восени 2007 року. Для спрощення і зручності було введено поняття «гіпертекст». Гіперпосилання (або просто посилання), є частиною гіпертексту, і вона може посилатися на інший **HTML документ** або на певну позицію всередині існуючої сторінки.

HTML був побудований так, щоб сторінки відображалися на всіх пристроях однаково. Пізніше додали каскадну таблицю стилів (CSS), що істотно розширило візуальну складову оформлення.

Елементи мови розмітки гіпертексту

Структурно веб сторінки складаються з елементів які описують відповідні параметри вмісту. Усередині елемента розташовується тег мови розмітки містить дані (рис.1). Теги мови розмітки розміщуються всередині фігурних дужок.

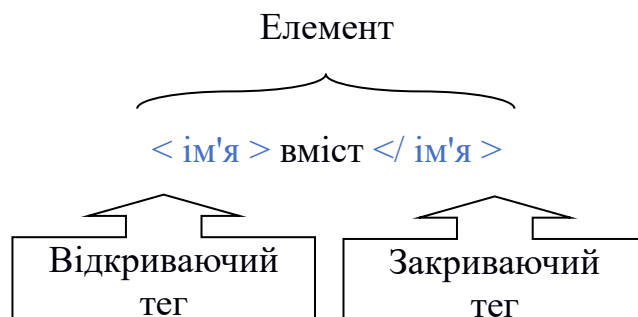


Рис.1. Структура елемента мови розмітки

У мові HTML відбувається розподіл тегів на одинарні (не парні) і парні теги. Отже, одинарні HTML теги складаються з одного тега, тобто, у них немає

закриваючого тега. А у парних дескрипторів є теги що відкриваються і закриваються.

Одинарні теги HTML

Одинарні теги не мають закриваючого тега. Наприклад: `
`, ``. Раніше писали так: `
`, ``, тепер такий стиль написання одинарних тегів не актуальний. Найуживаніші одинарні теги: `
` – перенесення на новий рядок, `<hr>` - розділова лінія, `` - вставка зображення.

Парні теги HTML

Парних тегів набагато більше. У парного дескриптора є відкриваючий і закриваючий тег. Парний тег утворює контейнер. Вміст контейнера – це те, що знаходиться між відкриваючим і закриваючим тегом. Наприклад тег заголовка першого рівня `<H1> Що таке дескриптори в HTML? </H1>`, це парний тег, вмістом цього тега є текст «[Що таке теги в HTML?](#)». Його початком є тег `<H1>`, а кінцем `</H1>`.

Структура HTML документа

Як говорилося вище – ця мова повністю складається з тегів. У будь-якої веб сторінки є початок і кінець, вони позначаються парним тегом `<html>` `</html>`. Розглянемо простий HTML документ:

код HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Тема сайту</title>
  </head>
  <body>
    <h1>Тема першого рівня</h1>
    <p>абзац</p>
  </body>
</html>
```

Як бачите, HTML документ починається з `<!DOCTYPE html>` – кожна веб сторінка повинна починатися саме з нього. Раніше була величезна кількість варіантів тега `DOCTYPE`, але відповідно до новою версії HTML, потрібен тільки цей `!DOCTYPE`. Це **тег**, який повідомляє браузеру версію HTML. Цей **тег** повинен бути написаний на першій сходинці HTML документа.

Після йде парний тег `<html>` `</html>`. Цей тег є контейнером веб сторінки, тобто до нього і після нього нічого не повинно бути.

В **парному тезі** `<head>` `</head>` пишеться обов'язковий тег `<Title>` `</Title>`. У тезі тайтл пишеться **заголовок сторінки**, Який відображається у видачі пошукової системи і у вкладці браузера. У контейнері `<Head>` `</Head>` зазвичай знаходяться різні мета-теги і **підключаються стилі і скрипти**, інформація що знаходиться в ньому є здебільшого службової і на веб сторінку не виводиться.

Весь текст, графіка та інший код повинен знаходитися всередині тегів `<body>` `</body>`. В даному прикладі HTML документа створений заголовок першого рівня `<h1>` і абзац `<p>` (рис.2). Зверніть увагу, що вони знаходяться всередині тега `<Head>`.

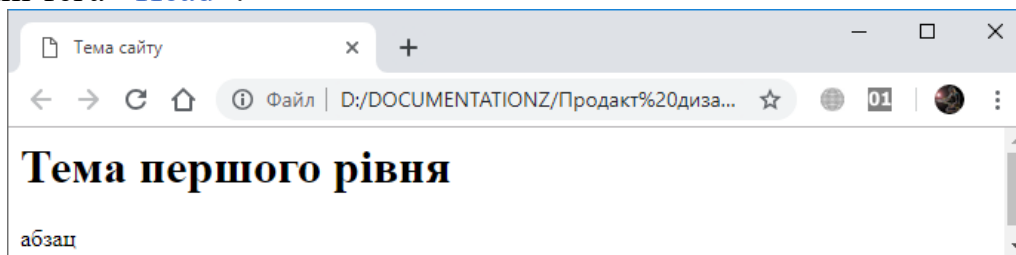


Рис. 2. Приклад оформлення

При форматуванні часто доводиться створювати вкладені одна в одну теги, наприклад, це правильне вкладення: `<P>` `` `ТЕХТ` `` `</p>`, правильність полягає не в тому, що тег «b» вкладається в тег «p», а в тому, що теги закриті послідовно. Ось не правильний запис: `<P>` `` `ТЕХТ` `</p>` ``, і щоб не плутатися в послідовності (візьмемо, наприклад, `<div>` `<p>` `` `ТЕХТ` `` `</p>` `</div>`), Рекомендується робити так:

- Крок 1. Написати найстарший контейнер (той який перший): `<div>` ... `</div>`
- Крок 2. Написати наступний тег всередину старшого тега: `<div>` `<p>` ... `</p>` `</div>`
- Крок 3. Пишемо останній тег, всередину другого тега: `<div>` `<p>` `` ... `` `</p>` `</div>`
- Крок 4. Дописувати текст, в саму середину: `<div>` `<p>` `` `ТЕХТ` `` `</p>` `</div>`

Наступний приклад показує всі основні парні теги. Щоб захистити себе від помилок рекомендується при роботі використовувати текстовий редактор Notepad ++, який містить інструментарій з перевірки парних тегів.

```
код HTML
<!DOCTYPE html>
<html>
  <head>
    <title>Тема сайту</title>
  </head>
  <body>
    <h1>Тема 1 рівень</h1>
    <h2>Тема 2 рівень</h2>
    <h3>Тема 3 рівень</h3>
    <h4>Тема 4 рівень</h4>
    <h5>Тема 5 рівень</h5>
    <h6>Тема 6 рівень</h6>
    <p>абзац</p>
    <b>жирний текст</b>
    <u>підкреслений текст</u>
```

```

<i>курсивний текст</i>
<ul>
  <li>Маркований список 1</li>
  <li>Маркований список 2</li>
</ul>
<ol>
  <li>Нумерований список 1</li>
  <li>Нумерований список 2</li>
</ol>
</body>
</html>

```

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Тема сайту</title>
5   </head>
6   <body>
7     <h1>Тема 1 рівень</h1>
8     <h2>Тема 2 рівень</h2>
9     <h3>Тема 3 рівень</h3>
10    <h4>Тема 4 рівень</h4>
11    <h5>Тема 5 рівень</h5>
12    <h6>Тема 6 рівень</h6>
13    <p>абзац</p>
14    <b>жирний текст</b>
15    <u>підкреслений текст</u>
16    <i>курсивний текст</i>
17    <ul>
18      <li>Маркований список 1</li>
19      <li>Маркований список 2</li>
20    </ul>
21    <ol>
22      <li>Нумерований список 1</li>
23      <li>Нумерований список 2</li>
24    </ol>
25  </body>

```

Рис.3. Код в Блокноті ++

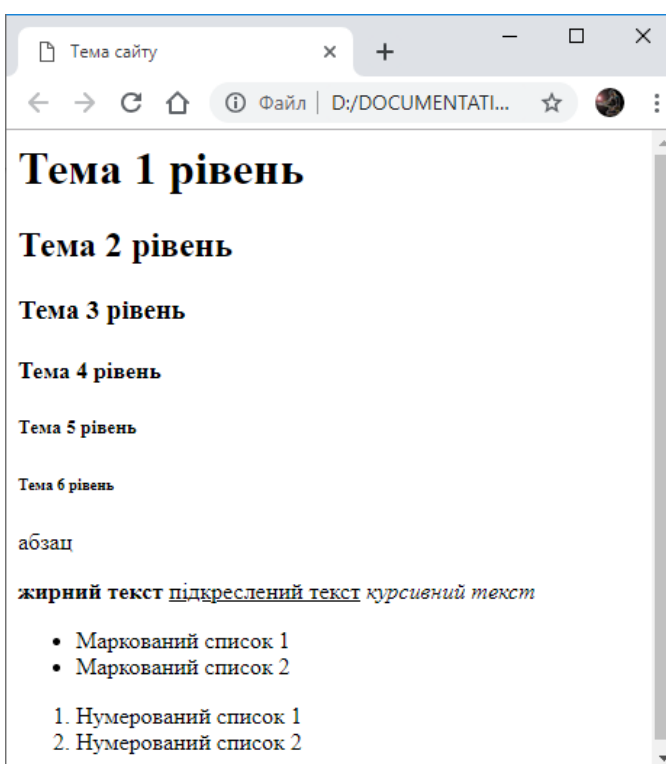


Рис.4. Приклад в браузері

Використання HTML метатегів

Мета-тег – це **тег мови HTML**, який розміщують в **заголовку веб сторінки**, тобто між тегами `<head> ... </head>`. Метатеги можуть містити опис сторінки, інформацію про автора, команди для пошукових роботів. Є спеціальні метатеги для соціальних мереж.

Мета-теги для соціальних мереж дуже корисно використовувати для збільшення переходів на сайт з соціальних мереж.

Мета-теги кодування HTML сторінки

Якщо на сторінці в інтернеті замість вмісту з'являються якісь знаки питання, квадратики та інше то у цього сайту проблема з кодуванням, цілком можливо що розробник її не вказав. Кодування, простими словами, це таблиця що містить набір символів. Для того щоб додати кодування сайту, потрібен

багатофункціональний тег `<Meta>`, який вставляється всередині тега `<Head>`. Отже, щоб браузер зрозумів, яке кодування ви використовуєте, необхідно вставити наступний код:

код HTML

```
<meta http-equiv = "Content-Type" content = "text / html; charset = utf-8">
```

Таким чином кодування – це набір символів, представлений в таблиці. Таблиця містить сам символ і його код, наприклад, двійковий (нулі і одиниці). Термін «набір символів» рідко використовується, найчастіше говорять «кодування». На даний момент популярні два кодування: ASCII і UTF-8. У наших роботах використовується кодування UTF-8.

Мета-теги для пошукових систем

Мета-тег для опису сторінки зазвичай індексується пошуковими системами і інформація виводиться на сторінці пошуку:

код HTML

```
<meta name = "description" content = "Опис сторінки">
```

Метатег для створення редиректу, де X - кількість секунд перед перенаправленням, а site.com - сторінка, куди перенаправляти:

код HTML

```
<meta http-equiv = "refresh" content = "X; url = site.com">
```

А так же група мета-тегів для керування роботами (індексацією сайту):

код HTML

```
<meta name = "robots" content = "X">
```

Замість X може бути наступне:

- noodr - не використовувати опис з каталогу DMOZ (Google) для сниппета в пошуковій видачі.
- noyasa - не використовувати опис з Яндекс.Каталогу для сниппета в пошуковій видачі.
- all - дозволено індексувати посилання і текст на сторінці.
- nofollow - не переходити за посиланнями, розташованим на даній сторінці.
- noindex - не індексувати текст цієї сторінки.
- none - заборонено переходити за посиланнями та індексувати текст на сторінці.

Мета-теги Google - місце розташування підприємства, офісу

Мета-тег для вказівки довготи:

код HTML

```
<meta property = "place: location: longitude" content = "0.0122">
```

Мета-тег для вказівки широти:

код HTML

```
<meta property = "place: location: latitude" content = "2.3122">
```

Мета-тег для вказівки країни:

код HTML

<meta property = "business: contact_data: country_name" content = "Країна">

Мета-теги Google для вказівки контактної інформації

код HTML

<meta property = "business: contact_data: postal_code" content = "49000">

<meta property = "business: contact_data: email" content = "ill3@ukr.net">

<meta property = "business: contact_data: phone_number" content = "+380 975096629">

<meta property = "business: contact_data: website" content = "http://sci-forum.net.ua">

Мета-теги для соціальних мереж (Facebook)

код HTML

<meta property = "fb: admins" content = "Facebook_ID">

<meta property = "og: type" content = "profile">

<meta property = "profile: last_name" content = "Прізвище">

<meta property = "profile: first_name" content = "Ім'я">

<meta property = "og: title" content = "Простір назв">

<meta property = "og: description" content = "Опис сторінки">

<meta property = "profile: username" content = "Нік">

<meta property = "og: image" content = "img.png">

Також існує велика кількість мета-тегів для різних служб і сервісів, інформацію про які можна дізнатися на сайтах відповідних служб.

Форматування тексту

При форматуванні тексту часто використовується тег параграфа <p>. При цьому текст можна всередині параграфа формувати наступними методами: по ширині, по правому або лівому краю і по центру. За це відповідають відповідно параметри атрибута вирівнювання абзацу **align**: justify, right, left і center. Синтаксис написання параграфа тексту з вирівнюванням по центру наступний: <P align = "center">Рядок по центру</P>. Більшість можливостей форматування в даний час здійснюється за допомогою каскадних таблиць стилів.

Додатково нижче представлені теги для форматування тексту із зазначенням на його приналежність.

- 1) Дуже важливий текст
- 2) Просто важливий текст
- 3) <Ins>підкреслений текст</Ins>
- 4) <Cite>Цитата курсивним текстом</Cite>
- 5) <Big>збільшили шрифт</Big>
- 6) <Small>зменшили шрифт</Small>
- 7) 3²

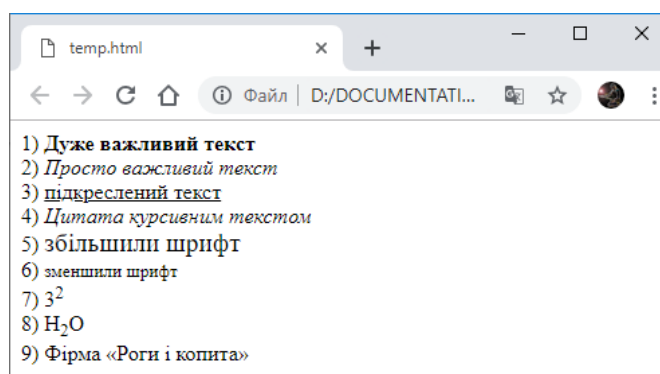


Рис.5. Приклади роботи тегів

8) Н₂О

форматування тексту

9) Фірма <q lang="Ru">Роги і
копита</Q>

В даний час роботу із шрифтами рекомендується здійснювати за допомогою каскадних таблиць стилів. Але іноді при створенні масових розсилок деякі системи вимагають застосовувати старий тег шрифту `font` що має наступний синтаксис: ` текст`.

Як атрибути у нього є 3 властивості: `color` - колір шрифту, `size` - розмір шрифту у відносних одиницях і `face` - гарнітура шрифту. Наприклад:

`<P> П`ерша буква цієї пропозиції написана шрифтом Arial, виділена червоним кольором і збільшена в розмірах.`</P>`

Робота з гіперпосиланнями

Тег `<a>` є одним з важливих елементів HTML і призначений для створення посилань. Залежно від присутності атрибутів `name` або `href` тег `<a>` встановлює якір або посилання. Якорем називається закладка всередині сторінки, яку можна вказати в якості мети посилання. При використанні посилання, яка вказує на якір, відбувається перехід до закладки всередині веб-сторінки.

Для створення посилання необхідно повідомити браузеру, що є посиланням, а також вказати адресу документа, на який слід зробити посилання. Як значення атрибута `href` використовується адреса документа (URL, Universal Resource Locator, універсальний покажчик ресурсів), на який відбувається перехід. Адреса посилання може бути абсолютною або відносною. Абсолютні адреси працюють всюди, незалежно від імені сайту або веб-сторінки, де прописано посилання. Відносні посилання, як впливає з їх назви, побудовані щодо поточного документа або кореня сайту.

Синтаксис:

`текст гіперпосилання`

`текст якоря`

Деякі їх можливих параметрів:

`href` - задає адресу документа, на який слід перейти;

`name` - встановлює ім'я якоря всередині документа;

`target` - ім'я вікна або фрейму, куди браузер буде завантажувати документ;

`self` - відкривається на тій же сторінці; `_blank` - відкриється нове вікно;

`title` - додає підказку до тексту посилання.

Робота з зображеннями

Тег `` призначений для відображення на веб-сторінці зображень в графічному форматі GIF, JPEG або PNG. Адреса файлу з картинкою задається через атрибут `src`. Якщо необхідно, то малюнок можна зробити посиланням на інший файл, помістивши тег `` в контейнер `<a>`. При цьому навколо зображення відображається рамка, яку можна прибрати, додавши атрибут `border = "0"` в тег ``.

Малюнки також можуть застосовуватися в якості карт-зображень, коли картинка містить активні області, які виступають в якості посилань. Така карта за зовнішнім виглядом нічим не відрізняється від звичайного зображення, але при цьому воно може бути розбите на невидимі зони різної форми, де кожна з областей служить посиланням.

Синтаксис: ``

Основні атрибути:

[align](#) - визначає як малюнок буде вирівнюватися по краю і спосіб обтікання текстом;

[alt](#) - альтернативний текст для зображення;

[border](#) - товщина рамки навколо зображення;

[height/width](#) - висота / ширина зображення;

[hspace/vspace](#) - горизонтальний / вертикальний відступ від зображення до навколишнього контенту;

[ismap](#) - говорить браузеру, що картинка є серверною картою-зображенням;

[src](#) - шлях до графічного файлу;

[usemap](#) - посилання на тег `<Map>`, що містить координати для клієнтської карти-зображення.

Робота з таблицями

Елемент `<Table>` служить контейнером для елементів, що визначають вміст таблиці. Будь-яка таблиця складається з рядків і осередків, які задаються за допомогою тегів `<Tr>` і `<Td>`. У середині `<Table>` допустимо використовувати наступні елементи: `<Caption>`, `<Col>`, `<Colgroup>`, `<Tbody>`, `<Td>`, `<Tfoot>`, `<Th>`, `<Thead>` і `<Tr>`.

Таблиці з невидимою кордоном довгий час використовувалися для верстки веб-сторінок, дозволяючи розділяти документ на модульні блоки. Подібний спосіб застосування таблиць знайшов втілення на багатьох сайтах, поки йому на зміну не прийшов більш сучасний спосіб верстки за допомогою шарів.

Синтаксис

```
<Table>
  <Tr>
    <Td>...</Td>
  </Tr>
</Table>
```

Атрибути

[align](#) - визначає вирівнювання таблиці;

[background](#) - задає фоновий малюнок у таблиці;

[bgcolor](#) - колір фону таблиці;

[border](#) - товщина рамки в пікселях або відсотках;

[bordercolor](#) - колір рамки;

[cellpadding](#) - відступ від рамки до вмісту осередку;

[cellspacing](#) - відстань між осередками;

[frame](#) - повідомляє браузеру, як відображати межі навколо таблиці;

[height/width](#) - висота / ширина таблиці.

Робота зі списками

Тег `` встановлює маркований список. Кожен елемент списку повинен починатися з тега ``. Якщо до тега `` застосовується таблиця стилів, то елементи `` успадковують ці властивості.

Синтаксис

```
<ul>
<li> елемент маркованого списку </li>
</ul>
```

Атрибути

type - встановлює вид маркера списку: disc (замовчування), circle, square.

Тег `` встановлює маркований список. Нумеровані списки є набором елементів з їх порядковими номерами. Вид і тип нумерації залежить від атрибутів тега ``, який і застосовується для створення списку. Кожен пункт нумерованого списку позначається тегом ``.

```
<ol>
<li>Перший пункт</li>
<li>другий пункт</li>
<li>третій пункт</li>
</ol>
```

Атрибути списків

type - встановлюють тип нумерації списку: 1 (замовчування - арабські цифри), A (заголовна латиниця), a (рядкова латиниця), I (римські заголовні), i (римські прописні);

start - для установки довільного початку нумерації. При цьому не має значення, який тип списку встановлений за допомогою **type**, атрибут однаково працює і з римськими і з арабськими числами;

reversed - нумерація в списку стає по спадаючій.

Розглянемо приклад форматування вмісту сторінки за допомогою табличної верстки.

```

1 <table width=500px border=0>
2 <tr>
3 <td>Ліва частина</td>
4 <td>Права частина</td>
5 </tr>
6 <tr>
7 <td colspan=2><p align=center>Центральна частина сторінки</p></td>
8 </tr>
9 <tr>
10 <td>Ліва частина</td>
11 <td>Права частина</td>
12 </tr>
13 </table>
14 <br width=150px>
15 <table width=500px border=1>
16 <tr>
17 <td>Ліва частина</td>
18 <td>Права частина</td>
19 </tr>
20 <tr>
21 <td colspan=2><p align=center>Центральна частина сторінки</p></td>
22 </tr>
23 <tr>
24 <td>Ліва частина</td>
25 <td>Права частина</td>
26 </tr>
27 </table>
28 <br width=150px>
29 <table width=500px>
30 <tr bgcolor=green height=100px>
31 <td>Ліва частина</td>
32 <td>Права частина</td>
33 </tr>
34 <tr>
35 <td colspan=2 height=100px><p align=center>Центральна частина сторінки</p></td>
36 </tr>
37 <tr height=100px bgcolor=gray>
38 <td>Ліва частина</td>
39 <td>Права частина</td>
40 </tr>
41 </table>

```

Рис.6. Лістинг табличної верстки

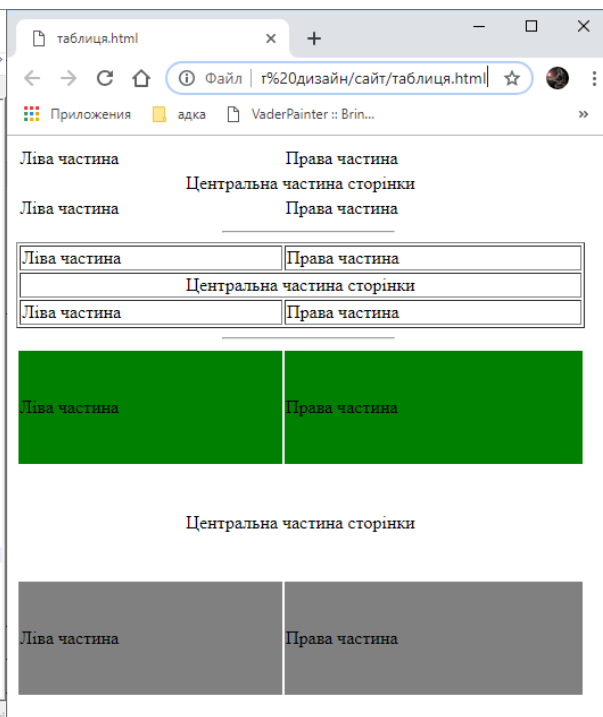


Рис.7. Відображення таблиць в браузері

У середній (другий) рядку останньої таблиці дві колонки об'єднані в одну за допомогою властивості тега `<Td>` стовпчика `colspan = 2`.

Завдання для виконання. Створіть макет для майбутнього сайту візитки, що представляє Вас як фахівця в галузі інженерії. Для цього створіть документ `index.html` і відкрийте його в блокноті ++. Наберіть нижченаведений лістинг і заповніть своїми даними технічні елементи що знаходяться в головній частині сторінки. Макет сайту повинен в майбутньому складатися з трьох сторінок: головна, портфолію і сторінка контактної інформації.

код HTML

```

<!DOCTYPE html>
<html>
  <head>
    <title>Персональна сторінка фахівця із інженерії</title>
    <meta http-equiv = "Content-Type" content = "text/html; charset = utf-8">
    <meta name = "description" content = "Опис сторінки (абзац рекламного
тексту):Виконую інженерні розрахунки будь-якої складності. Проектую обладнання і володію
засобами візуалізації. Звертайтеся за послугами на нашу адресу ">
    <meta property = "business: contact_data: Ukraine" content = "Країна">
    <meta property = "business: contact_data: postal_code" content = "49000">
    <meta property = "business: contact_data: email" content = "ваш@мыл.юа">
    <meta property = "business: contact_data: phone_number" content = "+380 ваш телефон">
    <meta property = "business: contact_data: website" content = "http://ntudp.com">
    <meta property = "fb: admins" content = "Facebook_ID">
    <meta property = "og: type" content = "profile">
    <meta property = "profile: last_name" content = "Прізвище">
    <meta property = "profile: first_name" content = "Ім'я">

```



```
<meta property = "og: title" content = "Простір назв">
<meta property = "og: description" content = "Опис сторінки">
<meta property = "profile: username" content = "Нік">
</head>
<body>
<table width = 700px>
  <tr bgcolor = gray height = 100px>
    <td> Логотип </td>
    <td>
      Меню
      <ul>
        <li> <a href=#> Головна </a> </li>
        <li> <a href=#> Портфоліо </a> </li>
        <li> <a href=#> Контакти </a> </li>
      </ul>
    </td>
  </tr>
  <tr>
    <td colspan = 2 height = 400px>
      <img src = "#" alt = "Фото фахівця" width = "200 px" height = "200 px" title = "Фото фахівця">
      <p> Спеціаліст з промислового дизайну та сертифікації. Має компетентність із використання
      сучасних систем керування контентом. </p>
    </td>
  </tr>
  <tr height = 100px bgcolor = gray>
    <td> Соціальні мережі </td>
    <td> Адреси </td>
  </tr>
</table>
</body>
</html>
```

Приклад виконання чорнової стадії макетування представлений нижче (рис.8). Тип макета для сайту можна змінювати на свій розсуд.

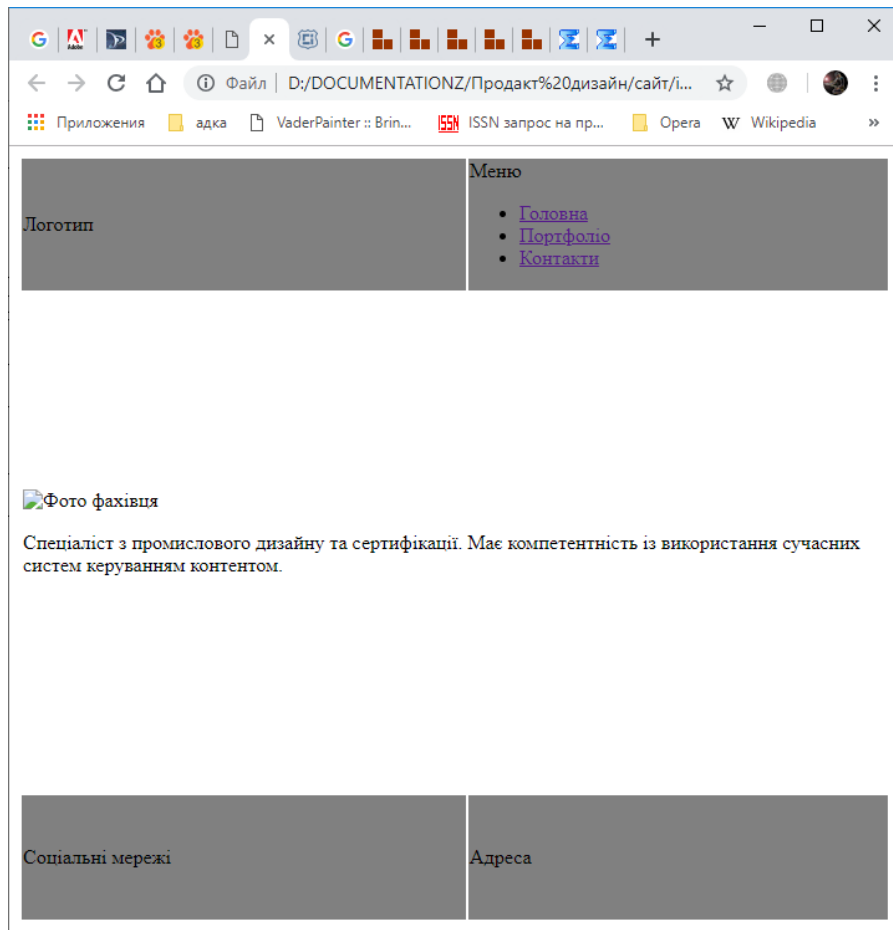


Рис.8. Приклад макетування персонального сайту (сайту візитки)

Лабораторна робота 2 Створення графічного дизайну сайту.

Мета роботи: розробити на основі корпоративних кольорів компанії концепцію та графічний дизайн персональної сторінки або візитки підприємства.

Грунтуючись на зразковому макеті отриманому в результаті виконання попередньої лабораторної роботи створимо графічний макет для сайту. Для даної мети можна використовувати будь-які графічні редактори. Ми будемо розглядати роботу по створенню макета в редакторі Adobe Photoshop, оскільки на нашому курсі студенти вже освоїли навички по роботі з ним.

Відкриємо графічний редактор і створимо документ за розмірами 700x768 px @ 72ppi.

Відкриємо палітру шарів натиснувши F7 і створимо три групи шарів, назвавши їх відповідно: header (шапка сайту), content (наповнення сайту), footer (нижня частина сайту).

В якості фону для сайту в нашому випадку буде використовуватися футуристичне зображення взяте з вільних джерел. Встановлюємо його після фонового шару і відмасштабуємо його на сторінці.

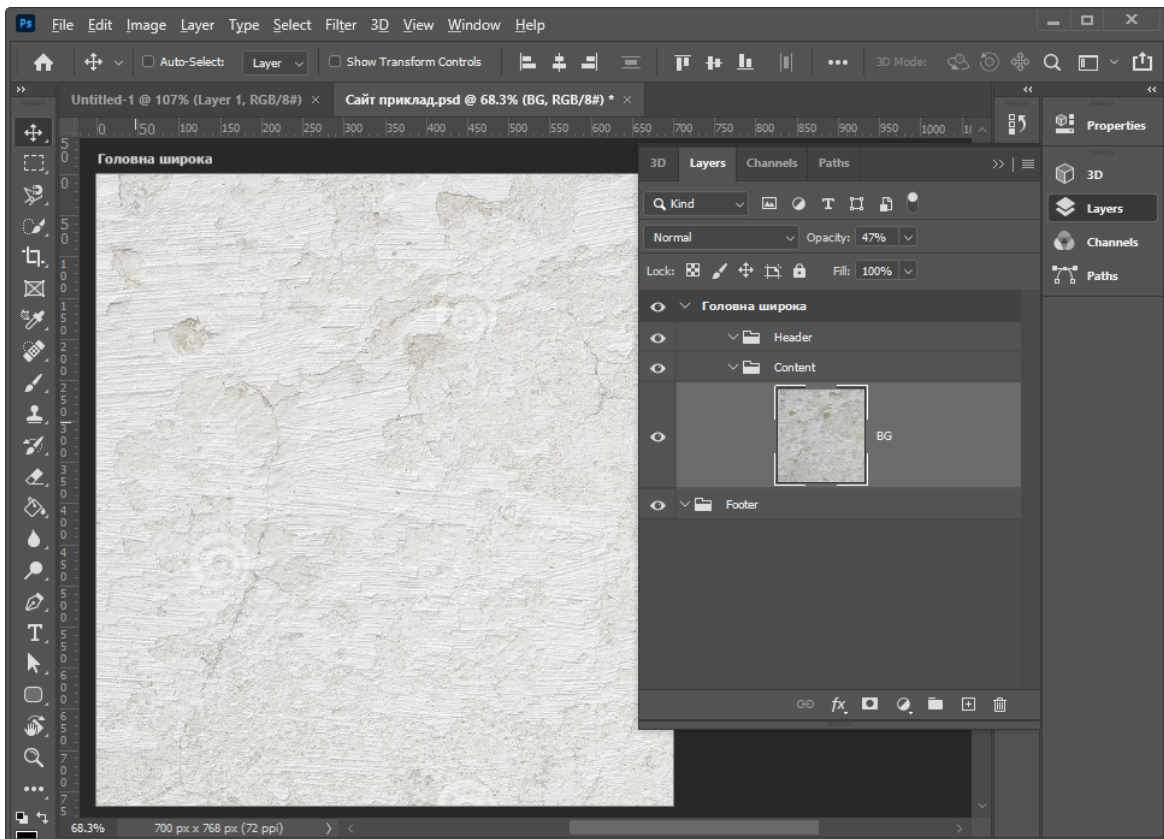


Рис.9. Установка фону для сторінки

Розгорнемо групу шарів content і створимо новий шар для фону внутрішньої частини сторінки. Зробимо прямокутним виділенням середню частину, згладимо кути прямокутника і зафарбуємо білим кольором.

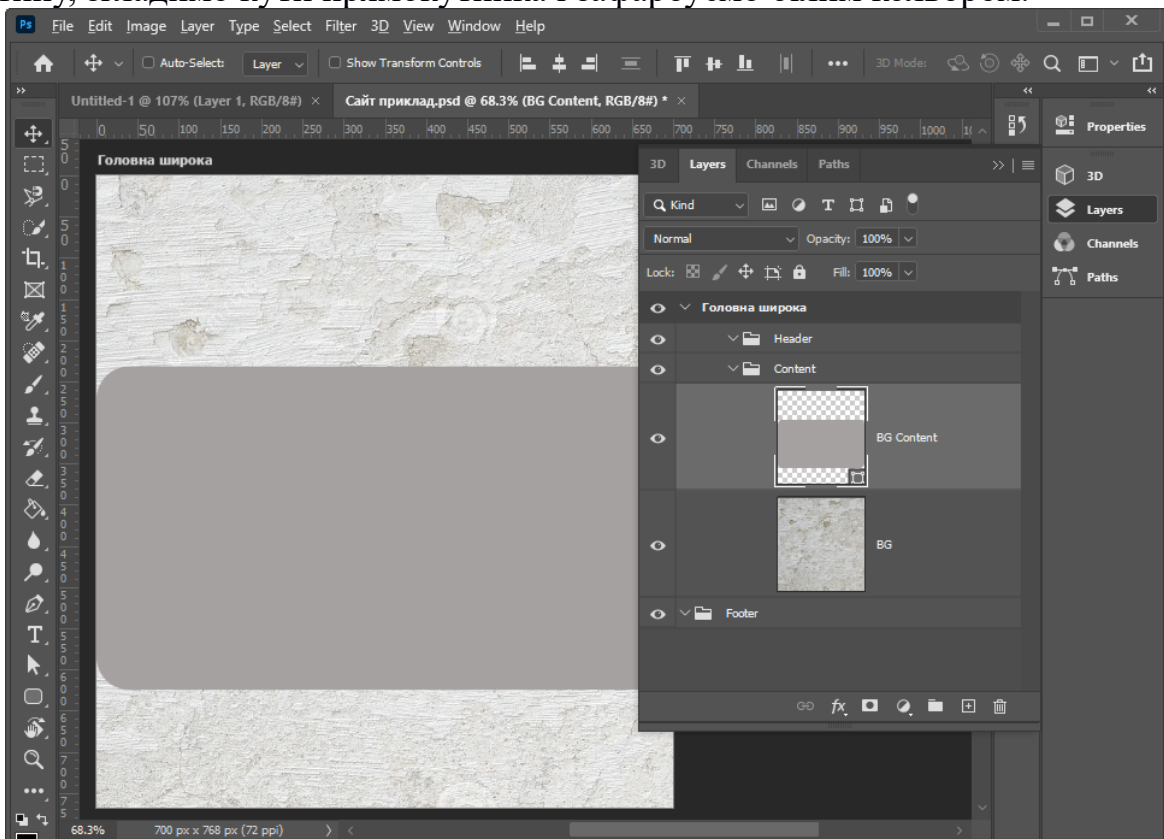


Рис. 10. Створення фону для внутрішньої частини сайту

Закриємо групу шарів контенту і відкриємо групу шарів шапки сайту. Додамо логотип нашого підприємства, створимо шар назвавши його меню і зробимо округлені прямокутні області для кнопочок. Зафарбуємо їх градієнтною заливкою. Поверх кнопочок вставимо текстові шари з назвами: Головна, Портфоліо і Контакти. На свій розсуд створимо дизайн для назви дисципліни «Сучасні WEB технології» і впровадимо в верхню частину.

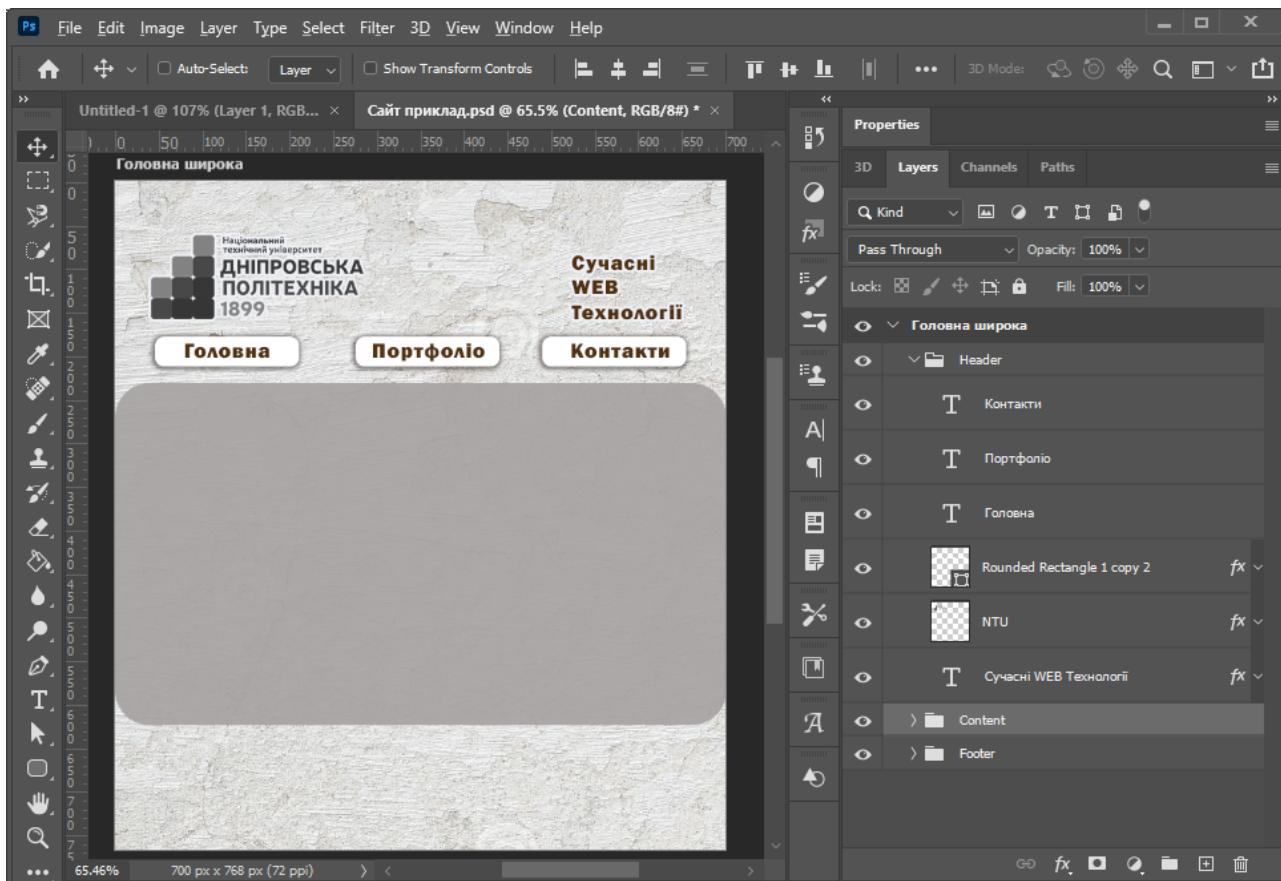


Рис. 11. Створення елементів верхньої частини сайту

Таким же чином в нижню частину сайту в відповідну групу шарів footer впровадимо іконки соціальних мереж і адресу підприємства.

У середню частину сайту додамо фотографію студента його ПІБ і навчальну групу. Також створимо текстовий блок що найбільш широко описує навички студента як майбутнього фахівця в своїй області.

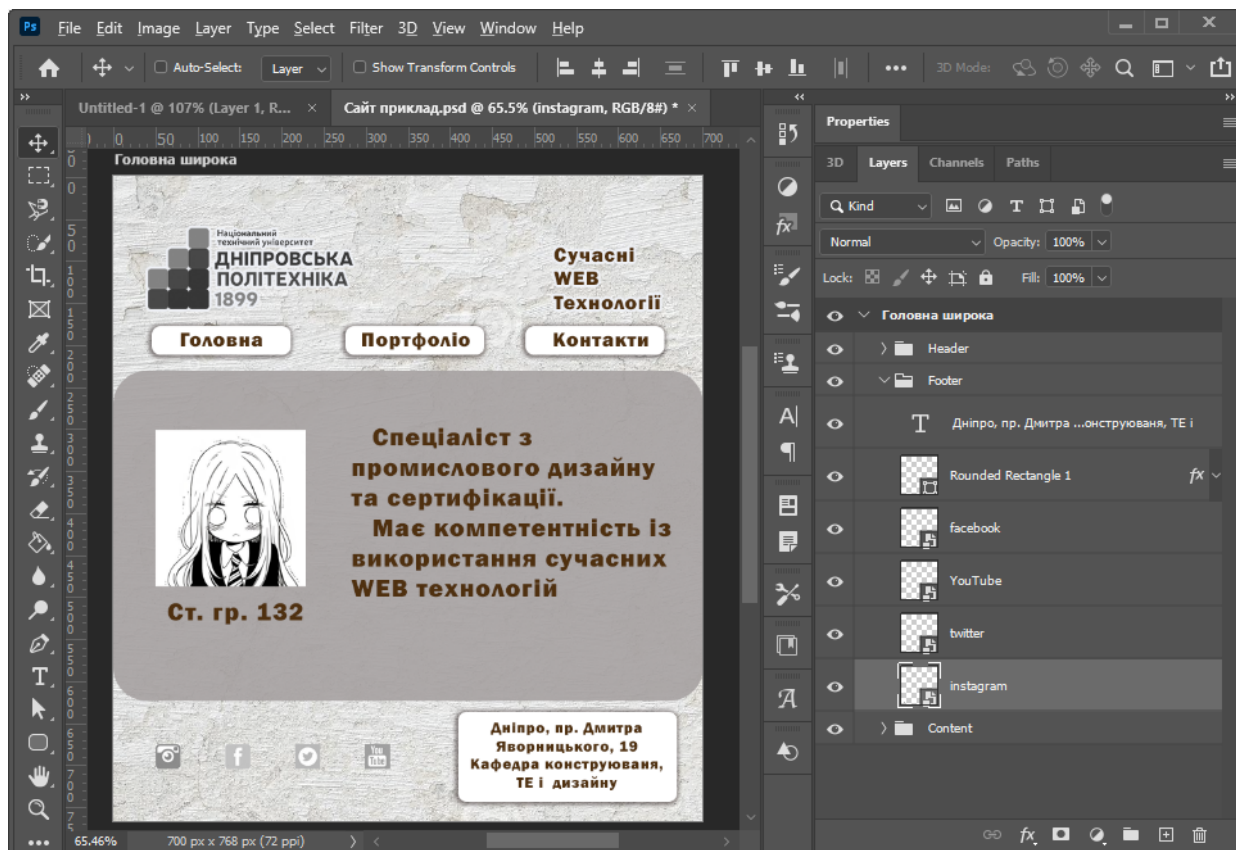


Рис. 12. Приклад створення макета сайту

Для використання графічних елементів при форматуванні інтернет сторінки всі елементи повинні бути збережені окремо у відповідних форматах. Кожен елемент що містить прозорі області повинен бути збережений у форматі PNG, що містить анімацію GIF, інші JPG.

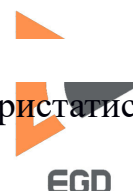
Також в Photoshop зручно вбудований індикатор кольору в шістнадцятковій системі (у вікні вибору кольору) для використання його в веб дизайні. Налаштування кольору в HTML і CSS задаються саме в ній, так для установки білого або чорного кольору необхідно відповідно вказувати параметр #000000 або #ffffff.

Завдання для виконання. Створіть графічний макет для майбутнього сайту візитки, що представляє Вас як фахівця в галузі інженерії. Для цього використовуючи запропоновану методику створіть свій унікальний шаблон або повторіть запропонований в роботі варіант.

Лабораторна робота 3 Базові принципи робота із CSS.

Мета роботи: ознайомитись з основними принципами верстки та оформлення інтернет сторінок з використанням каскадних таблиць стилей CSS.

Для створення будь-якої сучасної Web-сторінки необхідно скористатися



двома мовами: HTML і CSS. HTML використовується для логічної розмітки Web-сторінки, тобто вказує браузеру, що знаходиться на ній і в якому порядку. Завдяки мові HTML браузер виводить на екран тексти, малюнки, відео і формує Web-сторінку.

CSS «пояснює» браузеру, як повинні виглядати всі ці об'єкти. У таблиці стилів описується колір, накреслення, розмір шрифту, відступи, вирівнювання, кольори фонів або фонові малюнки, властивості таблиць, зображень, форм, позиціонування елементів і т.і.

Використання двох методів опису Web-сторінки дозволяє працювати окремо з вмістом (HTML) і дизайном сторінки (CSS).

Також існує ряд графічних елементів присутніх на всіх сторінках сайту і їх опис займав багато місця в коді кожної сторінки раніше. Завдяки CSS з'явилася можливість раз описавши об'єкт в окремому файлі таблиці стилів просто підключати його до необхідних сторінок не перевантажуючи кодом самі інтернет сторінки. Таким чином для редагування дизайну всього сайту в цілому тепер не потрібно редагувати всі сторінки сайту, досить відкрити файл з настройками таблиці стилів і поміняти параметри в ньому.

Опис стилів CSS

Стили можна описати двома способами: підключити файл або описати в HTML документі, в контейнері `<head>`. Опис стилів відбувається в контейнері `<style>`. Наприклад, в блоці `<style>` прописано для класу `userclass` властивість `font-size`. `font-size` змінює розмір тексту, в даному прикладі, в пікселях (px). Щоб застосувати цей клас до якого-небудь тегу, ми пишемо атрибут `class` і в значення пишемо ім'я нашого класу. В CSS клас починається з «.» (Крапка), а `id` з «#» (решітка, sharp). Приклад використання класу `userclass`:

Код HTML

```
<Html>
  <Head>
    <Title> codebra </title>
    <style type = "text/css">
      .userclass {
        width: calc (100% - 20px - 4px); /* Ширина шару */
        text-align: center; /* Вирівнювання тексту */
        font-size: 40px; /* Розмір тексту */
        color: # 39D1B4; /* Колір тексту */
        border: 2px solid # 39D1B4; /* Параметри кордону */
        margin: 0 0 20px 0; /* Значення відступів */
        padding: 10px; /* Поля навколо тексту */
      }
    </style>
  </Head>
  <Body>
    <p class = "userclass">Привіт</p>
  </Body>
</Html>
```

Так само коли стилі описані в тілі документа, вони можуть розташовуватися в атрибутах окремого елемента:

Код HTML

```
<p style = "font-size: 20px; color: green; font-family: arial, Helvetica, sans-serif">  
Текст зеленого кольору</p>
```

Підключення зовнішніх файлів CSS в HTML

Це другий спосіб опису стилів. Цей спосіб більш ефективний і зручний. Для його реалізації використовується тег `<link>`. Він має три важливих атрибута: `type`, `rel`, `href`. `Type` – вказує тип підключається документа. `Rel` - повідомляє ставлення між файлами. `Href` - посилання на файл.

Приклад підключення файлу CSS

Код HTML

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

Приклад підключення файлу, що лежить в папці, яка знаходиться в директорії "include"

Код HTML

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

Приклад підключення файлу розміщеного окремо від батьківського документа

Код HTML

```
<Style media = "all">  
    @import url (style.css);  
</ Style>
```

Правила побудови CSS

У випадках підключення зовнішнього файлу стилів CSS до документа кожне правило CSS з файлу має дві основні частини - **селектор** і **блок оголошень**. Селектор, розташований в лівій частині правила, до знака «{», визначає, на які частини документа (можливо, спеціально позначені) поширюється правило. Блок оголошень розташовується в правій частині правила. Він міститься в фігурні дужки, і, в свою чергу, складається з одного або більшої кількості оголошень, розділених знаком «;». Кожне оголошення є поєднанням властивості CSS і значення, розділених знаком двокрапки ":". Селектори можуть групуватися в одному рядку через кому. В такому випадку властивість застосовується до кожного з них.

```
селектор, селектор {  
    властивість: значення;  
    властивість: значення;  
    властивість: значення;  
}
```

Таблиця 1. Приклади деяких селекторів

Приклади селекторів			
тегів	класів	ідентифікаторів	псевдокласів
<pre>p { font-family: Arial, helvetica, sans-serif; }</pre>	<pre>.note { color: red; background-color: yellow; font-weight: bold; }</pre>	<pre>#paragraph1 { margin: 0; }</pre>	<pre>a:active { color: blue; }</pre>

Основна відмінність між класами елементів і ідентифікаторами елементів в тому, що ідентифікатор призначений для одного елемента, тоді як клас зазвичай привласнюють відразу декільком. Проте, сучасні браузерери, як правило, вірно відображають множинні елементи з однаковим ідентифікатором. Також відмінність в тому, що можуть існувати множинні класи (коли клас елемента складається з декількох слів, розділених пробілами). Для ідентифікаторів таке неможливо.

Імена класів і ідентифікаторів, на відміну від назв тегів і їх атрибутів, чутливі до регістру введення букв.

Властивості класів і ідентифікаторів задаються за допомогою відповідних селекторів. Причому може бути задано як властивість класу в цілому (в такому випадку селектор починається з «.», наприклад, «.big») або властивість ідентифікатора самого по собі (в такому випадку селектор починається з "#", наприклад, «#first»), так і властивість якогось елемента цього класу або з цим ідентифікатором.

В CSS, крім класів, що задаються автором сторінки, існує також обмежений набір так званих *псевдокласів*, що описують вид гіперпосилань з певним станом в документі, вид елемента, на якому знаходиться фокус введення, а також вид елементів, які є першими дочірніми елементами інших елементів. Також в CSS існує чотири так званих *псевдоелемента*: перша буква, перший рядок, застосування спеціальних стилів до і після елемента.

CSS-верстка

До появи CSS оформлення веб-сторінок здійснювалося виключно засобами ***HTML***, безпосередньо всередині вмісту документа. Однак з появою CSS стало можливим принципове розділення змісту і представлення документа.

Переваги:

- Кілька дизайнів сторінки для різних пристроїв перегляду. Наприклад, на екрані дизайн буде розрахований на велику ширину, під час друку меню не виводитиметься, а на **КПК** і телефоні меню буде слідувати за вмістом.
- Зменшення часу завантаження сторінок сайту за рахунок перенесення правил представлення даних в окремий CSS-файл. В цьому випадку браузер завантажує тільки структуру документа і дані, що зберігаються на сторінці, а

представлення цих даних завантажується браузером тільки один раз і може бути закешоване.

- Простота подальшої зміни дизайну. Не потрібно правити кожен сторінку, а досить лише змінити CSS-файл.

- Додаткові можливості оформлення. Наприклад, за допомогою CSS-верстки можна зробити блок тексту, який решта тексту буде обтікати (наприклад для меню) або зробити так, щоб меню було завжди видно при прокручуванні сторінки.

Недоліки:

- Різні відображення верстки в різних браузерах (особливо застарілих), які по-різному інтерпретують одні й ті ж дані CSS.

- Найпоширеніша необхідність на практиці виправляти не тільки один CSS-файл, але і теги HTML, які складним і не бачним способом пов'язані з селекторами CSS, що іноді зводить нанівець простоту застосування єдиних файлів стилів і значно збільшує час редагування та тестування.

Блокова верстка сайтів

Використання шарів замість табличної верстки з одного боку значно розширило можливості по зміні дизайну сайту і його функціонал, з іншого боку ускладнило роботу так як необхідно використовувати величезну кількість властивостей об'єкта і його позиціонувати.

Використовуючи шари для верстки сайту ми можемо управляти практично будь-якими параметрами позиціонування шару на сторінці, причому як в абсолютних координатах так і у відносних. Для блочно модульної верстки використовується тег `<div>`.

Елемент `<div>` є блоковим елементом і призначений для виділення фрагмента документа з метою зміни виду вмісту. Як правило, вид блоку керується за допомогою стилів. Щоб не описувати кожен раз стиль всередині тега, можна виділити стиль в зовнішню таблицю стилів, а для тега додати атрибут `class` або `id` з ім'ям селектора.

Як і при використанні інших блочних елементів, вміст тега `<div>` завжди починається з нового рядка. Після нього також додається перенесення рядка.

Код HTML

```
<div id="Content">
```

вміст блоку

```
</Div>
```

Для спрощення верстки сторінок в специфікації HTML 5 увійшли нові блокові елементи, такі як:

```
<Section>Якийсь розділ</Section>
```

```
<Article>Самостійний розділ</Article>
```

```
<Header>Верхня частина блоку</Header>
```

```
<Footer>Нижня частина блоку</Footer>
```

```
<Aside>Додатковий контент</Aside>
```

```
<Nav>Навігаційне меню</Nav>
```

Кожен з них має ті ж можливості що і `<div>`.

Позиціонування блокових елементів

Для опису шару і його позиціонуванні на сторінці використовуються наступні атрибути (рис.13):

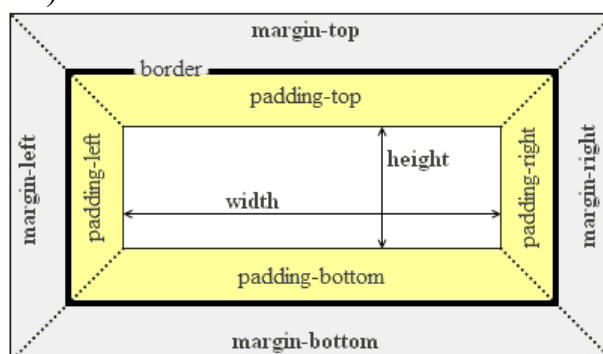


Рис. 13. Позиціонування шару

Margin - задає зовнішні відступи блоку, установка auto центрує об'єкт;

Padding - задає внутрішні відступи блоку;

Border - задає параметри лінії кордону (колір, тип, товщина);

Width / Height - ширина / висота блоку;

Position - розташування на сторінці (фіксоване, абсолютне і відносне).

Ми розглянули тільки найнеобхідніші властивості для виконання нашої роботи. Інші можливості ми розглянемо виконуючи практичний приклад. Зробимо блочну розмітку сайту студента замість табличної.

Спочатку щоб був зрозумілий принцип створимо блочну схему елементів сайту і подивимося як блоки будуть розташовуватися один щодо одного.

Для зовнішньої оболонки дизайну сайту використовуємо шар з класом `wrapper` (обгортка). Ім'я класу можна задавати на свій розсуд. **Внутрі** створимо блок для шапки сайту `<header>`, усередині якого два блоки відповідно для логотипу звичайний шар і для меню блок `<nav>`. Під блоком шапки створимо блок основної частини `<section>` і всередині нього додамо два блоки для фото студента і текстової частини. Під блоком основної частини створимо блок нижньої частини `<footer>`, в даному блоці буде два шари відповідно для іконок соціальних мереж і адреси ВНЗ. Схема розроблена, описуємо відповідний код для шаблону сайту (рис.14). На схемі сайту в якості імен блоків фігурують їх класи.

Код HTML

```
<div class="wrapper">Основний контейнер сайту

<header class=header>
<P class=letter>Головна частина сайту</p>

<div class=logo></div>
<nav class=navigate></nav>

</header>

<section class=content>
<div class=foto></div>
<div class=text></div>
</section>

<footer class=footer>
<p class=letter>Нижня частина сайту</p>

<div class=social></div>
<div class=address></div>
</footer>

</div>
```

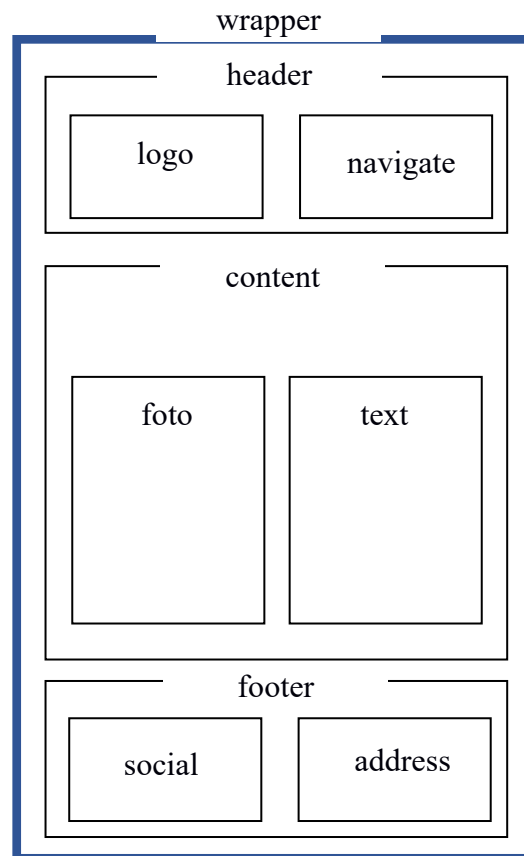


Рис. 14. Блокова схема сайту

Завершальною частиною при створенні дизайну буде додавання стилів для кожного класу блоку в окремий файл і підключення його до HTML документу.

Код CSS

```
.wrapper {
width: 700px; /* Ширина шару */
min-height: 768px; /* Мінімальна висота шару */
margin: auto; /* Значення відступів (встановлено вирівнювання по центру сторінки) */
padding: 0px; /* Поля навколо тексту */
border: 1px solid black; /* Параметри кордону */
background: #fc0; /* Колір фону */
}

.header {
width: 669px; /* Ширина шару */
height: 226px; /* Висота шару */
margin: auto; /* Значення відступів (по центру сторінки) */
padding: 0px; /* Поля навколо тексту */
border: 0px; /* Параметри кордону */
background: #cccccc; /* Колір фону */
background: url (img / head-bg.png); /* Установка фонового зображення */
}

.footer {
width: 669px; /* Ширина шару */
```

```
height: 220px; /* Висота шару */
position: relative; /* Відносне положення */
left: 0; bottom: -300px; /* Відносне зміщення вниз */
margin: auto; /* Значення відступів (по центру сторінки) */
padding: 0px; /* Поля навколо тексту */
border: 0px; /* Параметри кордону */
background: #cccccc; /* Колір фону */
background: url (img / footer-bg.png); /* Установка фонового зображення */
}

.letter {
color: #ffffff; /* Колір тексту */
padding: 0px; /* Поля навколо тексту */
border: 0px;
margin: 0px;
}
```

Запропонований варіант не є найбільш оптимальним але дозволяє побачити яким чином можливо позиціонування основних блоків в шаблоні сайту. Після його застосування зовнішній вигляд сторінки в браузері набуде вигляду показаний на рис.15.

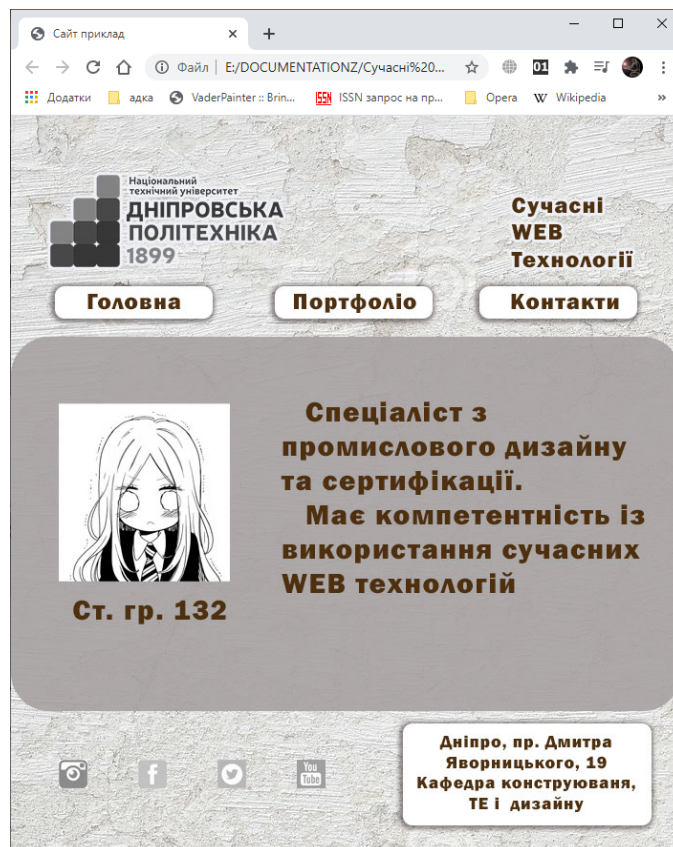


Рис.15. Зовнішній вигляд макета сайту з блочною версткою

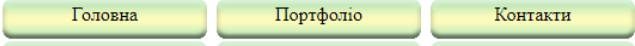
В даному макеті спеціально не встановлені дрібні елементи, а саме логотипи ВНЗ, соціальних мереж, зображення студента, текстовий опис, меню сайту. Студенти дивлячись на приклади позиціонування встановлених блоків повинні самі навчитися встановлювати необхідні блоки і їх позиціонувати використовуючи відповідні атрибути.

Даний сайт є шаблон що жорстко прив'язаний до розмірів, це є трохи застарілим. В даний момент актуально робити адаптивні сайти, для того щоб їх можна було б комфортно використовувати на мобільних пристроях. Для цих цілей можна використовувати відносне позиціонування і завдання розмірів у відносних одиницях або відсотках.

Створення меню

Виходячи із розмірів кнопок 175x35 пікселів створимо горизонтальне меню.

Код HTML	Код CSS
<pre><nav class=navigate> <ul class=menu> Головна Портфоліо Контакти </nav></pre>	<pre>.menu { list-style-type: none; } .menu>li { display: inline-block; vertical-align: top; text-align: center; padding-top: 5px; background: url(img/btns.png); font-size: 1rem; width: 175px; height: 35px; text-decoration: none; } a { text-decoration: none; color: inherit; }</pre>



Створення адаптивних сайтів

Розглянемо практичний приклад створення «гумового» сайту з сітчастою версткою 3x3 колонки.

Код HTML	Код CSS
<pre><div id="grid"> <div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> <div>6</div> <div>7</div> <div>8</div> <div>9</div> </div></pre>	<pre>#grid { display: grid; grid-template-rows: 1fr 1fr 1fr; grid-template-columns: 1fr 1fr 1fr; grid-gap: 2vw; } #grid > div { font-size: 5vw; padding: .5em; background: gold; text-align: center; }</pre>

Результатом відпрацювання даного коду стане наступний варіант адаптивного шаблону сайту (рис.16). Як можна побачити такий стиль дозволяє

змінювати розміри блоків в залежності від ширини оглядача (екрану), також змінюється і вміст блоку - в даному випадку цифри.

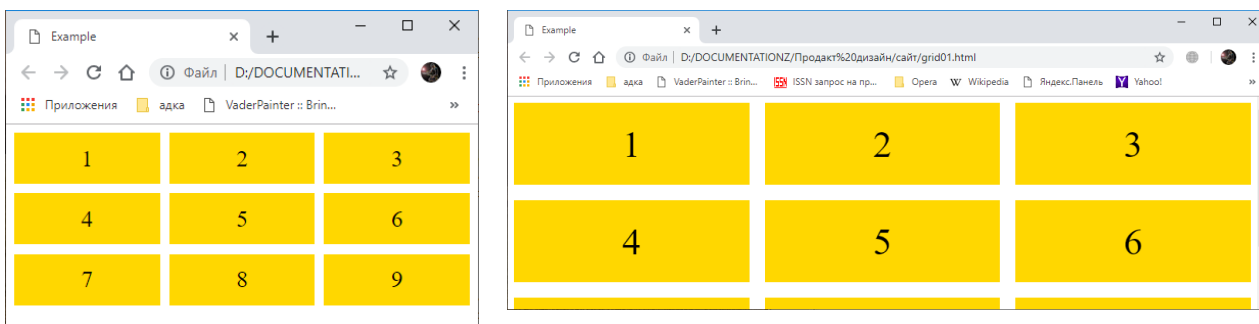


Рис. 16. Приклад роботи адаптивної сітки

Розглянемо докладніше як перетворити блокові елементи в адаптивну сітку: **display: grid**. Перетворює елемент в контейнер сітки. Це все, що потрібно для створення сітки. Тепер у нас є контейнер сітки і елементи сітки. Значення **grid** генерує блоковий контейнер сітки. Ви також можете використовувати **display: inline-grid** для створення вбудованого контейнера сітки, **display: subgrid** для створення підсітки (це значення призначене для використання з самими елементами сітки).

grid-template-rows: 1fr 1fr 1fr. Явно встановлює строки сітки. Кожне значення представляє розмір рядка. В цьому випадку всі значення **1fr** (що означає, що вони займають вільний простір), але їх також можна задати за використанням інших одиниць, наприклад **100px**, **7em**, **30%** і т.і. Ви можете також вказати імена рядків разом з розмірами треків (тобто розміри рядків і стовпців). Дивіться **grid-template-rows** для отримання додаткової інформації.

grid-template-columns: 1fr 1fr 1fr. Те ж, що і значення вище, за винятком того, що визначає стовпці сітки. Дивіться **grid-template-columns** для отримання додаткової інформації.

grid-gap: 2vw. Встановлює зазор. Зазор – це розрив між елементами сітки. Тут ми використовуємо одиниці довжини **vw** (вони співвідносяться з шириною вікна перегляду), але ми могли б так само легко використовуватися **10px**, **1em** і т.і. Властивість **grid-gap** насправді є скороченою властивістю для властивостей **grid-row-gap** і **grid-column-gap**. Дивіться **grid-gap** для отримання додаткової інформації.

Розглянемо наступний приклад адаптивної сітки і подробиці форматування.

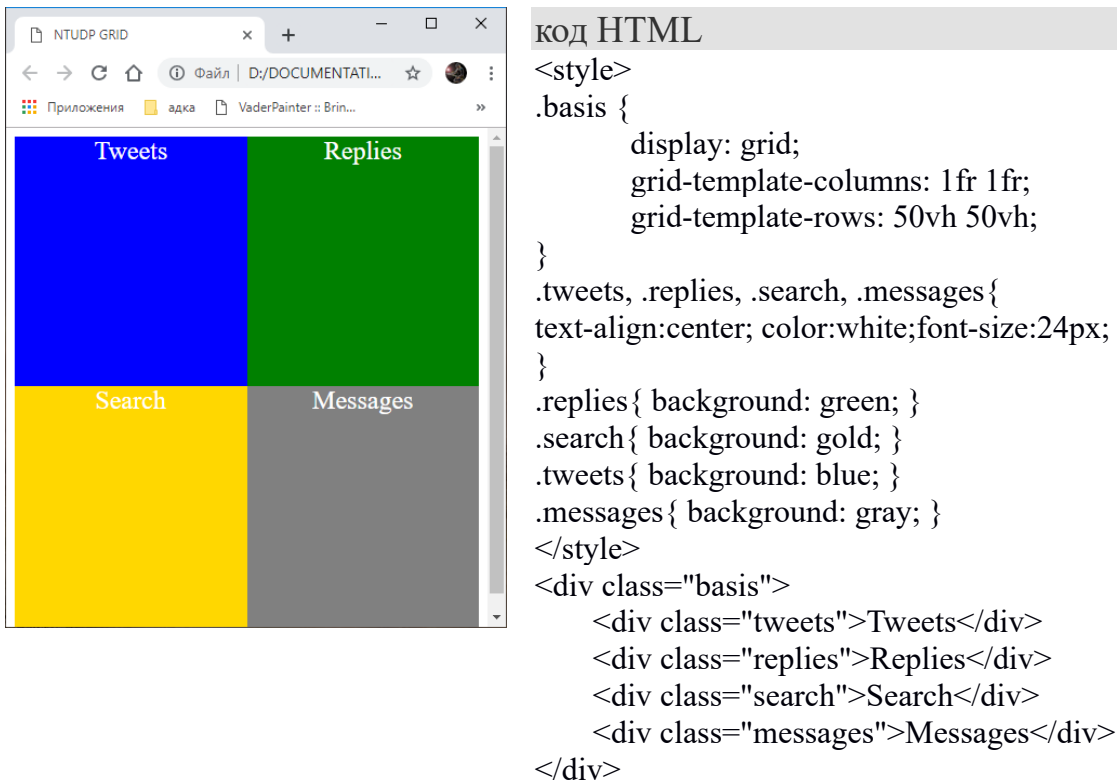


Рис. 17. Приклад макетування та роботи адаптивної сітки

Детальніше розглядаючи відносні **одиниці виявиться що** 1fr (одна частка) вільного простору в контейнері сітки, а 100vh це повна висота видового вікна. Отже задаючи 50vh елемент буде завжди займати половину висоти екрана.

Завдання для виконання. Створіть макет сайту використовуючи модульну верстку. Для цього використовуючи запропоновану методику створіть свій унікальний шаблон або повторіть запропонований в роботі варіант.

Лабораторна робота 4 Створення адаптивних елементів і сайтів.

Мета роботи: ознайомитись з основними принципами створення дизайну адаптивних сайтів використовуючи практичні настанови.

Створення адаптивного меню

В першу чергу необхідно зрозуміти для чого використовується адаптивний дизайн для сайтів. В даний час дуже поширені планшетні комп'ютери і мобільні пристрої з доступом до глобальної мережі та своїми браузерами. У даних пристроїв дуже сильно варіюються розміри екрану. Відповідно якщо інтерфейс сайту буде статичним на екранах монітора ПК він може бути ідеально зверстаний, але виглядати на екранах мобільних і інших портативних пристроїв буде не коректно. Щоб уникнути цих складнощів для користувачей, в даний час

прийнято створювати адаптивний дизайн для інтернет проєктів. Тобто на різних пристроях один і той же сайт може виглядати зовсім по різному, виходячи з розмірів екрану пристрою.

Отже розглянемо приклад адаптивного дизайну на елементі меню сайта. Створимо стандартне меню використовуючи маркований перелік (рис. 18).

Код HTML	Код CSS
<pre><header> <ul class="menu"> Головна Портфоліо Навички Контакти <i class="bar"></i> </header></pre>	<pre>header{ height: 80px; background-color: #234567; display: flex; /* елементи горизонтально */ justify-content: flex-end; /* праворуч */ align-items: center; padding-right: 16px; } .menu { display: flex; /* елементи горизонтально */ list-style-type: none; /* видаляє маркери меню*/ font-size: 24px; } .menu__link{ color: white; padding: 8px; } .bar{ background-image: url(images/menu.png); width: 55px; height: 55px; }</pre>

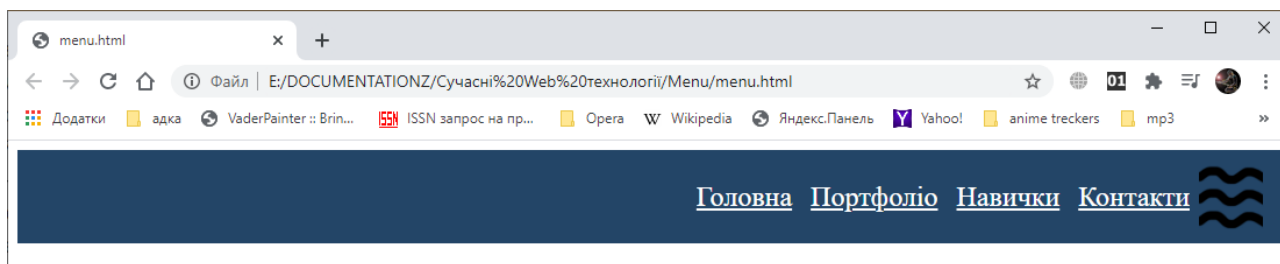


Рис. 18. – Приклад виконання коду меню

Для того щоб адаптивні елементи виглядали по різному на різних пристроях будемо використовувати медіа запити. Наприклад для вказівки виведення блоку на мобільні пристрої обмежені розмірами екрану до 560 пікселів необхідно написати наступне:

```
@media(max-width: 560px){}
```

Отже додаймо наступний код до все існуючого CSS щоб на мобільних пристроях меню згорталось до іконки и колір фону був зелений.

Код CSS


```
@media(max-width: 560px) {
  header{
    background-color: green;
  }
  .menu {
    display: none; /* сховати меню*/
  }
}
```

Перевірте я працює цей елемент зменшивши розмір браузера до мінімуму за шириною.

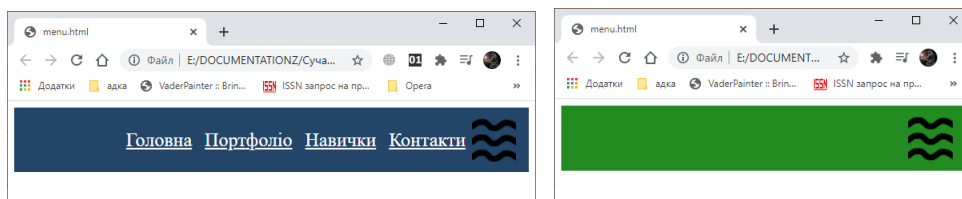


Рис. 19. – Приклад праці медіа запиту

Надалі створено адаптивне меню де в якості елементів керування будемо використовувати літери шрифту. Для відкриття меню можливо натискати на символ ☰, для закриття на той же символ або літеру X. В наданому прикладі скористуємось елементом властивістю checkbox елемента input.

Код HTML

```
<header>
  <input type="checkbox" id="galka">
  <label for="galka"><div
class="fas">☰</div></label>

    <nav>
      <ul class="menu">
        <li> <a class="menu__link"
href="#">Головна</a></li>
        <li> <a class="menu__link"
href="#">Портфоліо</a></li>
        <li> <a class="menu__link"
href="#">Навички</a></li>
        <li> <a class="menu__link"
href="#">Контакти</a></li>
      </ul>
      <label class="close"
for="galka">X</label>
    </nav>
  </header>
```

Код CSS

```
header{
  height: 80px;
  background-color: #234567;
  display: flex; /* елементи горизонтально */
  justify-content: flex-end; /* праворуч */
  align-items: center;
  padding-right: 16px;
}
.menu {
  display: flex; /* елементи горизонтально */
  list-style-type: none; /* видалити маркери меню*/
  font-size: 24px;
}
.menu__link{
  color: white;
  padding: 8px;
  text-decoration:none;
}
.close {
  font-size: 24px;
  color: white;
  display: none;
}
.fas {
  font-size: 24px;
```

```
color: white;
display: block;
}
#galka {
visibility: hidden;
}

@media(max-width: 560px) {
.menu {
width: 100%;
height: 80vh;
flex-direction: column;
justify-content: center;
align-items: center;
}
nav {
position: fixed;
width: 100%;
height: 100vh;
left: 100%;
top: 80px;
background-color: #567890;
display: block;
transition: left .5s;
}

.fas {
display: block;
font-size: 24px;
color: white;
}

.close {
position: absolute;
right: 24px;
top: 24px;
font-size: 24px;
color: white;
display: block;
}

#galka:checked ~ nav {
left: 0;
}
header {
background-color: green;
}
}
```

Для зручності опанування матеріалу надаємо готовий лістинг на хмарній платформі за посиланням: <https://codepen.io/darkness9991/pen/BajrWdd>



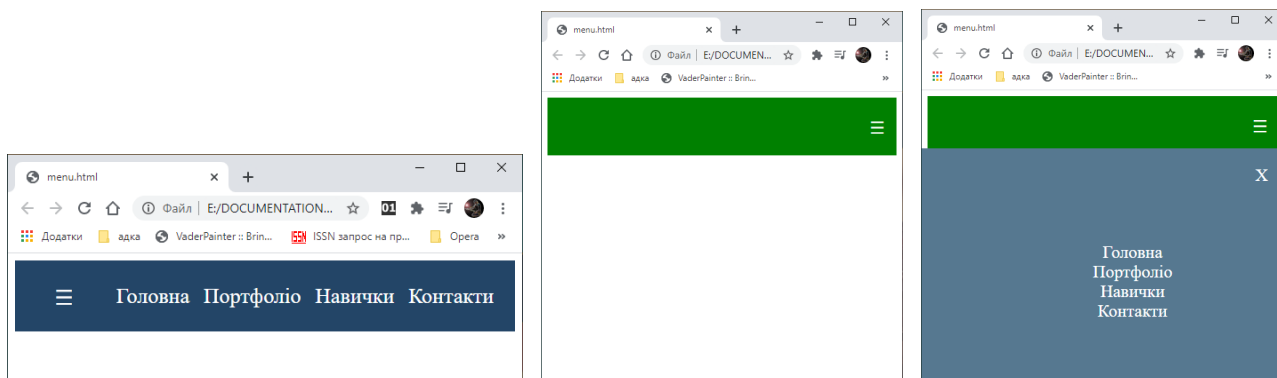


Рис. 20. – Приклад створення адаптивного меню

Також є змога використовувати у дизайні та елементах меню іконки із спеціальних шрифтів. Можливо користуватись сервісом <https://fontawesome.com/>. Після реєстрації на сайті є змога отримати код для активації на сайті можливості користування іконками. Наприклад для отримання символу меню (три лінії) можна встановлювати наступний тег у код:
`<i class="fas fa-bars"></i>`

Створення адаптивного сайту

Результат адаптивного дизайну – адаптивний сайт, яким зручно користуватися незалежно від того, на якому пристрої він переглядається. Сьогодні ця концепція – не примха, а важливий тренд розвитку мережі. І мова йде не просто про «тягнучі» сайти, а про їх «розумну» адаптацію до будь-якого пристрою. Користувачеві, який використовує великий монітор, необхідно декілька колонок тексту. Тому ж, хто зайшов на сайт з мобільного пристрою – лише той контент, який потрібен йому в дорозі (рис. 21).



Рис. 21. – Концепція адаптивних інтернет проектів

Користуючись наведеним прикладом розробіть свій варіант дизайну адаптивного сайту. Для зручності опанування матеріалу надаємо готовий лістинг на хмарній платформі за посиланням [MaxPower Design]: <https://codepen.io/kingsfull123/pen/zwrOZr> (рис. 22).

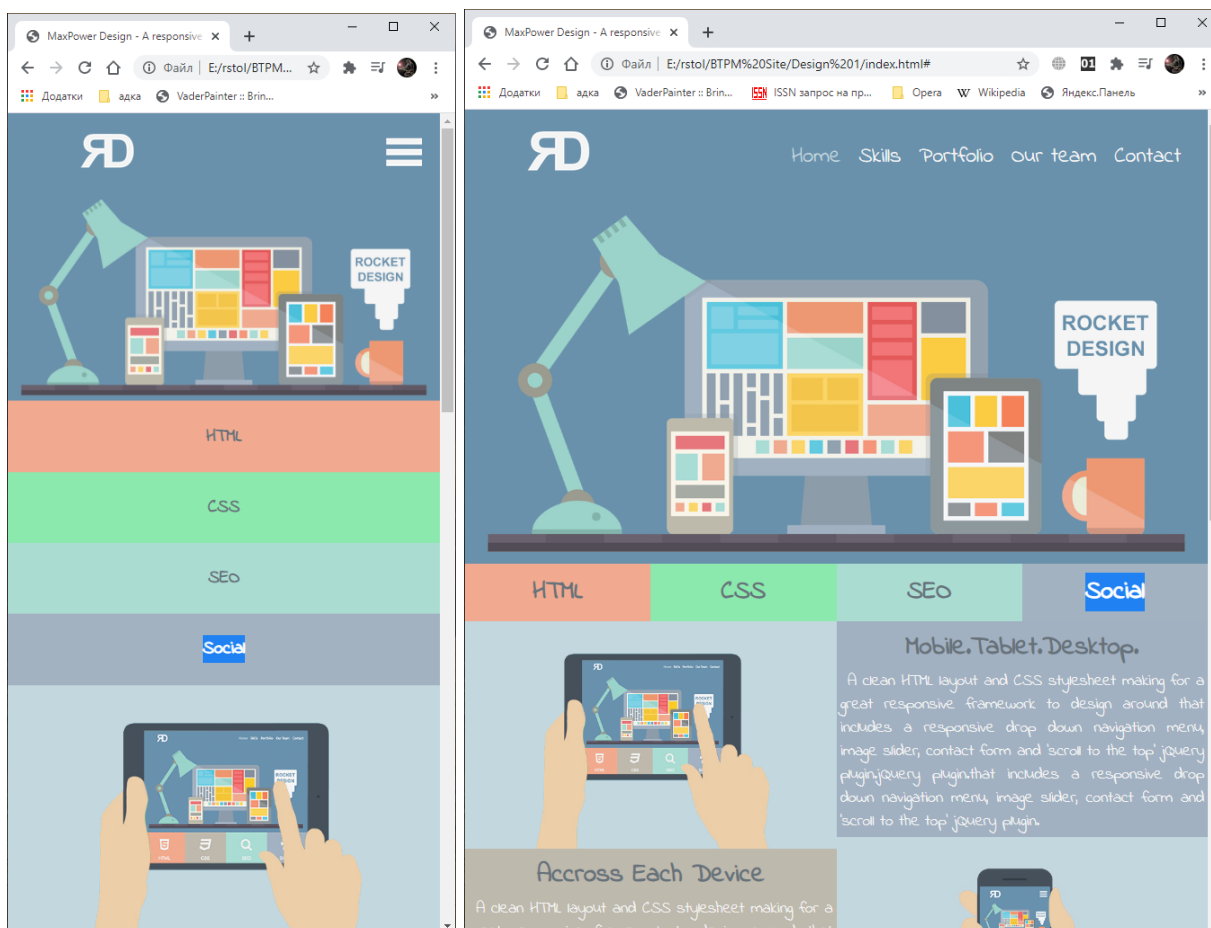


Рис. 22. – MaxPower Design - A responsive website page

Завдання для виконання. Виконуючи цю лабораторну студент повинен створити: адаптивне меню, шаблон адаптивного сайту.

Лабораторна робота 5 Додання елементів з використання Java Script.

Мета роботи: ознайомитись з основними принципами підключення готових скриптів до своїх проектів та ознайомлення з синтаксисом мови.

Впровадження в інтернет-проект JavaScript дозволяє зробити ваш сайт не статичним, а інтерактивним і досить розумним. Додатки на JavaScript дозволяють зробити веб-сторінки наочніше. Наприклад можливо створити слайд-шоу з послідовністю відображення окремих слайдів і динамічними ефектами та багато інших інтерактивних функцій та елементів керування.

JavaScript є об'єктно орієнтованою мовою, при праці потрібно дотримуватися точного написання операторів тому що регістр маленьких і великих літер розпізнається по-різному. Тобто написання ID та Id сприйматиметься як абсолютно різні елементи.

Почнемо знайомлення з цієї мови розглянувши наступний приклад, що дозволяю відобразити на веб сторінці версію браузера.

По-перше потрібно підключити зовнішній файл script.js із кодом у тіло сторінці за допомогою тега <script>, синтаксис:

```
<script src="/шлях/до/script.js"></script>
```

Також можливо створювати і користатись кодом мови у тілі веб сторінки, теж за допомогою цього тега, використовуючи наступний синтаксис:

```
<script type="text/javascript">
```

```
Код.скрипта("властивості");
```

```
</script>
```

Отже повністю лістинг цього завдання виглядає наступним чином:

Код HTML

```
<html>
<head>
<title>Перший скрипт</title>
<style type="text/css">
.highlight {font-weight:bold;}
</style>
</head>
<body>
<h1>Почнемо кодування )</h1>
<hr>
<script type="text/javascript">
<!--
document.write("Версія браузера "+ navigator.appVersion);
document.write(" від <span class='highlight'" + navigator.appName + "</span>.");
-->
</script>
</body>
</html>
```

Отже наданий код дозволяє використовуючи вбудовані функції виводити потрібну інформацію (рис. 23). Код скрипта про всяк випадок закоментовано символами <!-- щоб він не відображався в старих браузерах.

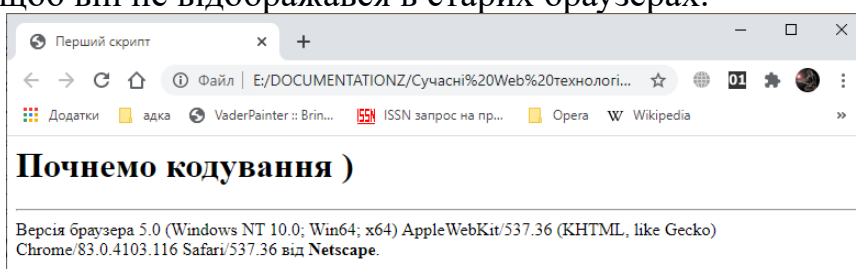


Рис. 23. – Результат праці кода виводу версії браузера

Для іменування об'єктів використовуються ідентифікатори тегів, наприклад <div id="mylove"></div>. А для отримання даних використовується наступна функція:

```
window.document.getElementById(mylove);
```

JavaScript для поділу елементів ієрархічного посилання використовує точку. Кожне посилання зазвичай починається із найбільш глобального рівня (document) і далі з використанням точки (.) як роздільника виконується все більша конкретизація посилання.

Розглянемо можливості роботи із властивостями об'єктів на прикладі тега `<input>`. Отже приклад:

```
<input type="button" id="clicker" name="clicker" value="Штовхни мене...">
<input type="text" id="entry" name="entry" value="Ваше Ім'я?">
```

Для отримання доступу до властивості об'єкта використовується той же синтаксис з використанням точок. Тільки він доповнюється ще додатковим розширенням що вказує на властивості. Наприклад:

```
document.getElementById("clicker").name
document.getElementById("entry").value
```

Якщо потрібно виконувати якісь дії з об'єктом в JavaScript використовуються методи, від властивостей відрізняються наявністю дужок. Розглянемо приклад двох методів:

```
document.getElementById("OrderForm").submit()
document.getElementById("entry").focus()
```

У першому випадку сценарій використовується для того щоб по клацанню відправити форму на сервер. У другому випадку виконується виділення тексту в текстовому полі з ім'ям `entry`.

Коли необхідне виконання певних дій у відповідь на роботу користувача **використовуються** обробники **подій**. Наприклад при натисканні на кнопку відображається повідомлення «Це неподобство!».

```
<form> <input type="button" value="натисни мене" onclick="window.alert('Це неподобство')"> </form>
```

Далі розглянемо приклад створення випадючих блоків, які можна використовувати для створення прихованих списків та спливаючих багаторівневих меню. Для цього створимо функцію `viewdiv`, та одну змінну, яку назвемо наприклад `el`. Лістинг коду скрипта та html коду наведено нижче:

Код HTML

```
<script type="text/javascript" src="menu.js"></script>
<div class="menu1">
<a href="javascript:void(0);"
onclick="viewdiv('mymenu');"><p>Перелік
переваг</p></a>
</div>
<div id="mymenu" style="display:none;">
<a href="#">Перший пункт</a>
<a href="#">Другий пункт </a>
<a href="#">Третій пункт</a>
</div>
```

Код JS (menu.js - файл)

```
function viewdiv(id){
var el=document.getElementById(id);
if(el.style.display=="block"){
el.style.display="none";
} else {
el.style.display="block";
}
}
```

Оскільки при вивченні даного курсу наголос становиться на отриманні професійних навиків програмування ми розглянемо останній приклад, що дозволяє істотно скрасити дизайн сайту та надати йому інтерактивності. У наступному прикладі створюється так званий слайдер. Тобто це блок, зміст котрого змінюється інтерактивно. Створимо блок що буде через рівний час змінювати графічне зображення із відповідної папки, усього три зображення. Наводимо приклад цього блоку:

Код HTML

```
<script type="text/javascript"
src="slider.js"></script>

<div class="slideshow-container">

<div class="mySlides fade">
  <div class="numbertext">1 / 3</div>
  
  <div class="text">Рисунок 1</div>
</div>

<div class="mySlides fade">
  <div class="numbertext">2 / 3</div>
  
  <div class="text"> Рисунок 2</div>
</div>

<div class="mySlides fade">
  <div class="numbertext">3 / 3</div>
  
  <div class="text"> Рисунок 3</div>
</div>

</div>
<br>

<div style="text-align:center">
  <span class="dot"></span>
  <span class="dot"></span>
  <span class="dot"></span>
</div>
```

Код JS (slider.js - файл)

```
var slideIndex = 0;
showSlides();

function showSlides() {
  var i;
  var slides =
document.getElementsByClassName
("mySlides");
  var dots =
document.getElementsByClassName("dot");
  for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }
  slideIndex++;
  if (slideIndex > slides.length) {slideIndex =
1}
  for (i = 0; i < dots.length; i++) {
    dots[i].className =
dots[i].className.replace(" active", "");
  }
  slides[slideIndex-1].style.display = "block";
  dots[slideIndex-1].className += " active";
  setTimeout(showSlides, 4000); // Change
image every 4 seconds
}
```

Отже щоб елемент виконував свою естетичну функцію необхідно додати стилі до відповідних елементів, подивитись на вихідний код можливо за посиланням: <https://codepen.io/darkness9991/pen/MWKGGZX>.

Завдання для виконання. Виконуючи цю лабораторну студент повинен створити: скрипт виводу інформації про браузер, блок що розгортається, слайдер на 3 зображення чи більше за вільним вибором.

Лабораторна робота 6

Фреймворків при створенні адаптивних сайтів.

Мета роботи: ознайомитись з основними принципами створення дизайну сайту з допомогою безкоштовного конструктору сайтів Mobirise.

В даний час набрали популярність односторінкові комерційні сайти націлені на продажу одного виду якоїсь продукції. Такі сайти мають дуже протяжні по вертикалі сторінки що прийнято називати Landing Page (лендінги).

Дані типи проектів характеризуються блоковою структурою і найчастіше при переході з одного блоку на інший використовуються анімаційні ефекти створені за допомогою мови програмування Java Script.

Таким чином для створення професійного сайту, що продає нам необхідні наступні фахівці: графічний дизайнер - розробка візуальної частини сайту, фахівець з верстки з використанням HTML і CSS, програміст JS для створення інтерактивних і динамічних елементів сайту. Відповідно для виконання робіт зі створення сайту потрібно певні витрати як засобів так і часу.

Засобом полегшити створення даних типів проектів стали інструментарії розробки сайтів, які являють собою якісь конструктори сайтів. Зі спеціальних блоків що входять до їх складу набираються графічні і функціональні елементи. При роботі із створення сайту на базі таких конструкторів не потрібно знати ні мови форматування, ні мови програмування.

Розглянемо безкоштовний конструктор сайтів Mobirise. При першому відкритті потрібно зареєструватися за допомогою електронної пошти або соціальної мережі. Після авторизації і відкриття програми інтерфейс прийме наступний вигляд (рис. 24).

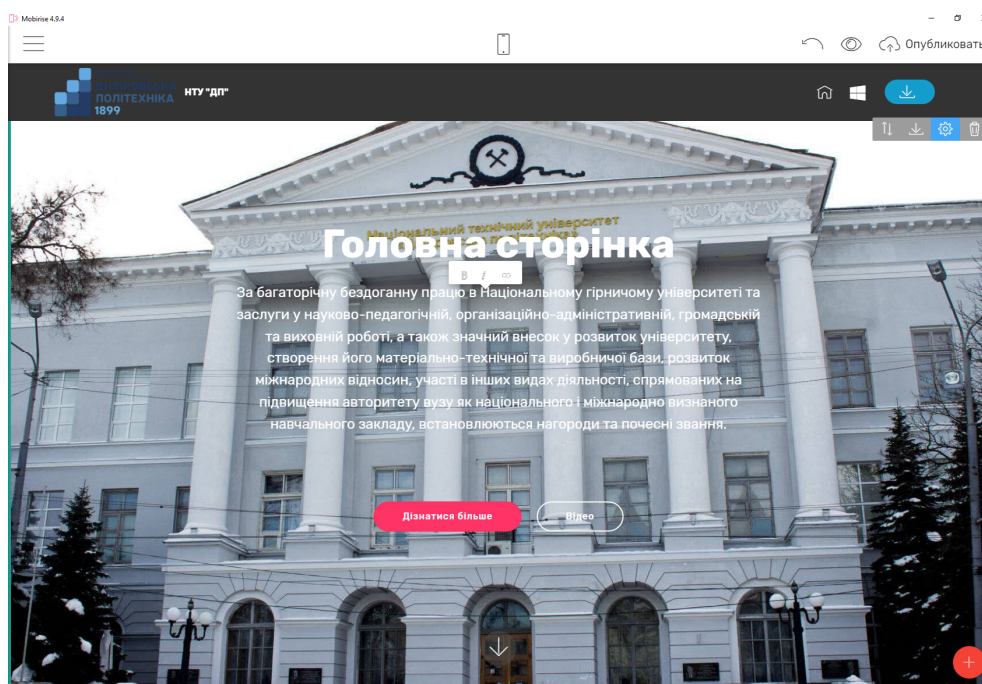


Рис. 24. Початкова тема конструктора сайтів




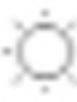
У вікні програми завантажиться початкова тема для сайту що складається з трьох блоків: верхньої шапки з кнопками, середньої частини з фоновим зображенням і двома кнопками посередині і нижній частині містить контактну інформацію.


Кожен з блоків і його елементів можна редагувати, для цього потрібно клацнути на потрібний елемент лівою кнопкою миші і якщо це блок то в правому верхньому кутку з'явиться керуюче меню що містить елементи представлені в таблиці 2, якщо це елемент то з'явиться вікно налаштувань елемента, приклад в таблиці 3.

Таблиця 2. Керуючі елементи блоків сайту

			
Переміщення блоку по сторінці	Зберегти блок в базу	Налаштування параметрів блоку	видалення блоку

Таблиця 3. Керуючі елементи текстової частини

			
вставка посилання	Зміна кольору	Вибір стилю шрифту / елемента	Вставка іконки з бібліотеки

Для виведення всіх керуючих елементів додатка потрібно зайти в меню програми натиснувши на кнопку  в верхньому лівому кутку програмного інтерфейсу. Далі необхідно вибрати пункт «Учетная запись» після чого пункт «Настройка приложения». В панелі, що з'явилася активуйте всі функції і натисніть застосувати.

В даному меню є пункт «Расширения», зайшовши в нього можна встановити ряд безкоштовних блоків (соціальні мережі, карти, аналітика та ін.), А також придбати платно більш функціональні модулі, наприклад редактор коду, який дозволяє змінювати будь-які блоки за бажанням.

У пункті меню «Сайти» можливо створювати різні сайти і зберігати їх дизайн в відповідно створених для цього директоріях (рис.25).

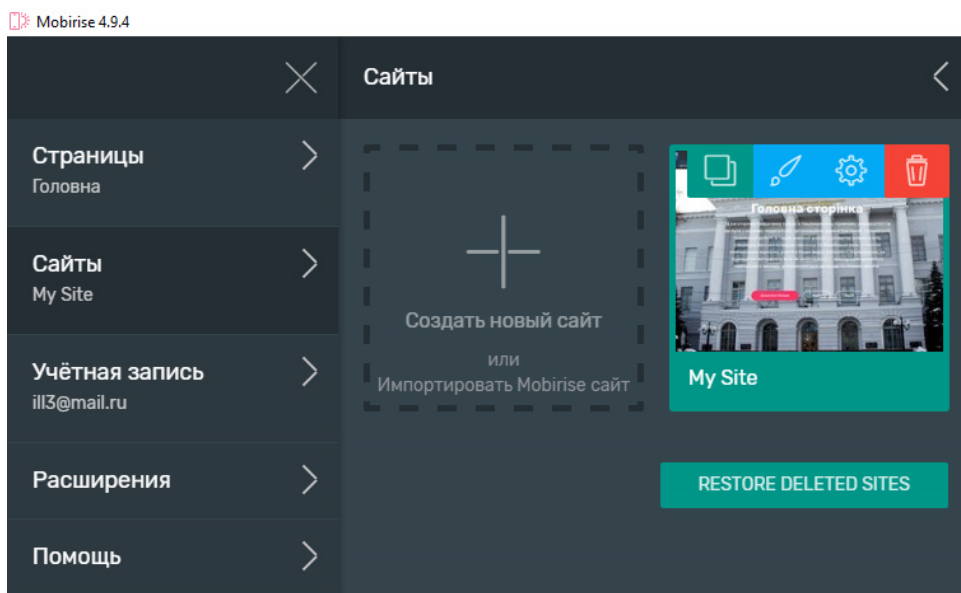


Рис. 25. Пункт главного меню «Сайты»

У пункті меню «Страницы» можливо створювати різні сторінки для сайту зі своїм унікальними дизайном, якщо сайт має містити кілька сторінок (рис.26). У властивостях сторінки можна змінювати заголовок сторінки, він же `<Title>`, а також опис сторінки, тобто `<Meta name = description>`.

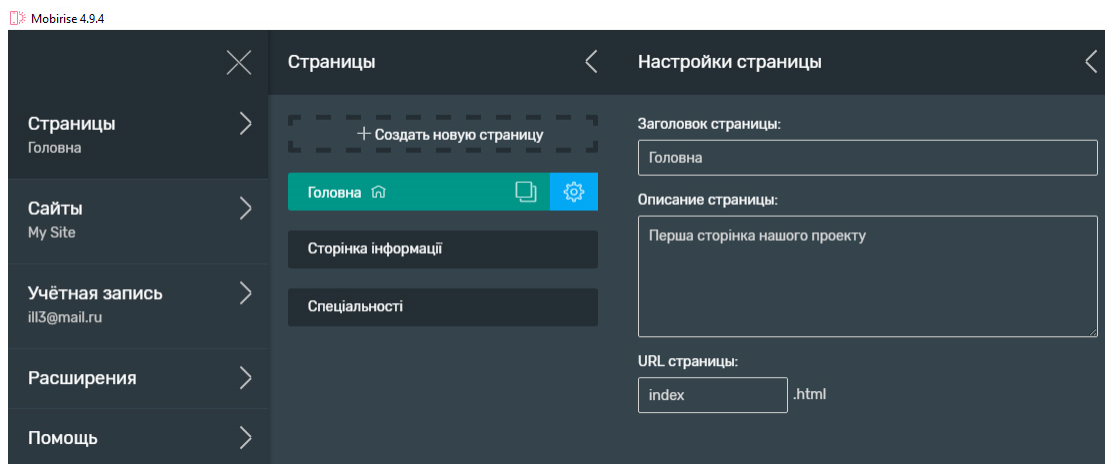


Рис. 26. Пункт главного меню «Страницы»

Основне оформлення дизайну сайту здійснюється за допомогою двох кнопок розташованих в нижньому правому куті – зміна стилів що є при натисканні на кнопку з пензликом. Вставка блоків сайту здійснюється при натисканні на кнопку з хрестом (рис.27).

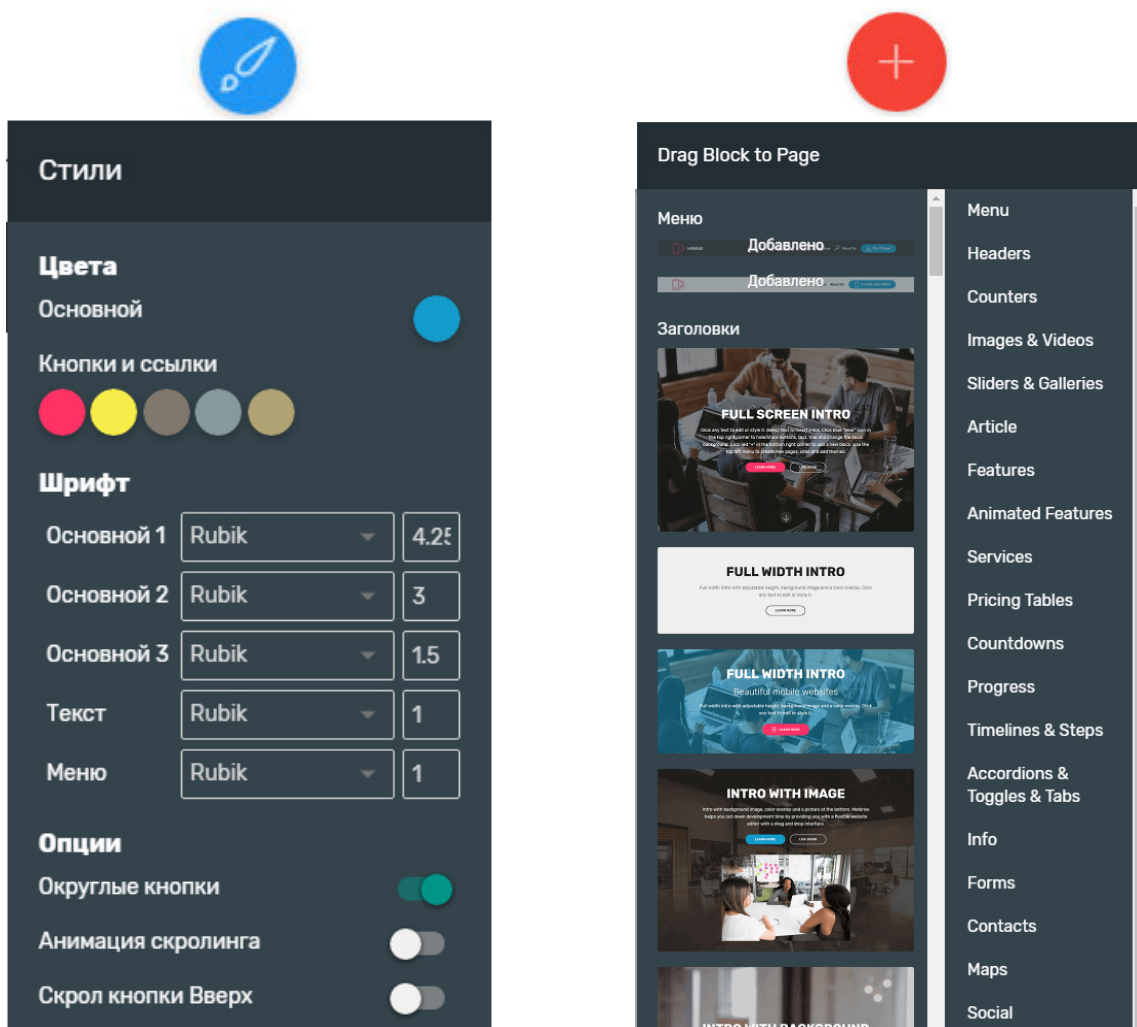


Рис. 27. Основні елементи управління дизайном сайту

Більшу частину блоків можна оформляти використовуючи в якості фону: колір, фонове зображення, відео з YouTube.

Завдання для виконання. Створіть сайт із однієї сторінки, що містить відомості про ваше хобі, протяжністю мінімум в 10 блоків. Обов'язкова наявність наступних елементів: карти Google, кнопки соціальних мереж, і наявність відео на задньому плані або в якості елемента спеціального блоку.

Лабораторна робота 7

Встановлення систем керування контентом CMS.

Мета роботи: ознайомитись з інструментарієм необхідним для встановлення та використання сучасних систем керування контентом CMS.

Потреба користувачів в засобах автоматизації web-ресурсів постійно зростає. Автоматизації розробки web-ресурсів можна досягти за рахунок використання так званих систем управління контентом - Content management system (CMS) - це системи, які підтримують створення, управління, розподіл, розміщення і розвиток загальної інформації. Вони покривають весь життєвий цикл сторінок на сайті, від надання простих інструментів для створення контенту і його розміщення до архівування. Також надають можливість управляти структурою сайту, дизайном сторінок і навігацією.

В даний час перспективним є інтерактивна взаємодія з учнями за допомогою інформаційних комунікаційних мереж, з яких масово виділяється середовище інтернет-користувачів.

Головна функція CMS – показувати сторінки сайту користувачам, формуючи їх вміст "на льоту" із заздалегідь визначених шаблонів з дизайном і контенту, тобто текстів, картинок, таблиць і інших матеріалів, які зберігаються в базі даних.

Інша важлива функція CMS – допомогти власнику сайту без якихось спеціальних навичок керувати сайтом, тобто публікувати нові сторінки, новини, викладати відео, робити посилання на зовнішні ресурси і так далі.

Важливо розуміти, що сайту як набору сторінок при такій схемі просто не існує. Є окремо дизайн (шаблони) і окремо набір різних матеріалів – текст, картинки, файли з архівами, документи MSOffice/PDF і інші матеріали. CMS створює сторінку користувачу в момент його запиту. При цьому, в залежності від ситуації, користувачеві може бути показана якась унікальна інформація, яка більше нікому і ніколи не буде видно. Наприклад, вміст його кошика в інтернет-магазині. Цю роботу і робить CMS, або "движок сайту". При цьому, CMS не просто генерує сторінки користувачам. Вона намагається зробити це якомога швидше, щоб обробити максимальну кількість запитів в одиницю часу, захищає дані від недобросовісних користувачів, не дає засмічувати базу даних спамерам і робить у фоновому режимі масу додаткової роботи, яку не видно відвідувачам сайту і навіть його власнику, але життєво необхідна для нормальної роботи сучасного сайту.

За способом розповсюдження системи управління бувають вільно-розповсюджуваними і платними. Разом з останніми в переважній більшості випадків клієнти отримують супровід і підтримку.

Узагальнена типова схема роботи сайту, який використовує CMS, представлена на рис. 28.

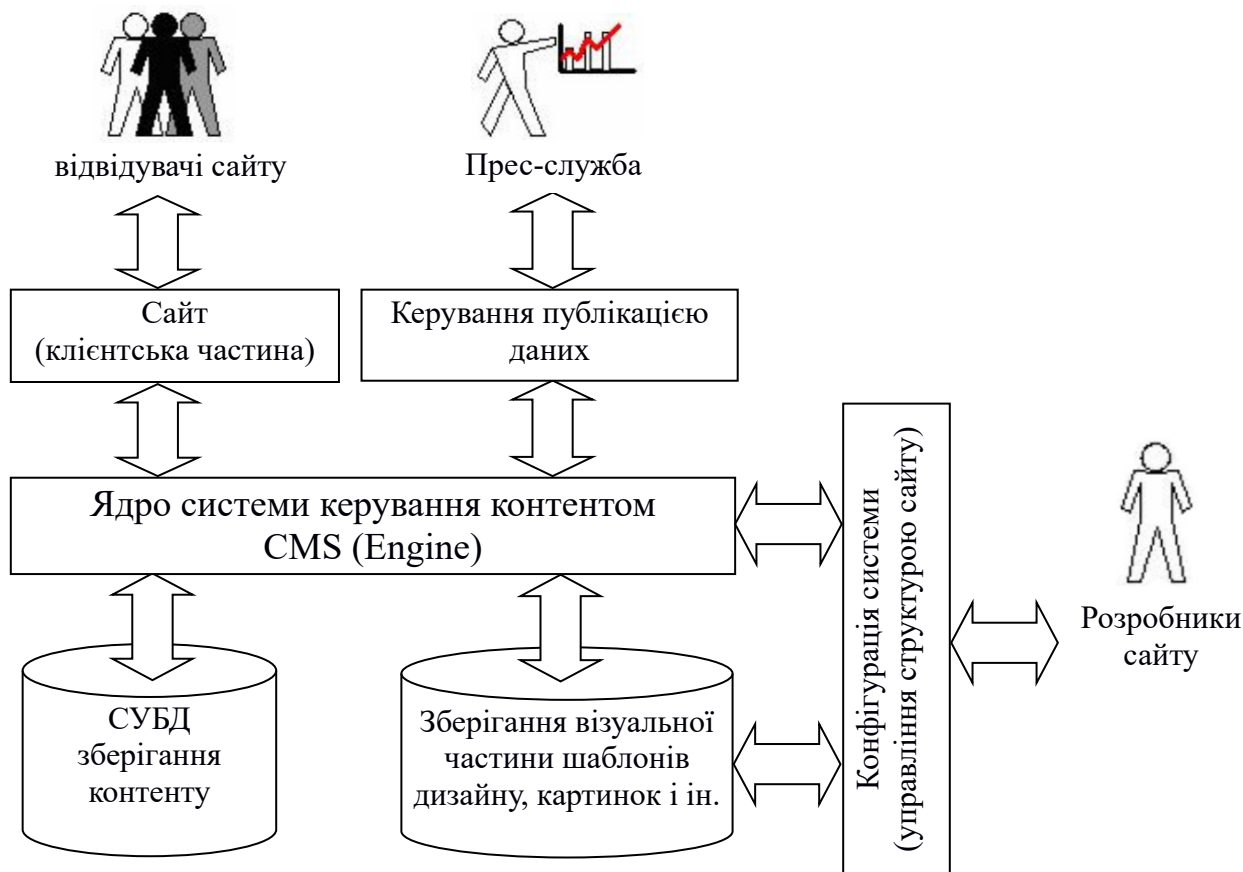


Рис.28. Узагальнена схема роботи CMS

В системі присутні два сховища. У першому (зазвичай реляційна СУБД) зберігаються всі дані, які публікуються на сайті. У другому (зазвичай файлова система) зберігаються елементи уявлення - шаблони, графічні зображення і т.д. Крім зовнішнього представлення сайту, яким його бачать всі користувачі, є як мінімум кілька спеціалізованих робочих місць. Одне робоче місце - для розробників сайту. З його допомогою вони задають структуру сайту, структуру контенту, визначають зовнішній вигляд сайту, налаштовують шаблони подання інформації. Цей інструментарій зазвичай не повністю автоматизований. Для настройки сайту розробники частково працюють через засоби CMS, частина інформації розміщується безпосередньо. Інше робоче місце - для власників сайту. Воно дозволяє співробітникам компанії самостійно розміщувати інформацію на сайті, без участі розробників. Менеджери замовника працюють тільки через спеціалізоване робоче місце.

На базі сучасної CMS може існувати проект практично будь-якої складності, наприклад: сайт візитка підприємства, новинний блог, медійний портал, сайт електронної комерції (інтернет магазин), соціальна мережа і ін.

Велика частина сучасних CMS пишуться на мові програмування PHP, з впровадження коду JS. При цьому основна інформація зберігається в базі даних SQL. В якості засобів захисту системи можуть використовувати додаткові модулі і засоби шифрування. Отже для використання CMS необхідний веб-сервер з встановленими сервісами: інтерпретатори мов програмування, сервери баз даних і ін. в залежності від складності системи.

Установка локального веб-сервера

Встановлювати і налаштовувати CMS відразу на інтернет сервері вважається поганим тоном. Зазвичай встановлюється закритий або локальний веб-сервер без доступу в мережу і налаштовується система, дизайн і різні її модулі. Після цього налаштована система переміщується фахівцями на зовнішній веб-сервер для функціонування.

У нашій роботі ми встановимо локальний веб-сервер скориставшись збіркою XAMPP (раніше ми користались джентельменським набором Web-розробника «Денвер», але останні роки він не підтримується). Для чого необхідно закатати актуальну версію з сайту розробника <https://www.apachefriends.org/>.

Сам набір складається з дистрибутивів (локальний сервер [WAMP](#)) і програмного середовища, що призначені для створення і налагодження сайтів (веб-додатків, іншого динамічного вмісту інтернет-сторінок) на локальному ПК (без необхідності підключення до мережі [Інтернет](#)) під керуванням ОС [Windows](#).

До складу набору входить веб-сервер Apache, з найбільш затребуваними сервісами PHP, MySQL, Perl, SSL і ін.

Після завантаження пакета його необхідно встановити. Необхідно запустити закатаний дистрибутив. У разі якщо система задовольняє вимогам почнеться установка. Під час установки можливо буде обрати необхідні сервіси (рис. 29). Для виконання лабораторних необхідними будуть наступні сервіси: Apache, MySQL, PHP, phpMyAdmin. Інші сервіси за бажанням можна не встановлювати. У наступному вікні буде надано можливість вказати директорію для установки, за умовчанням це C:\xampp.

Після встановлення із головного меню ОС запустить панель керування сервером (рис. 30) XAMPP Control Panel v... Запустить сервіси Apache та MySQL. Відкрийте інтернет оглядач та наберіть наступний URL: <http://localhost/>. У випадку якщо встановлення було вдале у оглядачі відкриється сторінка інформації щодо сервера (рис. 31). Перевірте за допомогою верхнього горизонтального меню чи працює PHPInfo та phpMyAdmin. Якщо ці сервіси активні то основні сервіси коректно встановлені.

Налаштування віртуального хоста

Для того щоб здійснювати доступ до локального сайту можна задати для нього домен. Зміст сайту зберігається в теці сервера C:\xampp\htdocs.

Отже для створення нового сайту (домен ntudp.com) нам потрібне буде:

- 1) Зупинити всі сервіси за допомогою кнопок Stop в панелі керування;
- 2) Створіть теку для домена у теці C:\xampp\htdocs. (наприклад назвіть ntudp.com). Перевірте чи існує відповідний путь, а саме C:\xampp\htdocs\ntudp.com.
- 3) Для того щоб доменне ім'я шукалось не у інтернеті, а на локальній машині додайте наступну строку
127.0.0.1 ntudp.com
до файлу налаштувань ОС C:\windows\system32\drivers\etc\hosts



4) Додайте налаштування хоста у файл C:\xampp\apache\conf\extra\httpd-vhosts.conf

```
<VirtualHost *:80>
    DocumentRoot "C:/XAMPP/htdocs/ntudp.com"
    ServerName ntudp.com
    ServerAlias ntudp.com

    ErrorLog "logs/pacificsky.log"
    CustomLog "logs/pacificsky.log" common
</VirtualHost>
```

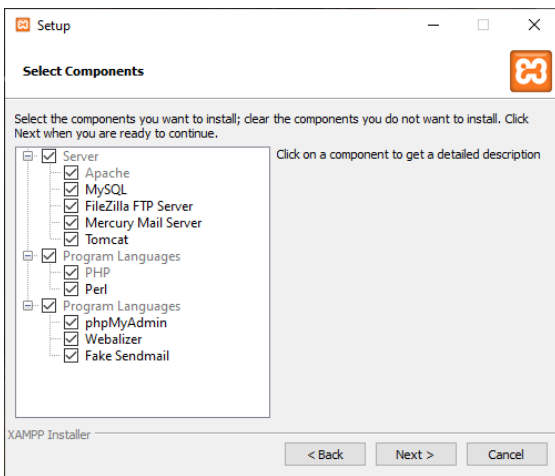


Рис.29. Вибір сервісів під час установки

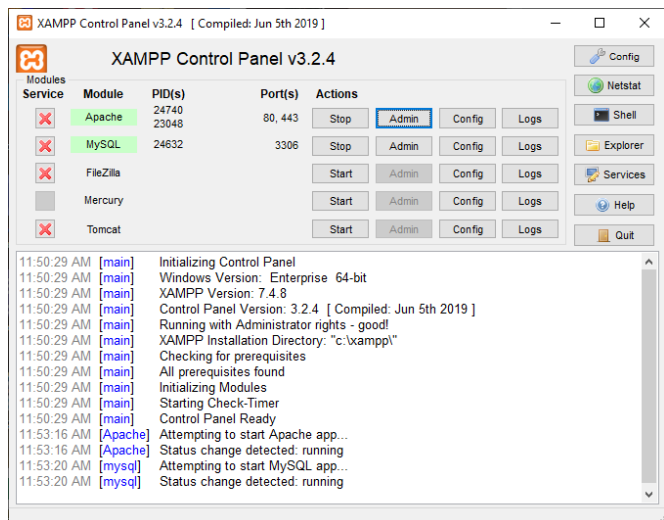


Рис.30. Панель керування сервером

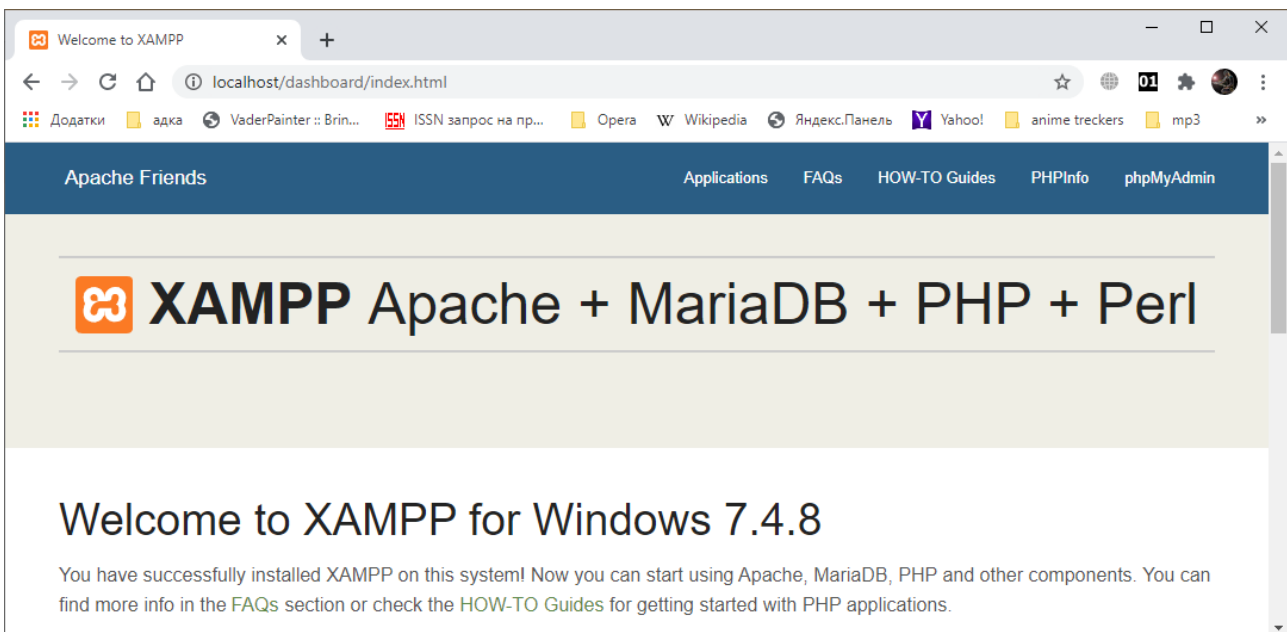


Рис. 31. Загрузка начальной сторінки встановленого сервера



Після налаштування віртуального хоста запустить сервіси у панелі керування сервером. Зайдіть на сайт ntudp.com через інтернет оглядач. Якщо відкриється пуста тека без помилок – налаштування хоста вдале (рис. 32).

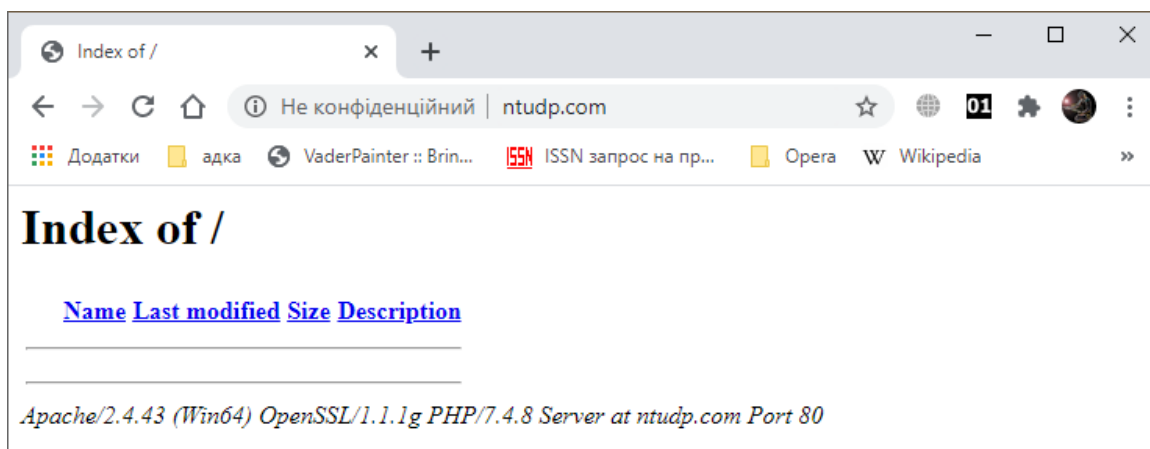


Рис. 32. Загрузка начальної сторінки віртуального хоста

Установка системи управління контентом

Розглянемо установку CMS на базі Wordpress. Принципово установка будь-якої системи управління сайтом буде мало відрізнятися. Потрібно зайти на сайт розробника (<https://wordpress.org/download/>) і закачати архів актуальної версії системи. Отриманий архів необхідно розпакувати в папку `C:\WebServers\home\ntudp.com\www`.

Оновлюємо в браузері сторінку ntudp.com, завантажиться вікно початку установки з проханням вибрати мову установки. Виберіть потрібну мову і натисніть кнопку продовжити (рис.33).

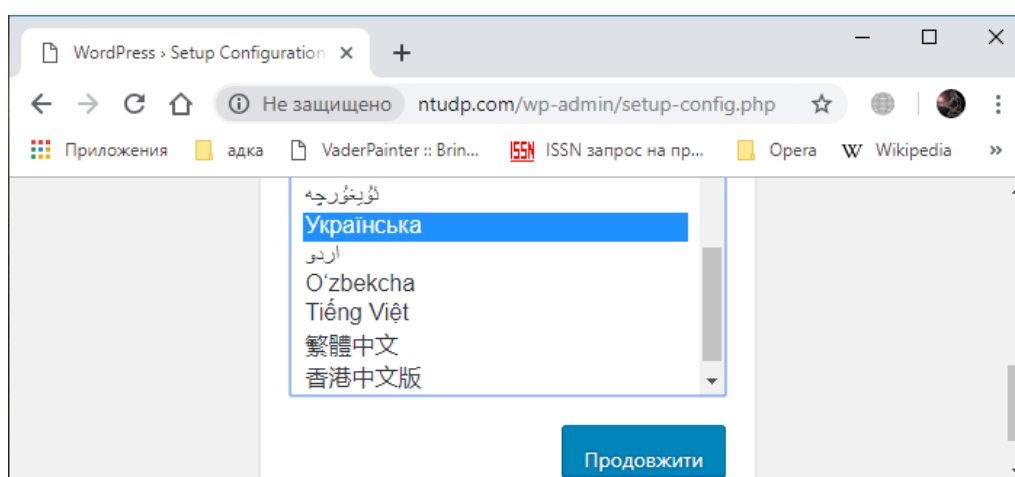


Рис.33. Вікно початку установки CMS

Наступна сторінка повідомляє про установки які будуть необхідний в подальшому для установки системи. Відсутні на даний момент дані це: ім'я бази даних, ім'я користувача бази, пароль бази.

Відкриємо сторінку <http://localhost/phpmyadmin/> панель керування phpMyAdmin. Відкриється інтерфейс управління базами даних (рис.34).

Перейдіть на вкладку баз даних. Створіть базу даних з ім'ям ntudp для зберігання даних системи управління нашим сайтом. Користувача можна в навчальному проєкті не створювати, так як в системі є користувач адміністратор з повними правами управління будь-якою базою даних. Цей користувач називається root і у нього відсутній пароль.

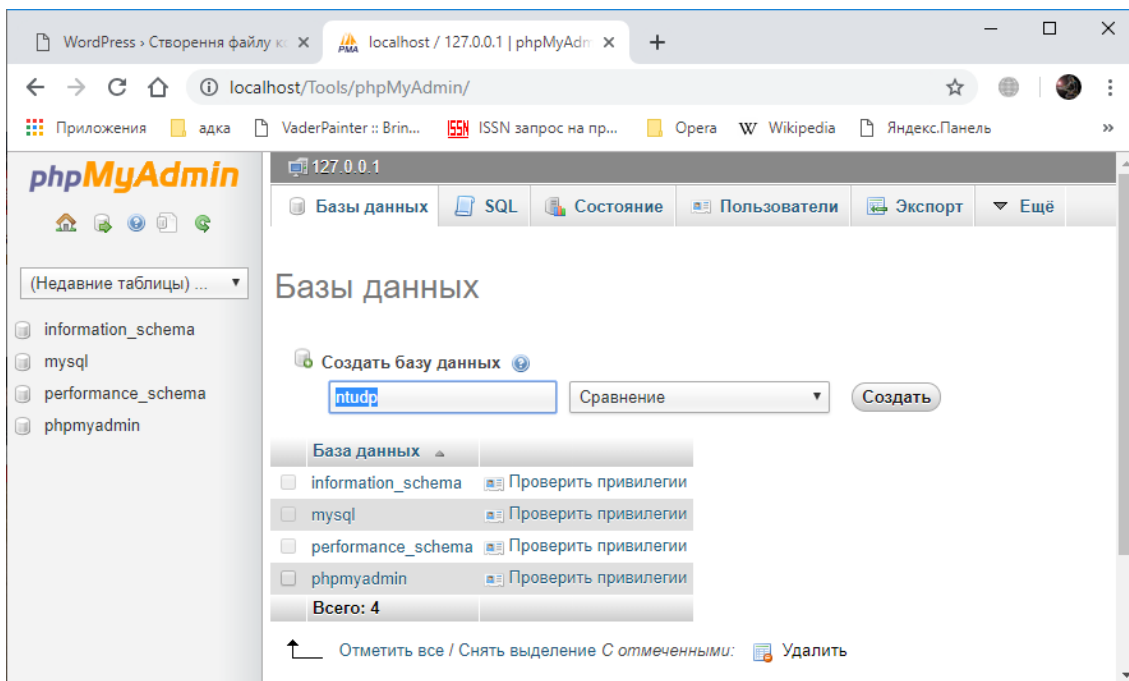


Рис.34. Створення бази даних

Перейдемо в вікно установки CMS і перейдемо до наступної сторінки. Введемо дані заповнивши форму (рис.31). Перейдемо на наступну сторінку установки. Якщо дані були коректно заповнені система запросить дані для адміністративного облікового запису, а саме ім'я користувача, пароль та електронну адресу. Також необхідно ввести назву для сайту, його можна буде пізніше змінити в разі потреби.

Після успішної установки системи відкриється вікно входу в адміністративну частину за адресою <http://ntudp.com/wp-login.php>. Потрібно вказати дані які вводилися раніше для входу в систему. Відкриється адміністративний розділ сайту, який видно тільки адміністраторам системи і не видно зовнішнім користувачам сайту (рис.35).

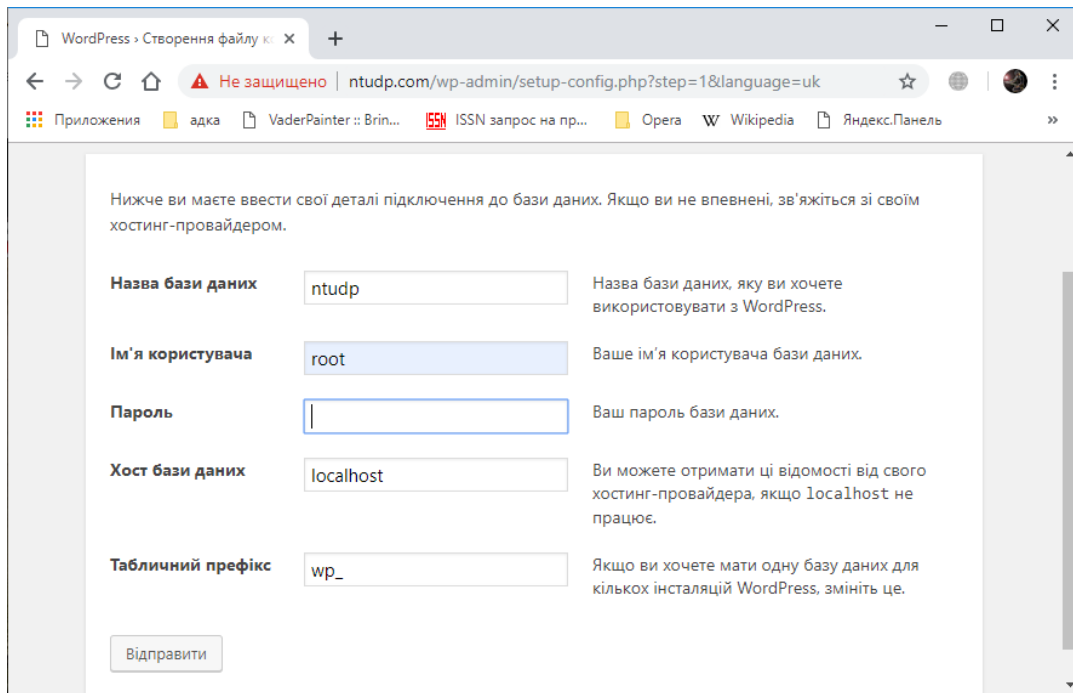


Рис.35. Установка інформації для бази даних

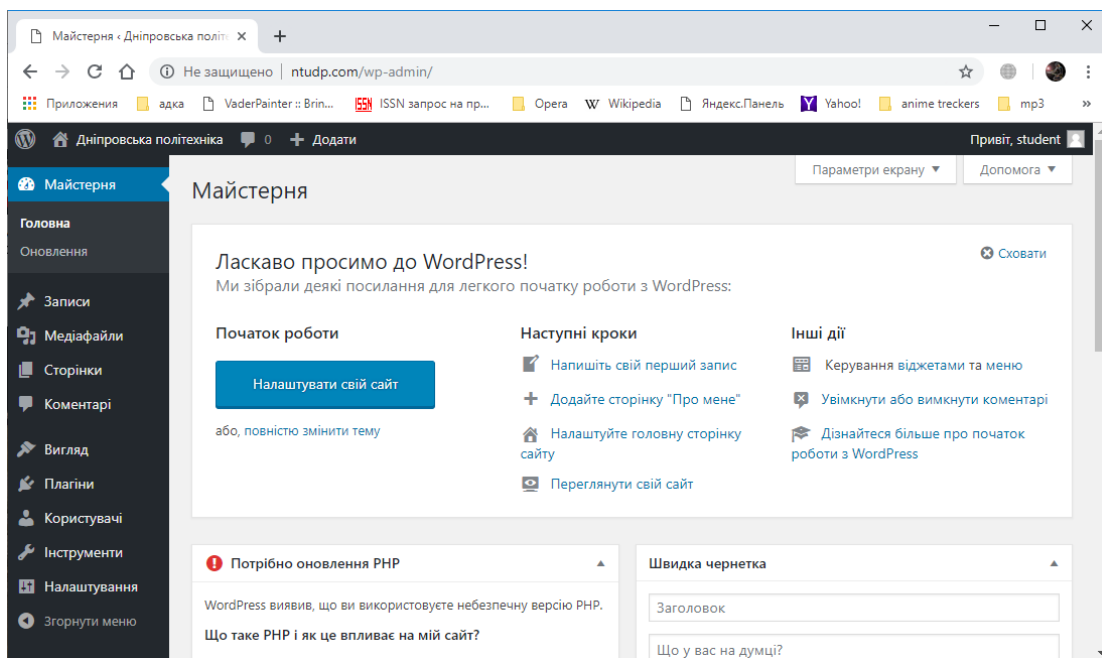


Рис.36. Зона адміністрування сайту

Введіть в адресний рядок <http://ntudp.com/> і подивіться як виглядає оформлення сайту встановлене за замовчуванням (рис.37).

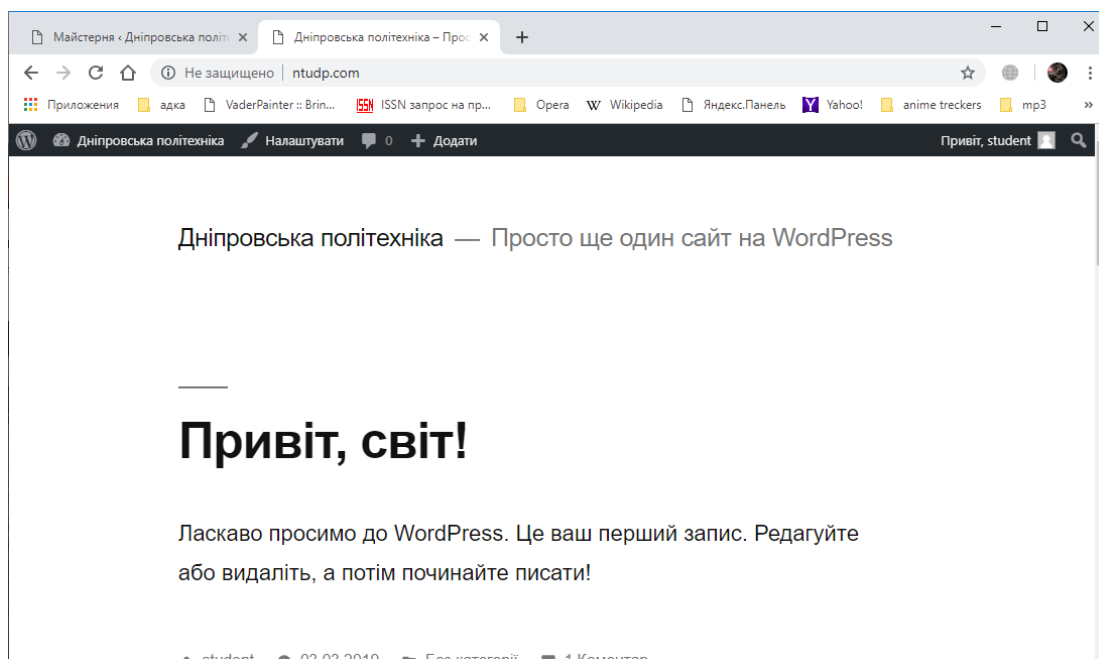


Рис.37. Зовнішній вигляд сайту з темою за замовчуванням

Завдання для виконання. Використовуючи набуті під час виконання лабораторних робіт уміння створіть сайт на базі CMS.

Лабораторна робота 8 Налаштування дизайну та можливостей CMS WordPress.

Мета роботи: ознайомитись з налаштуванням графічної частини сайту, розглянути розширення можливостей системи встановлюючи додаткові доповнення.

За даними сервісу W3Techs третину всіх сайтів в інтернеті використовують систему управління контентом WordPress. WordPress - це найпопулярніша в світі безкоштовна програма з відкритим вихідним кодом для створення сайтів і подальшого забезпечення їх роботи. Відкритий вихідний код движка, дозволяє підключити до розвитку проекту «колективний розум», що забезпечує йому активний і стабільний розвиток. Будь-який програміст, досвідчений чи, при бажанні та вмінні, може взяти участь в цьому процесі.

Спочатку WordPress замислювався як програма в якій може розібратися будь-який невіддодований користувач. Такою вона залишається і зараз.

На базі даної системи можна створювати лендінги, корпоративні сайти, інтернет-магазини, професійні портфоліо і багато іншого.

Налаштування системи управління контентом

Відкривши панель управління сайтом для вибору основних параметрів необхідно активувати пункт меню налаштувань - основні (рис.38). В даному пункті можна налаштувати назву сайту <Title>. Опис сайту <Meta name =

description>, URL доступу до сайту, пошту адміністратора куди буде приходити звітність роботи системи, можливість реєстрації користувачів в системі і їх права за замовчуванням, годинна зона, мова сайту, формат висновки дат і часу, початковий день тижня.

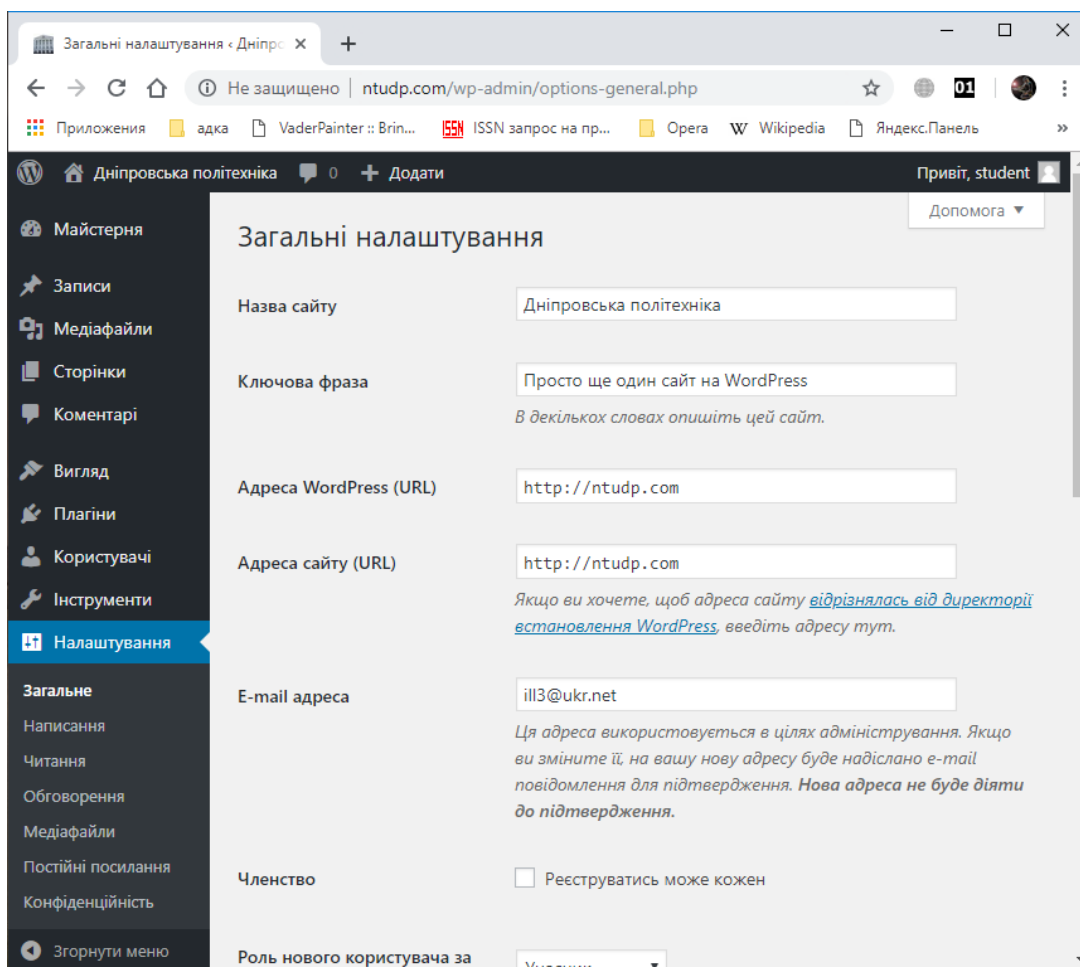


Рис.38. Основні настройки системи управління контентом

Використовуючи такі пункти меню можна задати поштові сервера для розсилок, вказати сторінку в якості головної при завантаженні сайту, встановити кількість статей виводяться на відповідних блоках, налаштувати можливості з коментування матеріалів, задати розміри для медіафайлів, налаштувати тип посилань в адресному рядку та інше.

Налаштування дизайну CMS

Система використовує шаблонну систему побудови дизайну. Дизайн - тема (шаблон) WordPress - це набір службових файлів, які перебудовують дизайн сайту, повністю змінюючи його зовнішній вигляд. Найчастіше шаблони розробляються професійними дизайнерами.

Теми для WordPress досить різноманітні. В залежності від уподобань і специфіки сайту можна вибрати тему з кількома колонками, з фіксованими або гумовими розмірами, в світлих або темних тонах, з різними відтінками кольору і так далі.

Перед установкою і активацією нової теми опублікуйте кілька записів і сторінок для об'єктивної оцінки теми - чи підійде вона сайту.

За замовчуванням в систему вже вбудовано декілька різних тем оформлення сайту, але активною може бути тільки одна. Для активації іншої теми необхідно натиснути на пункт меню зовнішній вигляд (рис.39).

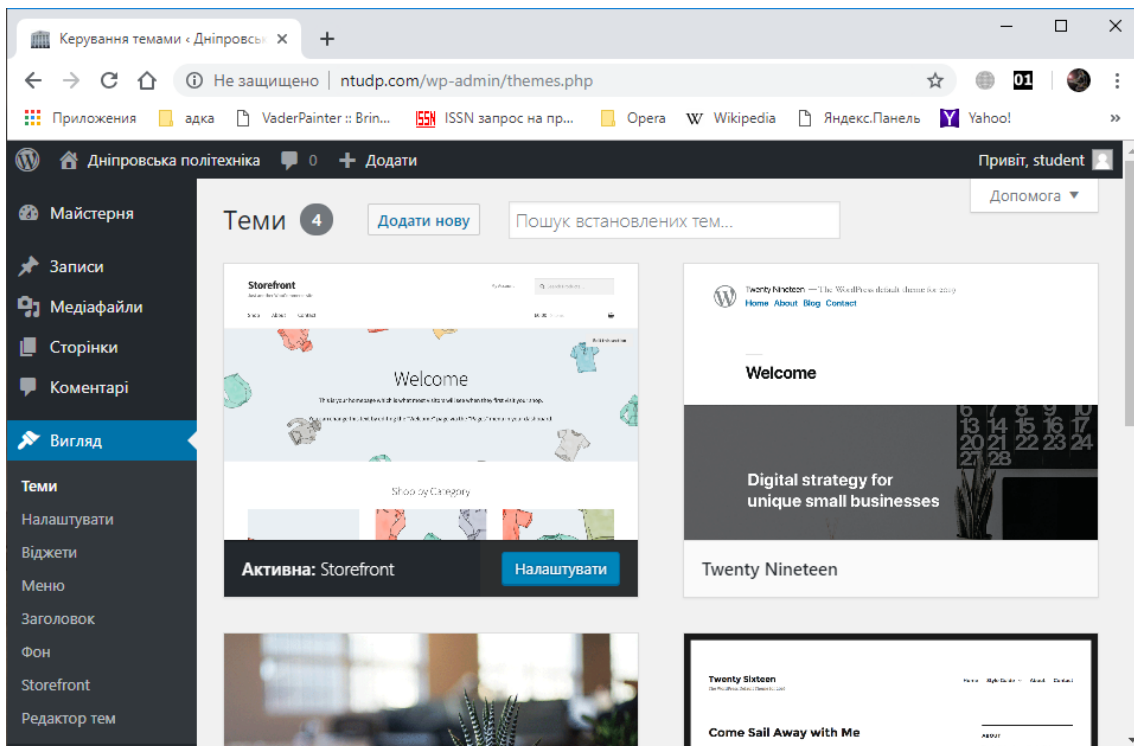


Рис.39. Вибір встановленої теми

Якщо шаблони оформлення встановлених тем не підходять під концепції сайту можна встановити нові теми. Для цього необхідно натиснути кнопку додавання тем. На сторінці, за допомогою системи пошуку тем за ключовими словами можна вибрати необхідну з оформлення і функціональними можливостями (рис.36). Кожну тему перед установкою можна переглянути натиснувши на відповідну кнопку, або перейти на сайт виробника конкретної теми і прочитати повний опис і можливості конкретного шаблону оформлення.

Одна з причин, по якій багато хто використовує WordPress, полягає в тому, що платформа дає доступ до тисяч красивих шаблонів під найрізноманітніші типи сайтів. Наприклад, якщо ви плануєте створити нішевий сайт, скажімо, для сайту будівельної фірми або салону краси, то зможете швидко і легко знайти потрібний вам шаблон, наприклад, на [ThemeForest](https://www.themeforest.net).

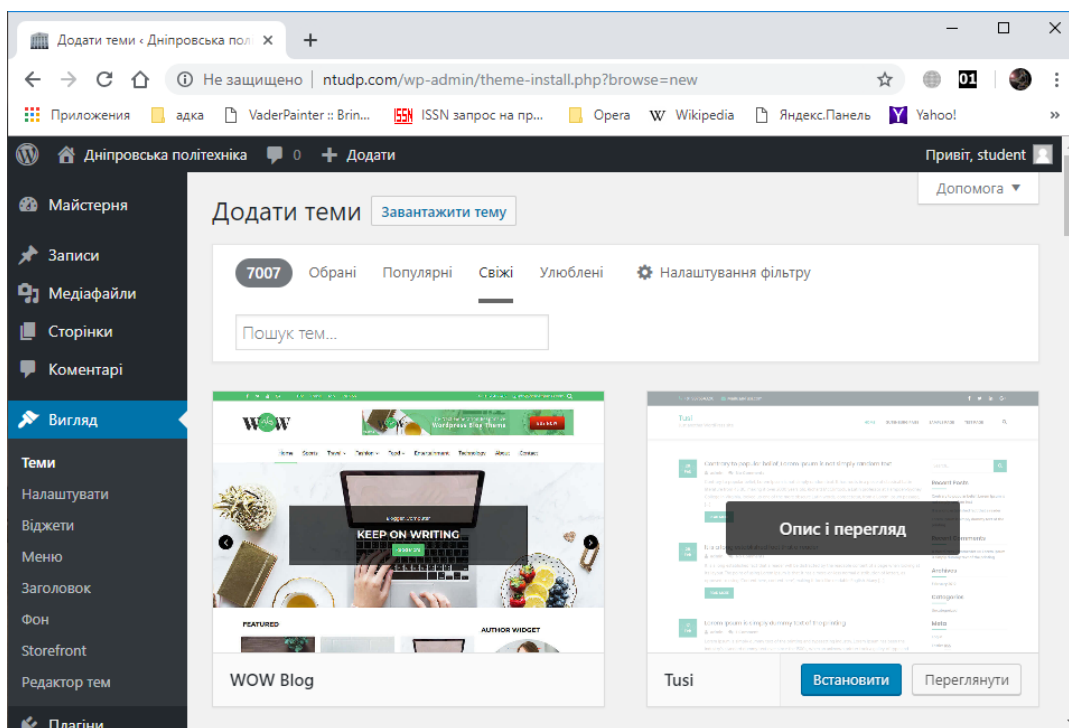


Рис.40. Установка нової теми оформлення сайту

Для настройки дизайну активної теми необхідно скористатися пунктом налаштувань меню оформлення. Відкриється вікно налаштувань активної теми (рис.41). У різних тем можуть бути і різні настройки, і свій функціонал.

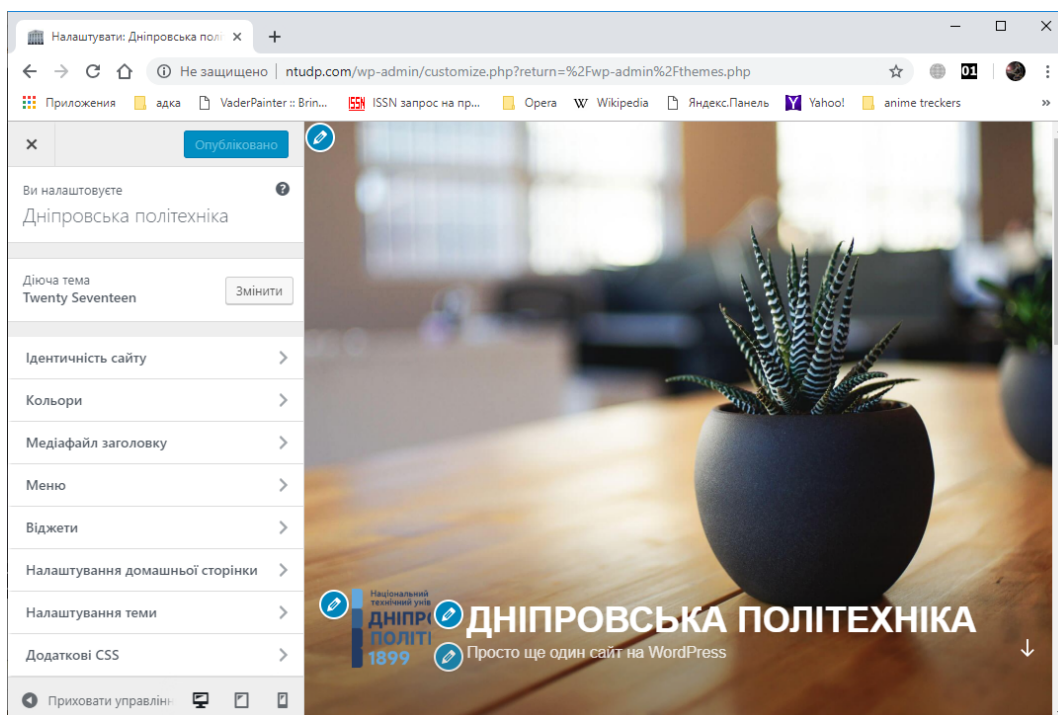


Рис.41. Налаштування активної теми оформлення сайту

Перший пункт дозволить поставити логотип і фон або фонове зображення. В наступному пункті можна змінити кольори елементів оформлення. Трохи далі можна попрацювати з додаванням медійних елементів і пунктів меню.

Істотно підвищити зручність користувача при роботі (юзабіліті) дозволяють спеціальні блоки з певним функціоналом, які можна розміщувати на різних місцях дизайну сайту. У ряді шаблонів присутні свої блоки, їх називають віджети. Але існує і ряд загальних.

Таким чином *віджет* – це блок що можливо налаштувати для виконання своєї певної функції на сайті. Це може бути вивід меню рубрик, форми пошуку або підписки на сайти, реклами, різна інформація з показом погоди, курсу валют, анекдотів, ... і багато іншого.

Області приміщення віджету в WordPress визначаються шаблоном (дизайн-темою) сайту. Їх може бути і одна, і п'ять, і більше. Це можуть бути бічні колонки (сайдбар) з основними і додатковими областями, підвал (футер) з декількома областями і ін.

Для налаштувань віджетів в головному меню необхідно вибрати пункт оформлення - віджети (рис.42).

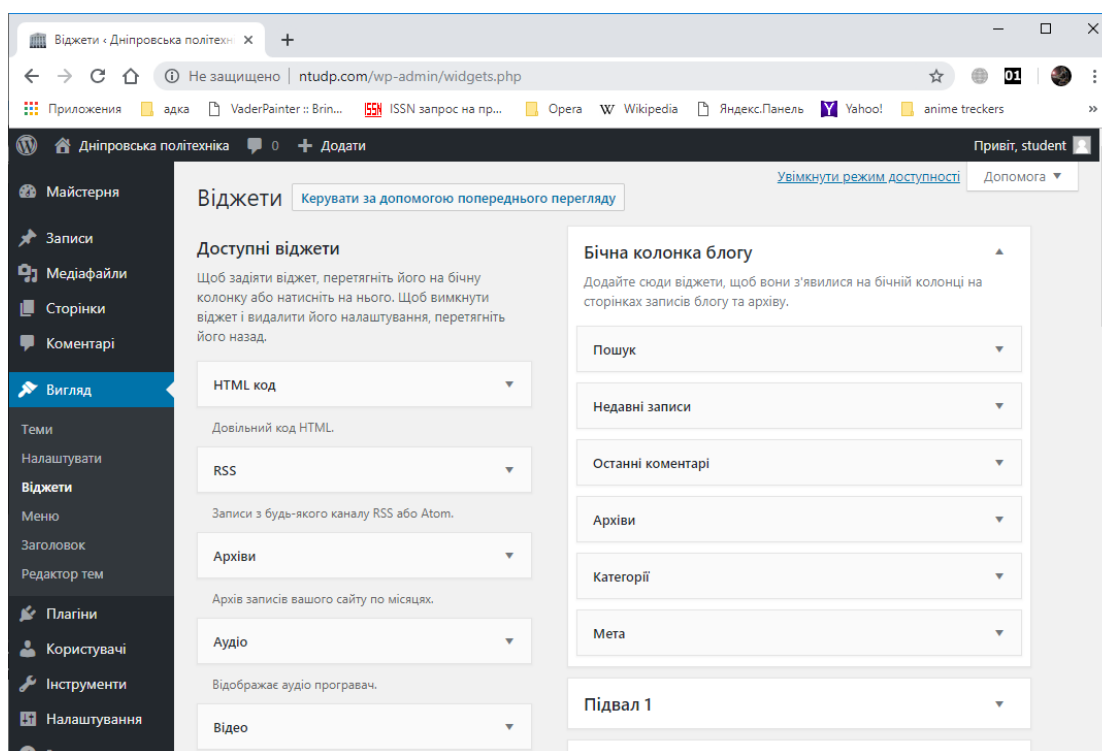


Рис.42. Налаштування віджетів для активної теми оформлення сайту

Останнім пунктом меню оформлення є редактор тем. При його активації відкриється текстовий редактор що дозволяє редагувати каскадні таблиці стилів і шаблон оформлення сайту в ручному режимі. Таким чином фахівці з верстки можуть докорінно змінити дизайн сайту під свої потреби.

Розширення функціоналу сайту за допомогою плагінів

Плагін (Розширення) – це самодостатній, обслуговуючий себе функціональний модуль, невелика і зазвичай легка програма, яка підключається до WordPress, і служить для розширення функціональних можливостей останнього.

Модулі WordPress дозволяють істотно розширити функціонал сайту. Використовуючи плагіни, можна створити контактну форму для свого сайту, додати систему оплати, поліпшити оптимізацію для пошукових систем, додати додатковий шар безпеки і навіть створити лендінги – і це все за кілька клацань. Існує понад [43 000 плагінів](#), доступних в офіційній базі даних WordPress. Так що є велика ймовірність того що вирішити поставлену задачу можна буде успішно.

Розглянемо на прикладах установку і використання плагінів які найбільш затребувані в комерційній діяльності підприємств.

Установка плагіна масової розсилки

Найбільш ефективний спосіб для впливу на клієнтів - це, без сумніву, e-mail-розсилка. З її допомогою можна побудувати довірчі відносини з передплатниками, бути на зв'язку з покупцями, повертати незадоволених споживачів, які припинили користуватися послугами компанії.

Величезний плюс такого методу спілкування полягає в тому, що вам не потрібно вкладати гроші, і в більшості випадків e-mail-розсилка авторизована.

Для установки нового плагіна скористаємося пунктом меню плагіни і натиснемо кнопку додати новий. Здійснимо пошук потрібного плагіна ввівши ключове слово newsletter (рис.43).

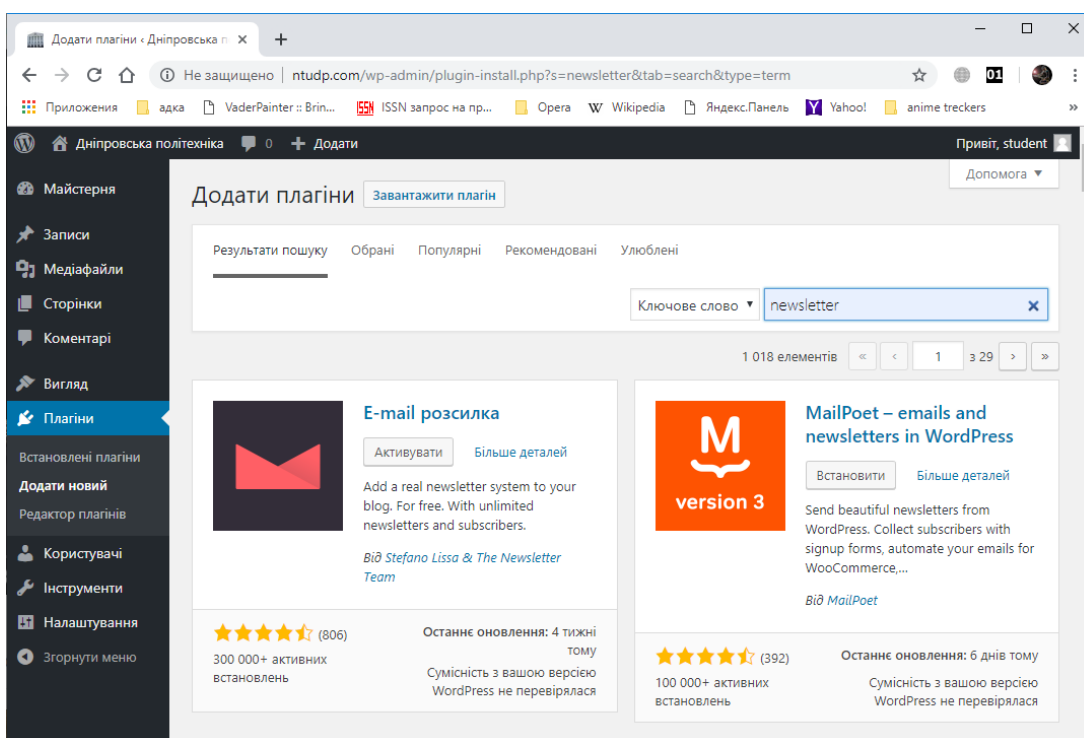


Рис.43. Додавання нового плагіна

У вікні відобразяться відповідні плагіни для роботи з розсилками. Перед установкою найкраще ознайомитися з їх можливостями, натиснувши на кнопку детального опису. Переглянути їх функціональні і системні вимоги. Також необхідно оцінити якість плагіна, це можна зробити подивившись на кількість установок даного плагіна і прочитавши відгуки про нього.

Після установки плагін необхідно активувати, тільки після цього його налаштування з'являться в меню управління сайтом (рис.44).

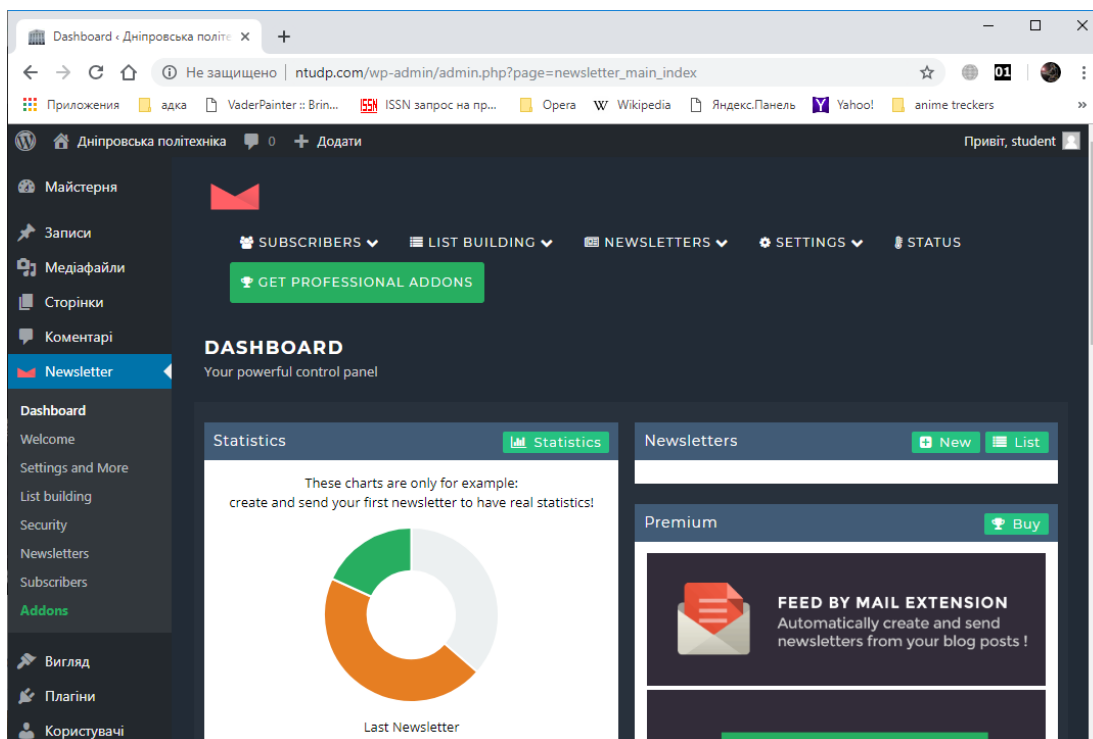


Рис.44. Налаштування плагіна розсилки

Налаштувавши всі поля в даному модулі ми зможемо створювати масові розсилки листів, оформлення в досить професійному стилі. Бази електронних адрес для розсилки можна імпортувати, використовуючи текстові файли. Також можна встановити на сторінках сайту блок, в якому користувачам буде запропоновано підписатися на розсилки.

Безкоштовна версія даного плагіна дозволяє оцінити якість поштової розсилки даючи статистику про успішності відправки пошти адресату, про прочитання або не прочитання листа, а також про кліки по посиланнях що містяться в листі.

Установка плагіна електронної комерції

Є багато додатків з розвитку інтернет-магазинів пов'язаних з WordPress. Але, мабуть, найбільш популярним і ефективним є WooCommerce. WooCommerce почав свою роботу у вересні 2011 року і за перші чотири роки роботи був завантажений інтернет-користувачами близько 10,2 мільйонів разів. Додаток може похвалитися значною цифрою в 30% від всієї частки ринку електронної комерції. Така цифра опублікована на їх офіційному сайті. Плагін розроблений в середовищі WooThemes, і підтримується нею ж. Як і інші додатки з відкритим вихідним кодом, WooCommerce можна завантажити безкоштовно. Але його якісна робота вимагає доповнень, які можуть бути як безкоштовними, так і на платній основі. Додаток може інтегруватися з будь-яким іншим додатком WordPress, але найкращим чином працює і оптимізується з додатками WooThemes.

Основний функціонал системи містить наступні опції:

- WordPress класу SEO;
- доступ до п'яти платіжних шлюзів;
- варіації продукту;
- гео-локація (автоматичне включення розрахунку суми податку, розрахунок суми доставки і регіональне обмеження з продажу);
- управління запасами товару;
- коди і купони на знижку;
- акаунти для клієнтів;
- опціональний гостьовий контроль;
- відгуки про продукт включаючи клієнтську підтримку;
- базова звітність та аналітика;
- мобільний доступ і адміністрування.

WooCommerce має значний набір функцій. Серед цього набору також є підтримка цифрових завантажень, онлайн-контент і підписки. Додаток досить систематично і оперативно виводить всі звіти. Ви в реальному часі маєте змогу бачити кількість товару, ціну доставки, ціну податків тощо. У той час як ці функції достатні (але не надто розширені) для нових людей в сфері e-commerce, додаток пропонує платний розширений функціонал для впевнених користувачів.

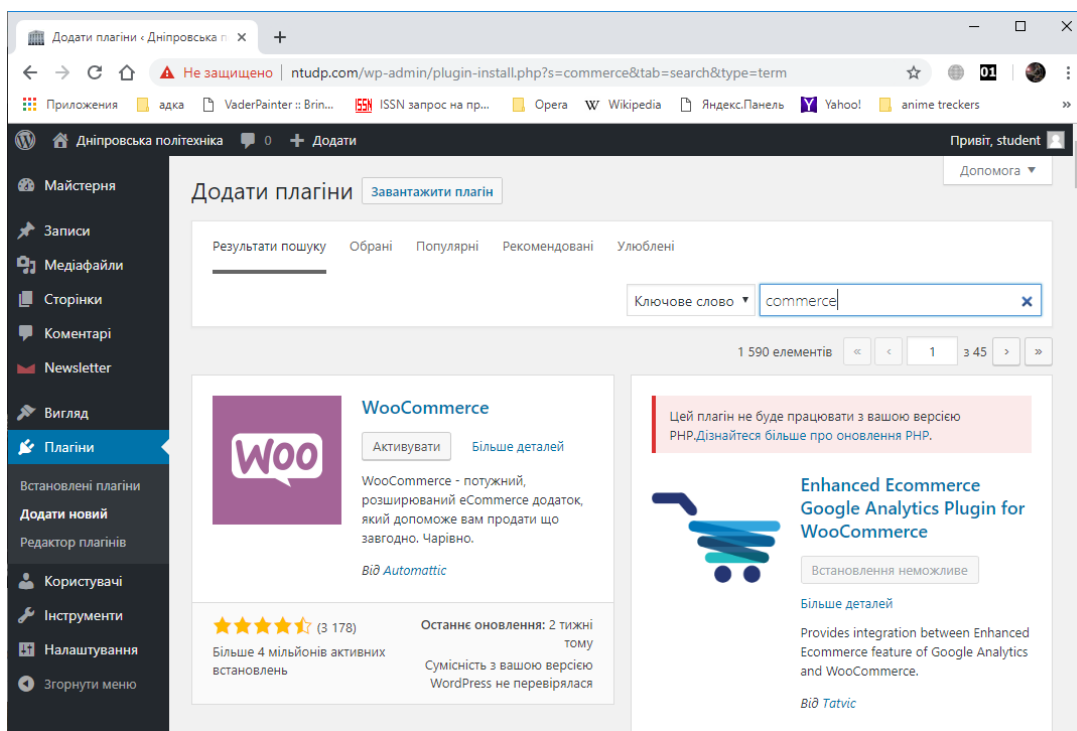


Рис.45. Додавання плагіна магазину

Для пошуку плагіна в вікні добавки нового плагіна в поле ключових слів введіть commerce (рис.45). Після фільтрації плагінів роботи з електронною комерцією встановіть, а потім активуйте необхідний. На пропозицію додати в базу товарів тестові відповідайте ствердно. Побачити як виглядають категорії товарів і тестові товари можна використавши меню налаштувань оформлення -

настройка теми - WooCommerce - каталог товарів (рис.46). Редагування товарів і категорій здійснюється в основній консолі через пункт WooCommerce - Товари.

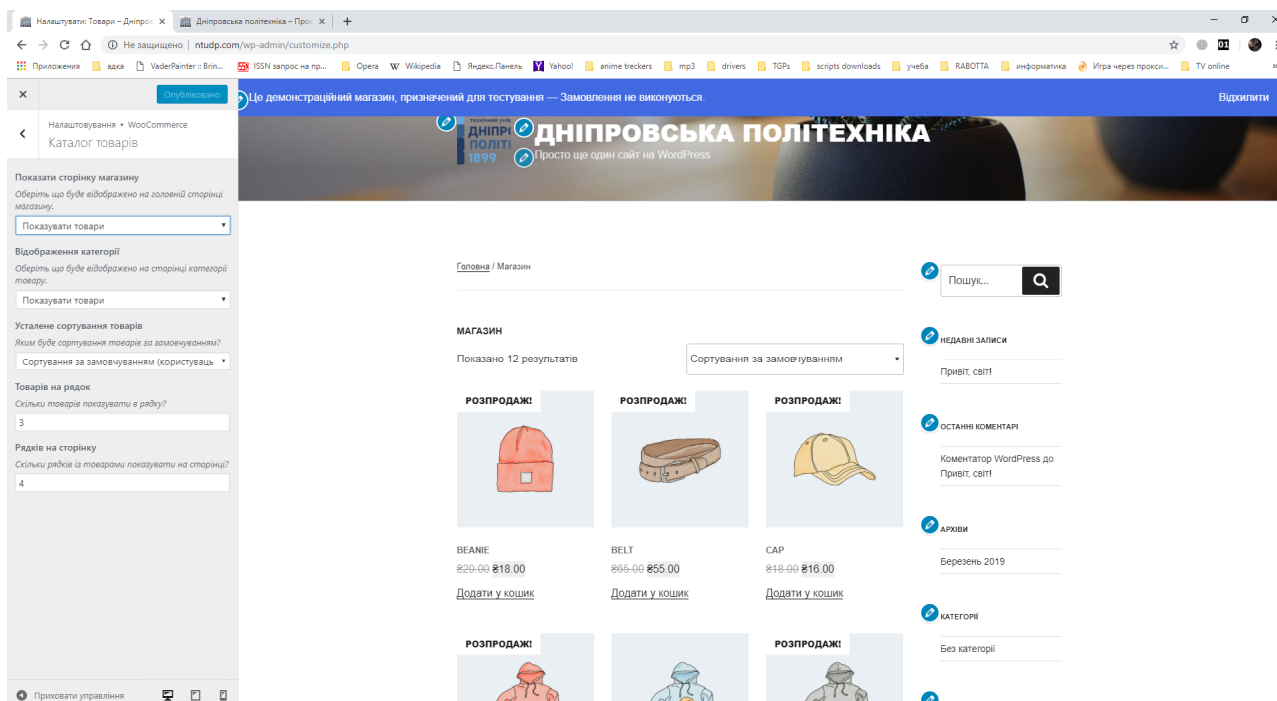


Рис.46. Налаштування каталогу товарів магазину

Завдання до виконання. Встановити систему Wordpress та плагіни листування та електронного магазину. Додати 5 своїх товарів для нової категорії.

Лабораторна робота 9 Публікація сайту у глобальній мережі.

Мета роботи: ознайомитись із системою керування хостингом та отримати навички з налаштувань та встановлення сервісів потрібних для публікації сайту у мережі.

Основні засоби для публікації сайту

Після розробки дизайну сайту, налаштування системи управління сайтом з розробленим дизайном, наповнення сайту необхідною інформацією (сторінками, розділами, документами та іншим), перевіркою працездатності сайту і усунення всіх помилок сайт необхідно опублікувати в глобальній мережі.

Для публікації сайту необхідно орендувати простір на сервері, що має постійне підключення до інтернет і доступний 24/7. Простір під сайт на сервері називається *хостингом*, а компанії, що надають місце під сайт на своїх (або орендованих) серверах називаються *хостинг провайдерами*. Оплачувати хостинг можна як помісячно, так і на більш тривалі терміни. Зазвичай при

оплаті хостингу на 3 і більше місяців відразу дається знижка в 5%, а при оплаті на рік знижка може бути 20%.

Для доступу до сайту просто орендувати місце недостатньо. Необхідно щоб користувачі знали як потрапити на сайт. Для доступу до сайтів зазвичай використовується URL основною частиною якого є *домен*. Наприклад [HTTP://ntudp.com](http://ntudp.com). В даному випадку доменом є ім'я `ntudp.com` – це домен комерційного підприємства другого рівня за класифікацією доменів [1]. Відповідно організації що здають в оренду домени називаються реєстраторами доменних імен або *домен провайдерами*. Домени можна орендувати терміном на 1 або більше років.

Хостинг можна придбати в однієї організації, а домен в іншій. Кожен інтернет провайдер надає також зазвичай послуги по хостингу та реєстрації доменів. Але на території України і країн ближнього зарубіжжя потужності сервером обмежені, нові сервіси та операційні системи оновлюються не завжди своєчасно. Більш якісні послуги надають організації з серверами, що знаходяться в розвинених європейських країнах і США. На них і слід орієнтуватися при виборі хостингу.

Робота з панеллю керування хостингом

Величезна частина інтернет серверів працює під управління операційних систем на базі Unix/Linux отже ми будемо орієнтуватися саме на них.

Для комфортної роботи користувача з хостингом були створені спеціальні програми, звані системами управління хостингом. Вони мають приємний графічний веб-інтерфейс, за допомогою якого і здійснюється вся робота по налаштуванню хостингу.

Існують платні системи і системи з відкритим вихідним кодом, що надаються безкоштовно. Перерахуємо кілька панелей управління хостингом в порядку їх популярності:

cPanel

Розробник: компанія cPanel Inc.

Перша версія: 1996 год.

ISP Manager

Розробник: компанія [ISP System](#).

Перша версія: 1997 год.

Plesk Panel

Розробник: компанія [SWSOFT](#), пізніше [Parallels](#).

Перша версія: 2003 год.

Vesta CP

Розробник: компанія Vesta.

Перша версія: 2012 рік.

[Direct Admin](#)

Розробник: компанія IBM Software.

Перша версія: 2003 год.

ISPConfig

Розробник: projektfarm GmbH.



Перша версія: 2007 рік.

Ми будемо орієнтуватися на найбільш популярну систему, а саме cPanel. Розглянемо докладніше її можливості.

Загальні характеристики

cPanel багатомовна і підтримує 25 мов, в тому числі українську. Має у своєму розпорядженні інтуїтивно зрозумілий інтерфейс.

З панеллю встановлено основне необхідне для роботи сайту ПО: Apache, MySQL, PHP, exim.

Кожне меню cPanel включає відео-інструкції і посилання на документацію від розробників.

Управління доменами

Підтримує додавання і видалення додаткових, паркованих доменів і піддоменів, редагування DNS-записів, настройку перенаправлень.

Управління файлами

Містить диспетчер файлів, який підтримує пошук по імені файлу і три способи редагування файлів (у вигляді тексту, коду, HTML), web-диск.

Управління базами даних

Зручний функціонал управління базами даних MySQL і PostgreSQL: базові опції створення, видалення [бази даних](#), Користувачів, настройка прав користувача, доступ в phpMyAdmin, phpPgAdmin, дозвіл віддаленого доступу до бази даних.

Робота з поштою

Широкі можливості налаштувань обробки пошти та адміністрування користувачів: сценарії автоматичної конфігурації популярних поштових клієнтів, доступ до 3 клієнтів веб-пошти (Horde Webmail, SquirrelMail і RoundCube), настройка фільтрів пошти, спам-фільтр, настройка списків розсилки, імпорт поштових адрес.

Налаштування безпеки

Включають функціонал управління SSL-сертифікатами, захист папок сайту паролем, захист Hotlink (від несанкціонованого використання прямих посилань на ваш сайт), обмеження доступу до сайту по IP-адресою.

Резервне копіювання

Створення резервної копії можливо тільки в ручному режимі. Є опція створення повного або часткового бекапа файлової системи, бази даних та налаштувань облікового запису cPanel. Підтримується стиснення резервної копії.

FTP

Функціонал створення, видалення і управління користувачами FTP, визначення квоти FTP-акаунта, управління активними сеансами FTP. Є готові конфігурації основних FTP-клієнтів (FileZilla, CoreFTP, Cyberduck).

Журнали та статистика

Журнал доступу, журнал помилок. Меню статистики поточного споживання ресурсів. Засоби аналізу статистики відвідувань: AWstats, Webalizer, а також аналізу трафіку.

Практичне використання системи керування сайтом

Всі перераховані функції доступні з відповідних груп вкладок всередині панелі управління. Для отримання доступу до панелі управління хостинг провайдер видає посилання на адміністративну сторінку, яка складається зазвичай з домена і порту підключення, наприклад: <http://домен:2082>.

Після введення даного посилання в рядок браузера відкриється вікно авторизації користувача. Туди необхідно ввести дані користувача і пароля, отримані від хостинг-провайдера. Після авторизації відкриється головна сторінка панелі управління (рис. 47).

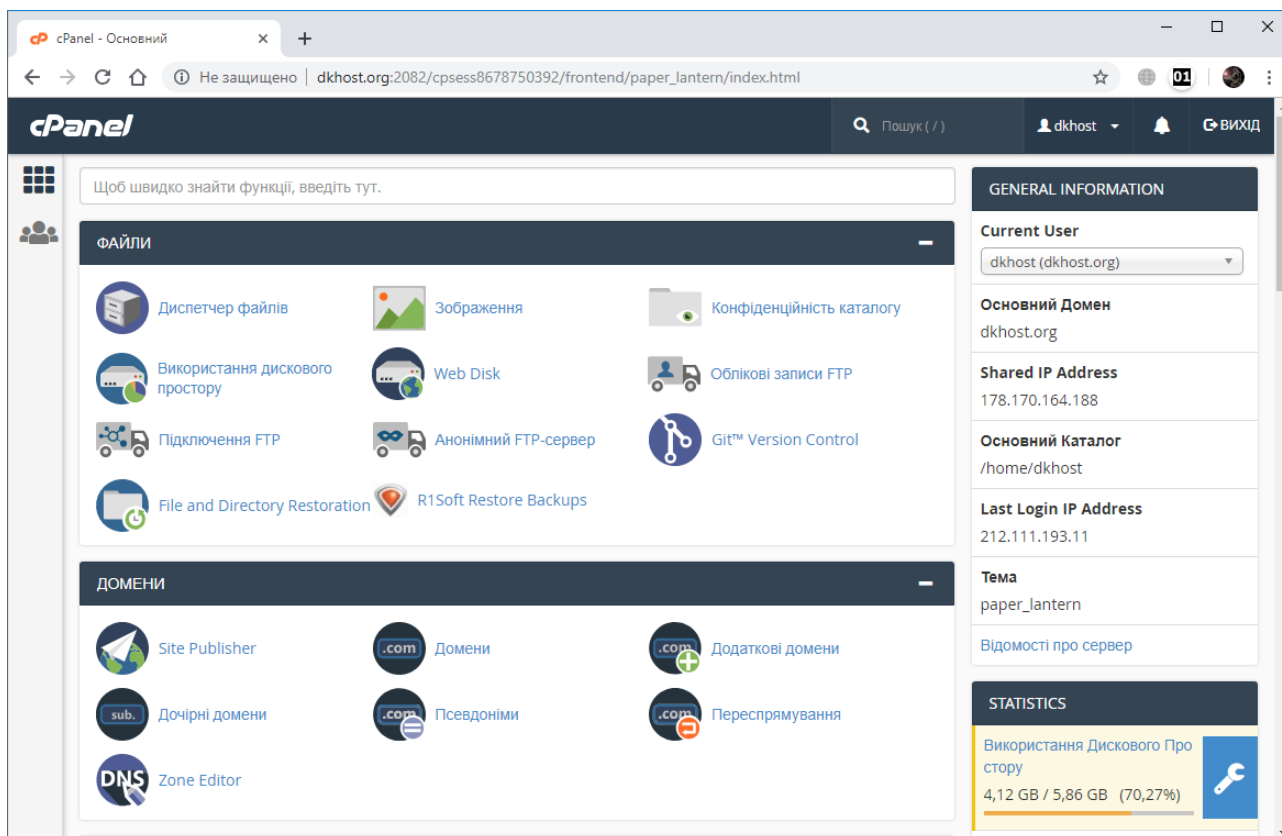


Рис.47. Головна сторінка панелі керування

Для підключення домену купленого в іншого реєстратора в групі Домени необхідно використовувати пункт Додаткові домени. У наступному вікні треба вказати назву домену, теку для зберігання сайту. Зазвичай це коренева тека що має назву `public_html`, якщо підключається ще один або декілька доменів необхідно для них створити теки для зберігання сайтів, відповідно це будуть адреси `public_html/папка1`, `public_html/папка2` і так далі. Якщо необхідно щоб під різними доменами відображався один і той же сайт, то потрібно вказати однаковий шлях до однієї і тієї ж директорії декількам доменів. Приклад підключення домену `avanguard-fitness.com` продемонстрований на рис. 48.

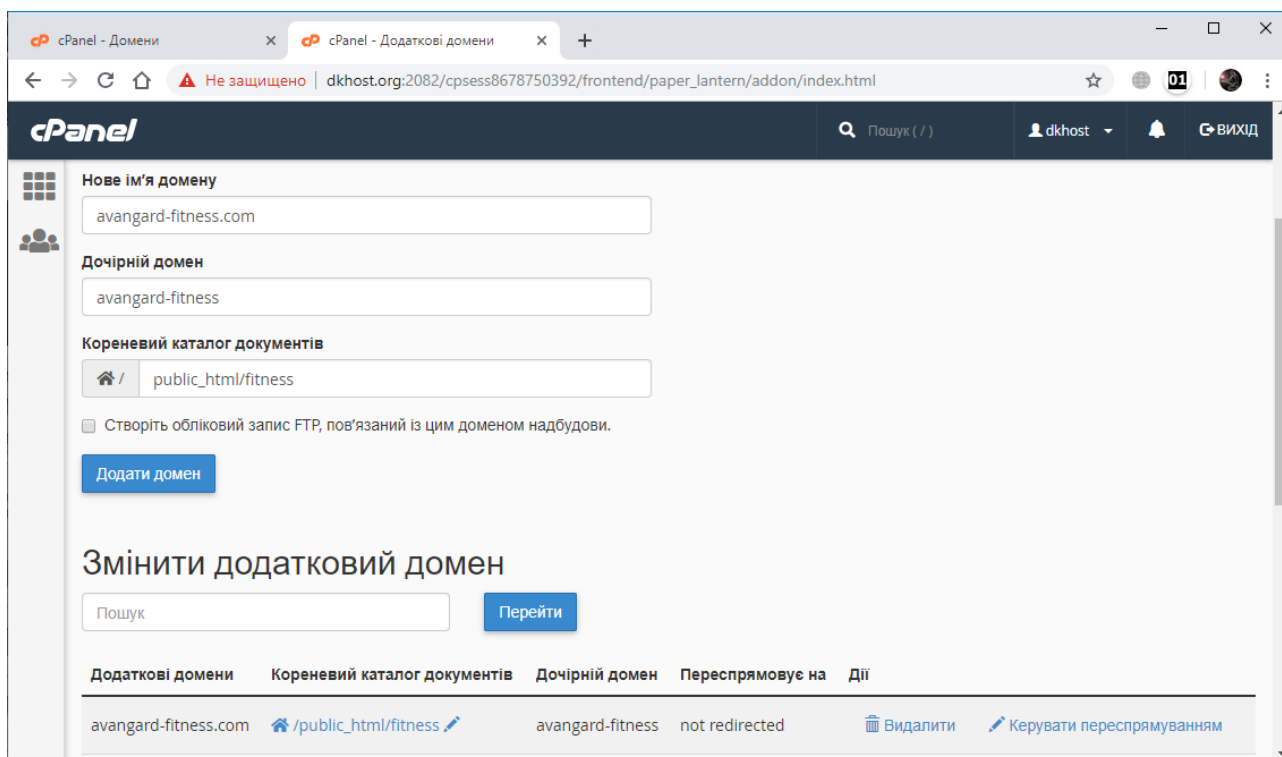


Рис.48. Підключення додаткового домену

За допомогою пункту групи Домени – Дочірні домени можна створювати необмежену кількість доменів нижнього рівня.

Також можна створювати необмежену кількість поштових адрес (e-mail) на базі домену. Для цього треба скористатися пунктом Домени. У списку доменів знайти необхідний і натиснути кнопку створення поштової адреси навпроти нього. Відкриється сторінка створення поштового облікового запису. У ній необхідно задати параметри, а саме адреса до собачки, наприклад vasia, перевірити чи правильно обраний домен для адреси, встановити пароль і підтвердити його, система автоматично перевіряє пароль на складність і може генерувати складні паролі в разі потреби. Останнім пунктом задається квота дискового простору для зберігання пошти (рис.49). При переповненні даного простору нова пошта на заповнений ящик перестає приходити.

Простір під пошту береться з основного хостингу аккаунта, тому важливо правильно підібрати тарифний план і задати відповідні квоти для зберігання сайту і поштових скриньок.

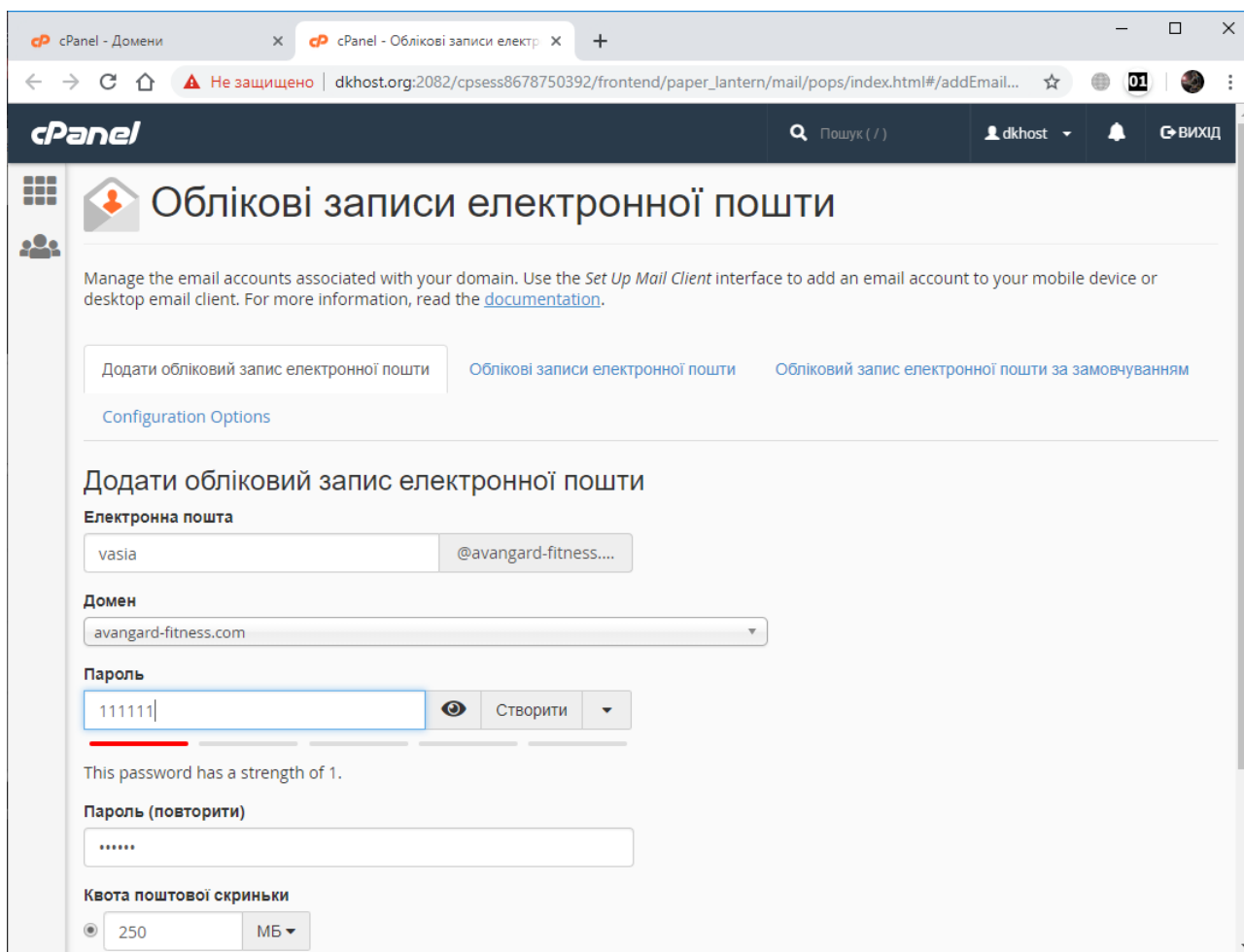


Рис.49. Створення поштової скриньки

Розгортання системи управління сайтом OpenCart

Створимо піддомен shop для майбутнього інтернет-магазину. Після цього рекомендується встановити безкоштовний FTP (FileZilla) клієнт для завантаження файлів і редагування їх на сервері. Після завантаження потрібно налаштувати FTP клієнт створивши в меню файл підключення до сервера, для чого необхідно ввести облікові дані, а саме: сервер, порт підключення: 21, ім'я користувача і пароль, протокол безпеки при підключенні (рис. 50).

Після підключення до сайту за допомогою FTP клієнта зайдіть на сайт електронного магазину і закачайте останню версію. Розпакуйте на комп'ютері теку із магазином і через FTP клієнт завантажте на хостинг в теку /public_html/shop/ (рис.51). У разі необхідності прочитайте інструкції по установці магазину і задайте на деякі теки потрібні права доступу, що робиться контекстним меню на відповідних теках.

Для розгортання систем управління контентом також знадобиться база даних і користувач, який матиме права на управління базою. Для створення бази скористаємося пунктом Бази даних і натиснемо на посилання Бази даних MySQL® (рис.52). Далі створюємо базу з потрібним назвою (рис.53), наприклад shop. До імені бази автоматично дододеться з нижнім підкресленням ім'я хостинг аккаунта, наприклад: імя_shop.

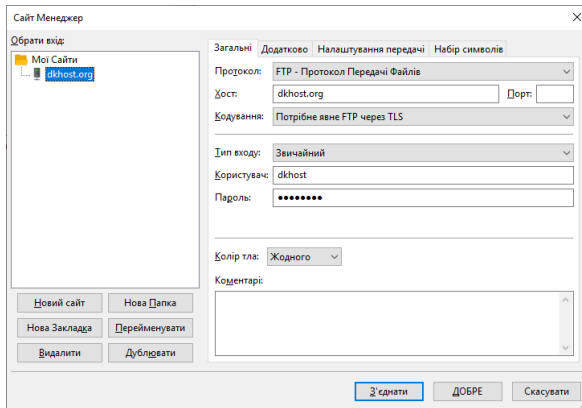


Рис.50. Створення підключення

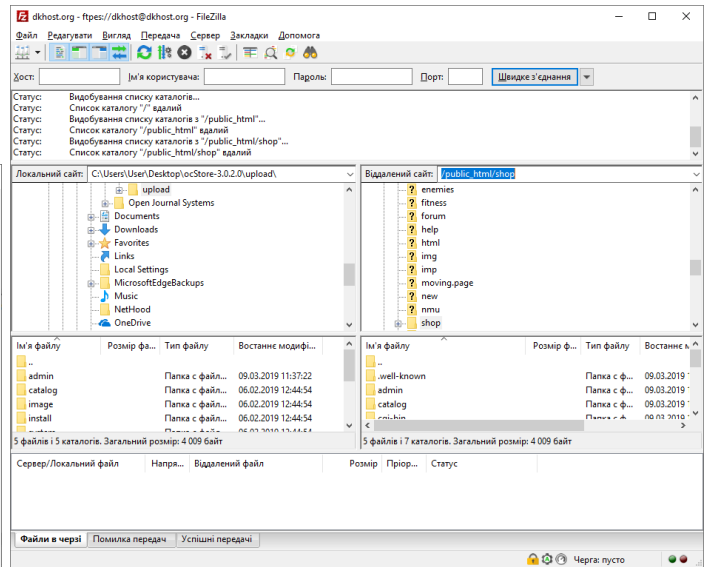


Рис.51. Завантаження сайту в папку

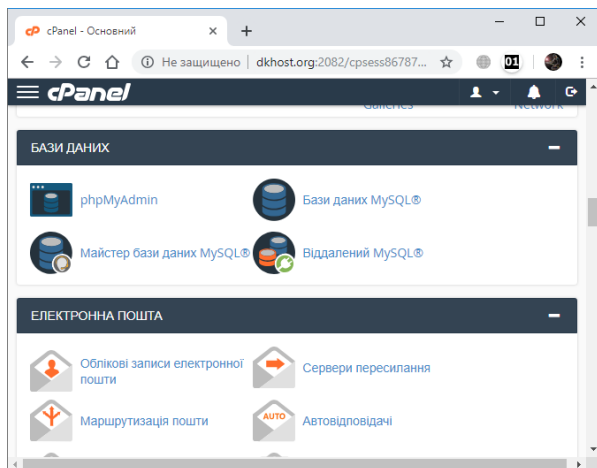


Рис.52. Вкладка баз даних

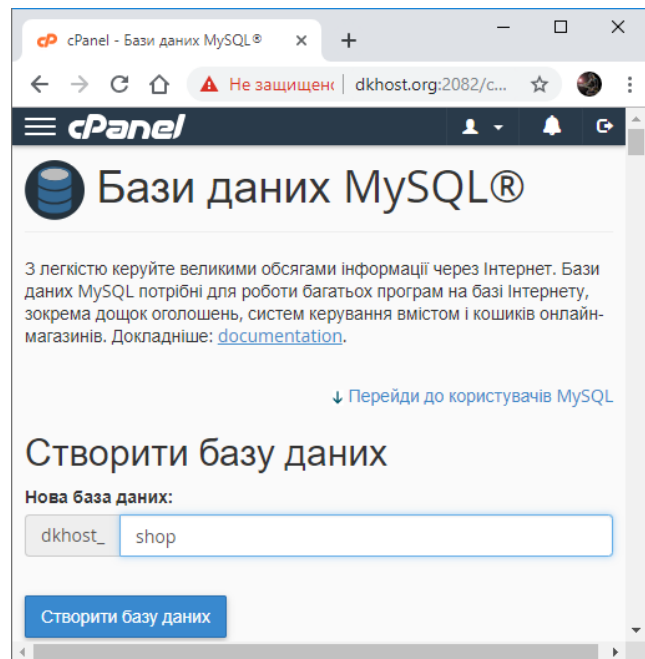


Рис.53. Створення бази даних

На тій же сторінці Бази даних MySQL® прокрутивши нижче знаходимо пункт створення користувача. До імені користувачеві знову додоється ім'я хостинг аккаунта. Вводимо ім'я (userx) та пароль і створюємо користувача (рис.54).

Тепер необхідно додати даного користувача в базу даних і дати йому права на всі операції. Це робиться все на тій же сторінці трохи нижче (рис.55). Зі списків вибираємо ім'я і базу, потім тиснемо кнопку додати.

На наступній сторінці необхідно задати права на використання бази даних для користувача, що додається (рис.56). Встановлюємо прапорець навпроти всіх прав і натискаємо кнопку Внести зміни.

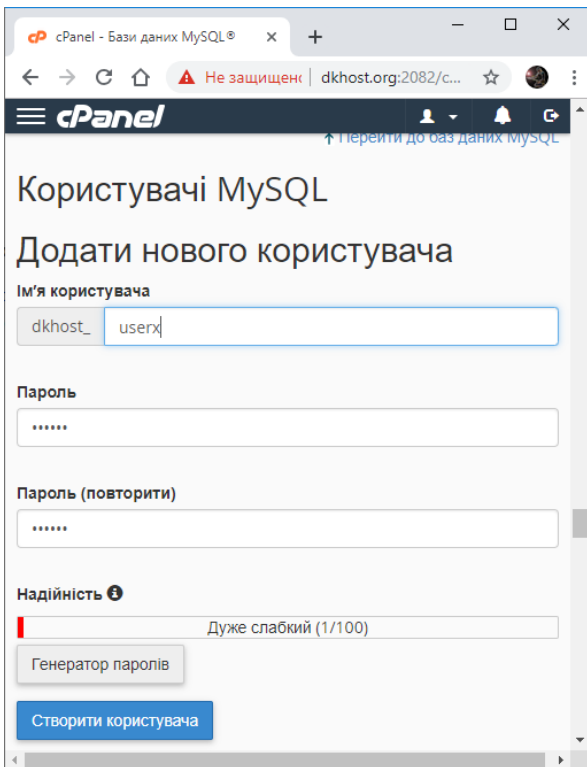


Рис.54. Створення користувача

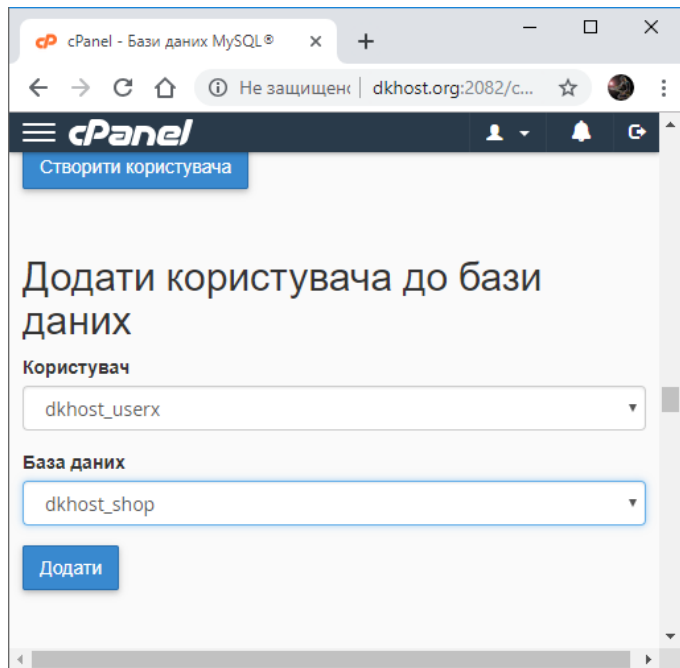


Рис.55. Додавання користувача в базу

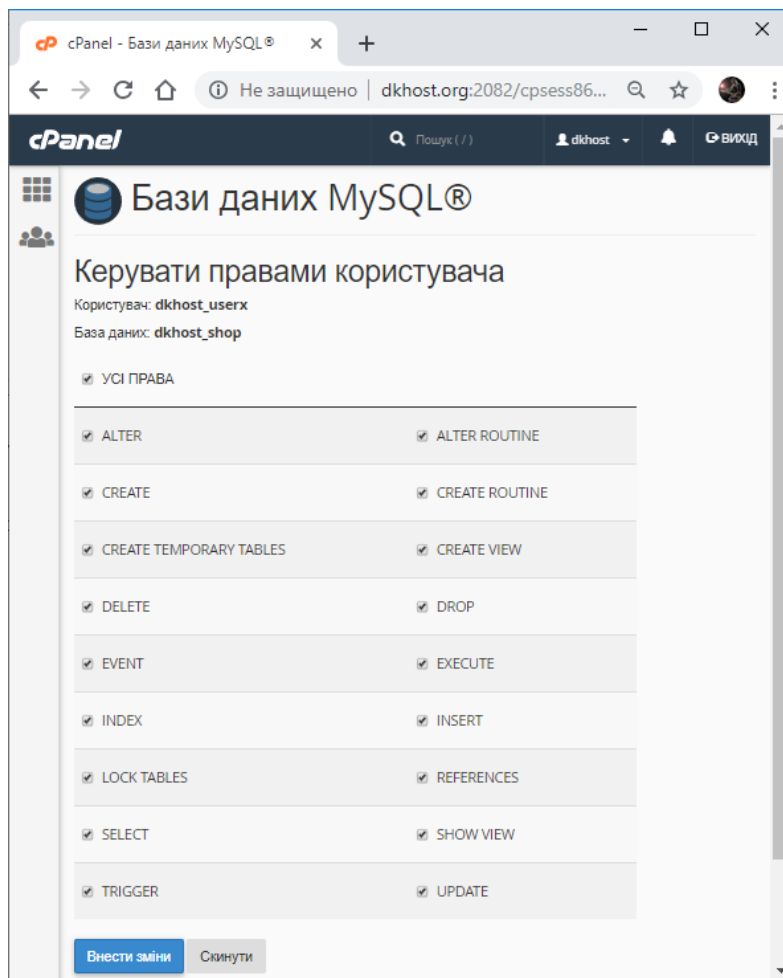


Рис.56. Наділення користувача правами роботи з базою

Переходимо на сторінку shop.домен і слідуючи інструкціям по установці, аналогічним розглянутим в попередньої лабораторної встановлюємо систему управління електронним комерційним сайтом.

Приклад сайту наповненого тестовим вмістом на рис.57.

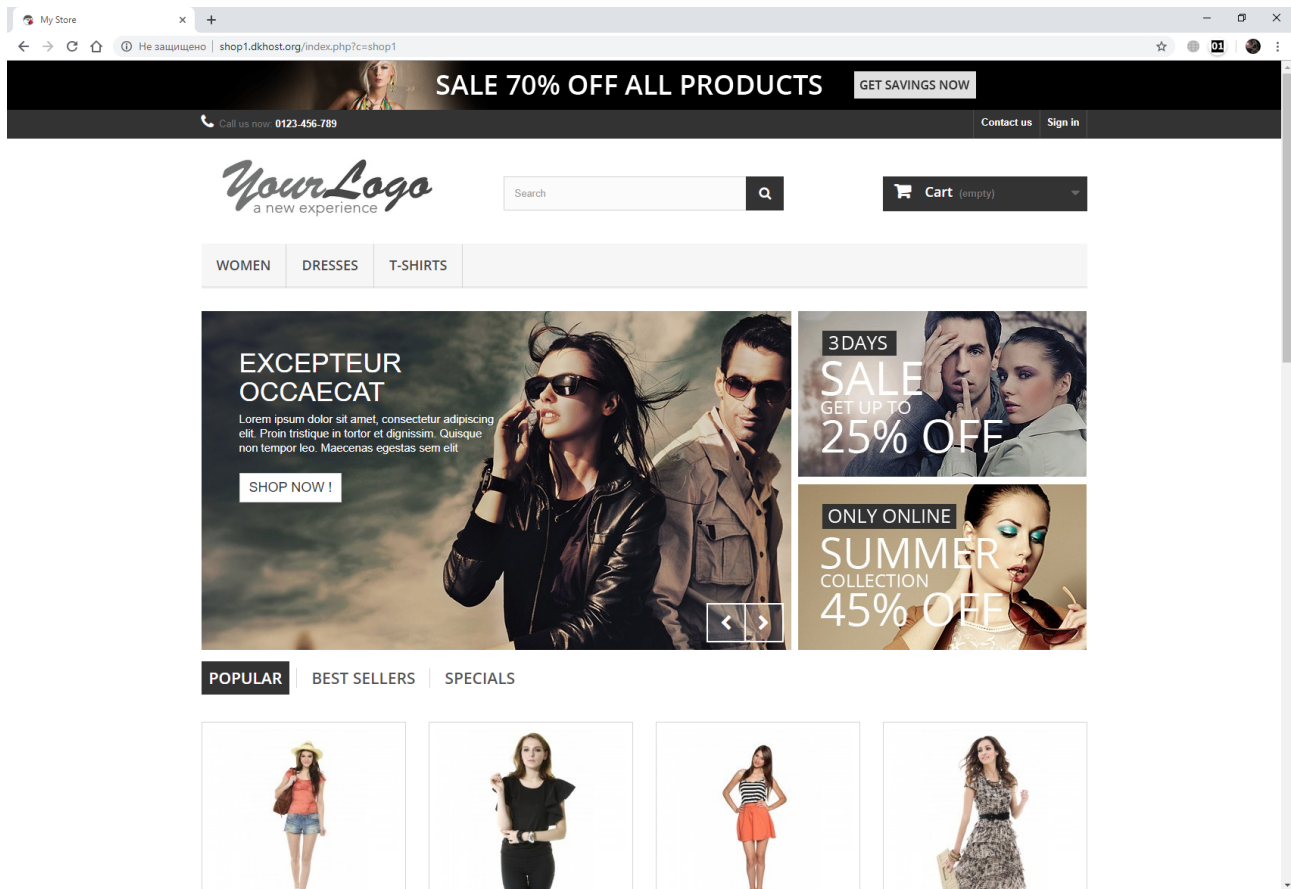


Рис.57. Приклад встановленого електронного магазину

Завдання для виконання: розгорнути на зовнішньому сервері систему управління сайтом на базі Wordpress або OsCommerce (Presta Shop, OpenCart). Відповідно на WP створити сайт-візитку студентської діяльності із 5-ти сторінок (головна, про себе, портфолію, хобі, контакти).

На CMS комерційної спрямованості створити свій розділ товарів і розмістити 5 товарів з фото та описом.

ЛІТЕРАТУРА

1. Письменкова Т.О. Інформаційні системи і технології у інженерії: Навч. посібник / Т.О. Письменкова, А.О. Логінова, С.О. Федоряченко, О.В. Федоскіна, І.В. Вернер; Дніпро: Національний технічний університет «Дніпровська політехніка», 2019. – 227 с.
2. Методичні вказівки з використання растрової графіки при виконанні лабораторних робіт з дисципліни «Методи та засоби дизайнерських рішень» для студентів всіх спеціальностей / С.О. Федоряченко, І.В. Вернер, Т.О. Письменкова – Д.: НГУ, 2016. – 52 с.
3. Flexbox Froggy – гра для вивчення CSS Flexbox / [Electronic resource]. URL: <http://flexboxfroggy.com/> (Date of access: 1.04.2020).
4. Sams Teach Yourself PHP, MySQL & JavaScript All in One, Sixth Edition. – Copyright © 2018 by Pearson Education, Inc.
5. Developing Business Applications for the Web: With HTML, CSS, JSP, PHP, ASP.NET, and JavaScript // Laura Ubelhor and Christian Hur / MC Press Online, LLC. - 2017. - 930 p.
6. Навчальні відео до дисципліні «Продакт дизайн» / Сайт кафедри КТЕД [Electronic resource]. URL: <http://okmm.nmu.org.ua> (Date of access: 1.03.2020).
7. Історія розвитку Web [Електронний ресурс] – Режим доступу до вебсередовища: <https://uk.wikipedia.org/wiki/Інтернет>
8. Опис Web технологій [Електронний ресурс] – Режим доступу до вебсередовища: <http://htmlweb.ru/>
9. Класифікація веб-сайтів [Електронний ресурс] – Режим доступу до вебсередовища: <http://scbali.com/ua/web-studiya/typy-saytiv.html>
10. Завантаження та встановлення Denwer [Електронний ресурс] – Режим доступу до веб-середовища: <http://www.denwer.ru/base.html>