Міністерство освіти і науки України
Житомирський державний університет імені Івана Франка

Мосіюк О.О.

WEB-ТЕХНОЛОГІЇ

Частина 1. Верстка

Навчально-методичний посібник

Житомир
Вид-во ЖДУ ім. І. Франка
2020

Зміст

Передмова

На сьогодні web-технології займають важливе місце у сучасному інформаційному суспільстві. За їх допомогою переважна більшість людей дізнається про важливі події у світі та країні, звичною стала купівля речей у онлайн магазинах, широкого розповсюдження набув Internet-речей, активно використовуються системи онлайн навчання (Moodle) та освітні платформи такі як Coursera та Edx.com тощо. Тож не дивно, що у рамках шкільного курсу інформатики вивчається багато тем, пов'язаних із Internet, а у одинадцятому класі, відповідно до програми із інформатики 2017 року, присутній окремий вибірковий модуль «Web-технології», при вивченні якого учні знайомляться із процесом проектування, верстки та програмування серверної частини сучасних сайтів. Саме тому в освітній програмі «Середня освіта. Інформатика» одним із обов'язкових предметів для вивчення є «Web-технології та web-дизайн».

Отже метою представленого навчально-методичного посібника є ознайомлення майбутніх учителів інформатики із основними підходами, принципами, методами та засобами верстки сайтів за допомогою мови гіпертекстової розмітки HTML5 та каскадних таблиць стилів CSS3.

Загалом навчально-методичний посібник складається із двох частин та додатків.

У першій частині описується стандарт мови HTML5, рекомендації щодо використання тегів та правильної побудови семантики web-сторінки.

Друга частина розкриває особливості застосування каскадних таблиць стилів CSS, технологій Flex та CSS Grid для оформлення сайту, керуванням «потоком» сторінки сайту тощо.

Додатки містять посилання на важливу літературу та інтернет джерела, а також завдання до лабораторних робіт.

Навчально-методичний посібник, що пропонується, може бути рекомендованим студентам ЗВО, а саме студентам фізико-математичних факультетів спеціальності 014.09 Середня освіта (Інформатика), вчителям інформатики, слухачам курсів підвищення професійної кваліфікації в процесі професійного вдосконалення.

ЧАСТИНА 1.

МОВА ГІПЕРТЕКСТОВОЇ РОЗМІТКИ

HyperText Markup Language

HTML (скорочення англійських слів HyperText Markup Language) є стандартизованою мовою розмітки документів, які відображаються у Web-браузері. Вона дозволяє публікувати у мережі Internet інформацію із заголовками, текстами різного обсягу, таблицями, списками, фото та відеоматеріалів, розміщувати матеріали із гіперпосиланнями і інтерактивними формами для передачі та опрацювання даних на серверах тощо. Стандартизацією мови розмітки займається World Wide Web Consortium (W3C)[1]. Останнім стандартом, який рекомендовано послуговуватися при виконанні верстки сайтів, є версія HTML 5.2 від 14 грудня 2017 р.[2] Він є основним для формування семантичної структури документів у електронній мережі.

Представлення інформації на web-сторінці відбувається за допомогою спеціалізованих команд – тегів, які записуються у файлі із розширенням .html або .htm. Всі теги поділяються на дві великі групи.

Парні теги – інструкції, що мають команди, які позначають початок та кінець (приклад №1).

Приклад № 1.

```
<p> ТЕКСТ</p>
<body> … </body>
<head> … </head>
```

Одинарні теги задаються однією інструкцією (приклад №2).

Приклад № 2.

```
<br>
<hr>
<link>
```

Вони використовуються для того, щоб задати семантичну структуру документу в браузері та стандартизувати передачу даних у мережі за допомогою протоколу http.

[1] W3C [Електронний ресурс] – Режим доступу до ресурсу: https://www.w3.org/Consortium/.
[2] HTML 5.2. W3C Recommendation [Електронний ресурс] – Режим доступу до ресурсу: https://www.w3.org/TR/2017/REC-html52-20171214/.

Розглянемо більш детально ієрархію команд у файлі із розширенням html. Нижче (у прикладі № 3) представлено обов'язкові теги для кожної web сторінки.

Приклад № 3.

```
<!DOCTYPE html>
<html>
<head>
  <title>Sample#3</title>
</head>
<body>
  <!-- Тут розміщується основний вміст документу-->
</body>
</html>
```

У першому рядку міститься обов’язкова декларація `<!DOCTYPE html>`, яка вказує браузеру на який стандарт орієнтуватися при відображенні інформації. На даний момент ця інструкція вказує, що використовується специфікація HTML 5.

Тег `<html></html>` визначає основну частину сторінки та містить два вкладених контейнери `<head></head>` та `<body></body>`. У першому контейнері задається інформація призначена для пошуковиків, браузерів, завантаження додаткових елементів оформлення (зокрема каскадних таблиць стилів) тощо. Між відкритим та закритим тегами `<body></body>` подається всі матеріали, які відображатимуться у браузері.

Специфікація HTML

Особливості використання тегів HTML при верстці сторінки докладно описується специфікацією[3]. Вона є повним описом мови розмітки та особливостей її використання. Загалом текст документу поділений на такі важливі розділи як: вступ (Introduction); загальної

[3] HTML 5.2. W3C Recommendation [Електронний ресурс] – Режим доступу до ресурсу: https://www.w3.org/TR/2017/REC-html52-20171214/.

інфраструктури (Common infrastructure); семантики, структури та API HTML документів (Semantics, Structure and API of HTML documents); елементи HTML (The elements of HTML); взаємодія із користувачем (User interaction); завантаження Web-сторінок (Loading Web-pages); API web-додатків (Web application APIs); синтаксис HTML (The HTML syntax); синтаксис XML (The XML syntax); рендер сторінки (Rendering) тощо. Кожен із них описує важливі елементи та етапи формування сторінки, взаємодію із користувачем і браузером, передачу даних.

Зазначені розділи є тим переліком правил, які повністю регламентують формування структури web-сторінки та основу для її візуалізації. Саме тому, у питаннях правильного розміщення та вкладення тегів (і не тільки) варто звертатися до специфікації.

Основні категорії тегів

Відповідно до розглядуваної специфікації (а точніше у відповідності до четвертого розділу специфікації – the elements of HTML) всі теги можна поділити на такі великі категорії.

Metadata content[4] – метадані, які необхідні для браузерів, систем пошуку та SEO оптимізації.

Flow content – теги потокового контенту, які дозволяють задати ключові елементи web-сторінки.

Sectioning content – схожі за структурою до попередньої категорії, але обов'язковую умовою є наявність заголовку (наприклад тег `<article>`).

Heading content – теги заголовків.

Phrasing content – інструкції цієї категорії дозволяють представляти текстовий матеріал на web-сторінці (наприклад тег `<article>`).

Embedded content – сторонні вбудовані елементи, найчастіше це зображення.

Interective content – всі елементи сторінки, які безпосередньо будуть взаємодіяти із користувачем.

Варто зауважити, що теги можуть до декількох категорій. Наприклад тег `<scripts>`, який дозволяє розмістити код JavaScript

[4] Назви категорій подаються англійською мовою так як вони представлені у оригіналі специфікації.

або підключити сторонній js файл у html документ дозволяється розміщувати як метадані у контейнері <head></head> так і разом із потоковими тегами у тілі самого документа <body></body>. Схематично всі категорії можна зобразити наступним чином (рис. 1.)

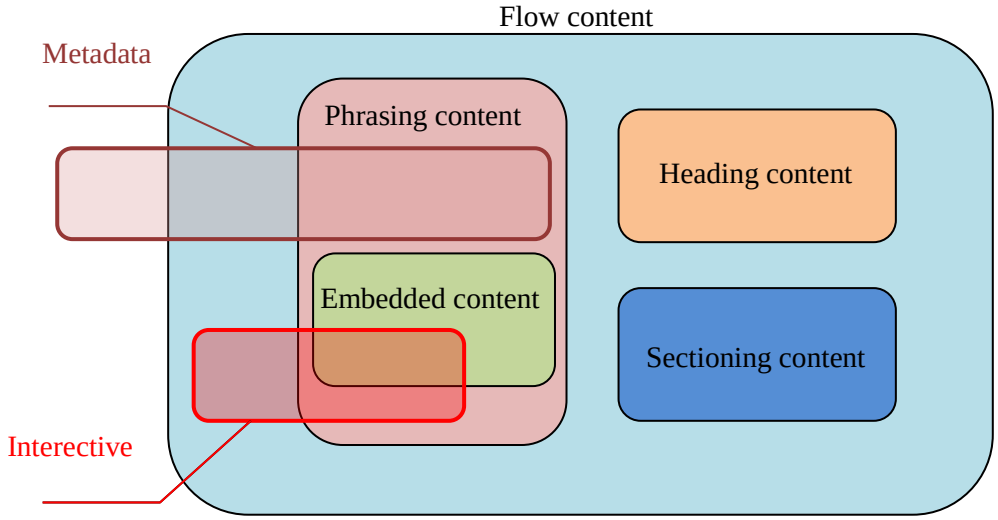


Рис. 1. Співвідносність категорій тегів HTML5

Зображена схема дозволяє зрозуміти яким чином співвідносяться теги HTML у відповідності до зазначеної класифікації.

Ще одним важливим моментом є визначення можливостей вкладеності одних тегів у інші. Для того, щоб перевірити чи можна розміщувати умовний тег <X> у середині тега <Y> варто дотримуватися наступних рекомендацій.

1. Використовуючи документацію стандарта HTML 5.2, необхідно перевірити контентну модель тега <Y>, у якому буде розміщуватися тег <X>.

2. Перевірити категорію тега <X>, який буде вкладатися у тег <Y>.

3. Якщо категорія тега <X> співпадає із умовами контентної моделі тега <Y> та не накладаються додаткові вимоги, то у такому випадку розміщення дозволяється.

У всіх інших випадках, при перевірці сторінки на відповідність вимог стандартизації (процес валідації сторінки), буде видавати помилку, а отже такі сайти понижуватимуться у пошукових системах.

Окремо варто розглянути таку контентну модель як Transparent. Теги <T>, у описі яких вказано таке поняття, не підпадають під загальні правила визначення вкладеності. У такому випадку запропонований алгоритм видозмінюється. Основна увага, за такої

ситуації, звертається на тег, який містить розглядуваний тег <T>. Якщо у специфікації зовнішнього тега дозволяється вкладення, то і для тегів із контнтною моделю Transparent також буде дозволятися вкладення, якщо ж модель зовнішнього тега вкладення не дозволяє, то забороняється вкладення і для тега <T>.

Перевірити на правильність виконання верстки можна за допомогою спеціалізованого ресурсу – валідатора W3C[5]. Серед повідомлень, які видає сервіс після перевірки, варто виділити такі: Error та Warning. Перше вказує на серйозну помилку верстки, а друга – на потенційну проблемну ситуацію, що може погіршити ефективність аналізу цієї сторінки пошуковими системами. Сервіс відображає повідомлення про помилку наступним чином. Спочатку йде запис повідомлення Error, далі йде коротке повідомлення про сутність помилки і нарешті вказується номер рядка, де було зроблено помилку. Починати виправлення помилок варто із самих перших, оскільки, зазвичай, саме вони спричинюють подальше нарощування критичних зауважень і їх корекція може видалити більшість повідомлень Error та Warning.

Основні теги html, які найчастіше використовуються при верстці

Стандарт HTML налічує велику кількість тегів, проте не всі вони використовуються при верстці. Розглянемо теги, що найчастіше застосовуються при створенні сучасних web-сторінок. Для початку опишемо команди, які задають ключові структурні елементи Internet-документу.

<section> … </section> – задає важливу структурну одинницю web-сторінки. Може містити у собі інші, об'єднані загальною тематикою частини. Бажане розміщення заголовка, який може і не відображатися.

<article> … </article> – являє собою самостійну частину Internet-документа, призначену для незалежного поширення або повторного використання. Цей елемент може представляти статтю на форумі, журналі або газеті, запис в блозі або будь-який інший самостійний фрагмент. Обов'язковим є наявність заголовка.

[5] Markup Validation Service [Електронний ресурс] – Режим доступу до ресурсу: https://validator.w3.org/.

<nav> … </nav> – розділ сторінки сайту, який містить посилання на інші сторінки або ж на певні занчимі частини сторінки. Найчастіше рекомендується використовувати для представлення основної навігації на сайті.

Наступними за частотою використання є теги заголовків <h1>, <h2>, …, <h6>.

Серед ключових рекомендацій, які стосуються використання тегів заголовків при верстці сайтів варто назвати такі.

1. На сторінці сайту дозволяється розміщення тільки один заголовок першого рівня.

2. Не варто змінювати ієрархію заголовків на web-сторінці.

3. При верстці сайту слід враховувати, що не завжди великий розмір шрифту тексту відповідає заголовку першого рівня і нести суто функцію оформлення та привернення уваги користувачів та не нести жодного смислового значення.

Не дотримання щойно перелічених рекомендацій, може привести до того, що пошукові системи не розумітимуть сутність документу та понижатимуть рейтинг сторінки у свої вибірках.

Тепер розкриємо сутність тегів, які описують структурні одниці сторінки.

<header> …</header> – дозволяє позначити вступний контент на сторінці, зазвичай групу визначальних для сайта елементів або навігаційних засобів. Він може бути складним і містити інші теги-заголовки, а також логотип, форму пошуку, ім'я автора та інші елементи.

<main> … </main> – позначає унікальну частину контенту Internet-сторінки. Його слід використовувати на сторінці лише одноразово.

<footer> … </footer> – є ще одним важливим потоковим тегом, який використовується у кінці сторінки сайту та згруповує всю інформацію про аторів, захист авторських прав, форми зворотнього зв’язку тощо. Так як і для тегу <header> …</header> його можна використовувати декілька разів при формуванні структури сторінки.

Окрім важливих інструкцій мови HTML, які задають ключові структурні елементи сторінки, варто виокремити теги, що дозволяють формувати основний контент.

 … та <ol> … – теги, які дозволяють створювати списки на web-сторінці, а також їх часто використовують при створенні типових повторюваних частин документу, наприклад карток товарів у інтернет магазинах.

 … – застосовують при оформленні великих блоків тексту (абзаців) або групуванні фразових елементів.

<table> … </table> – застосовується при описі тільки табличних даних, але суворо забороняється використовувати при формуванні сітки розміщення елементів. У рамках контейнеру <table> … </table> також викоритовуються теги <tr> … </tr>, <td> … </td>, які дозволяють задавти рядки та стовпчики таблиці відповідно.

 … – універсальний за своєю суттю та використанням тег. Він може використовуватися як обгортка для групування інших структурованих тегів так і на рівні із ними.

Серед найбільш використовуваних тегів також варто назвати і такі інструкції, які описують представлення фразового контенту.

<img> – одинарний тег для завантаження інформаційно важливих зображень. Для нього є обов'язковим атрибута src. Також вважається правилом гарного стилю написання коду використання атрибуту alt, який описує саме зображення. Завантажувати фонові графічні файли, яа допомогою цієї інструкції HTML не рекомендується.

<a> … – створює посилання на іншу частину документу, сторінку сайту або інший Internet ресурс. Атрибут href задає адресу переходу, якщо його при верстці не вказується, то це означає, що посилання буде додаватися скриптом за певних умов.

<button> … </button> – тег, який задає кнопку для форми, що передаватиме дані або виконувати скрипт.

 – перенесення тексту на новий рядок, але це не означає що створюється абзац.

 … – універсальний фразовий елемент, який найчастіше використовується для налаштування зовнішнього вигляду певних фразових елементів у тексті.

Атрибути тегів

Щоб розширити можливості тегів і більш гнучко керувати вмістом контейнерів застосовуються атрибути тегів, які розштрюють їх функціональність або ж навпаки – інструкції HTML повністю втарчають свою сутність та функціональність. Розглянемо найпоширеніші із них.

Атрибут `class` – задає зовнішнє оформлення тега. Якщо необхідно застосувати декілька селекторів типу «клас», то вони перераховуються у лапках через пропуск (пробіл).

Атрибут `id` – діє схожим чином як і попередній, але має значну відмінність. На одній сторінці має зустрічатися тільки один раз селектор id із унікальним ім'ям.

Окрім цих двох найпоширеніших атрибутів існують і спеціалізовані. Вони можуть відноситися до певного типу (текст, число, шлях до файлу тощо), який обов'язково повинен враховуватися при записі значення атрибута. Наприклад тег `<img>`, що додає на web-сторінку зображення, а його атрибут `width` задає ширину зображення в пікселях, `height` – його висоту, `src` – шлях до файла малюнка, а атрибут `alt` додає опис до зображення. Для тега `<a>` важливим є наявність атрибута `href`, що вказує посилання на документ у мережі Internet.

Значення атрибута задається у лапках.

Організація структури документу

Структурування web-сторінки – це процес макетування документу із використанням тегів, які відповідають змісту представленого контенту. Його ще можна означити ще розміткою документа.

Розрізняють два підходи до структурування.

1. Web-сторінка як документ. Для такої організації сайту характерним є правильне представлення графічного та текстового контенту. Як приклад таких можна навести сайти новин і їх сторінки, які описують докладно вибрану користувачем новину.

2. Web-сторінка як інтерфейс користувача. Ключовою особливістю у цьому випадку є зручне і функціональне розміщення елементів керування. Гарною ілюстрацією такого підходу до структурування є інтерфейс поштових скриньок або є сховища Google Drive.

Варто зауважити, що сайтів, які мають сторінки структуровані відповідно до першого або другого підходу є небагато. Переважна більшість Internet ресурсів суміщають ці підходи при верстці у різних пропорціях. Тож наведемо загальні рекомендації до створення розмітки сучасних web-сторінок.

1. Виокремлювати повторювані елементи сторінок сайту (зазвичай це заголовок та підвал сайту, а отже використовувати такі потокові теги як `<header>` і `<footer>`).

2. Розмічувати важливі частини за допомогою тегів `<section>`, `<article>` та `<nav>`.

3. Виділити унікальну частину документу за допомогою тега <main> або задати змістовою назвою класу.

4. У кожному важливому розділі задавати заголовок.

5. Виконувати структурування менш значимих компонентів сторінки, спираючись на такі інструкції:

А) Якщо є можливість підібрати необхідний за змістом тег, то використовують сааме його, а не будь-який інший.

Б) Якщо відповідного тега не існує, то у такому випадку використовують потоковий тег `<div>` із відповідною назвою класу.

В) Якщо необхідно виконати групування тексту або ж іншого фразового контнету, то застосовують тег `<p>`.

Г) При виділенні окремого фразового контенту використовують тег `<span>`.

При цьому варто розуміти, що подальша робота із верстки сторінки може призвести до удосконалення, а також і ускладнення, структури web-сторінки.

Теги для верстки форм

Важливим частиною процесу верстки сучасних web-сторінок є створення форм для заповнення та пердачі даних від користувача на сервер для подальшого опрацювання і реалізації зворотнього зв'язку. Для їх представлення використовують наступний тег `<form> … </form>`.

Основними атрибутами тега є такі: `accept-charset`, `action`, `autocomplete`, `enctype`, `method`, `name`, `novalidate`, `target`. Розглянемо їх значення докладніше:

`accept-charset` – задає кодування, в якій сервер зможе приймати та обробляти дані;

`action` – адреса скрипта, який обробляє дані форми;

`autocomplete` – умикає автозаповнення полів форми (значеннями, які набуває атрибут є on та off);

`enctype` – вказує способи кодування даних форми (значеннями, які набуває атрибут є application/x-www-form-urlencoded, multipart/form-data та text/plain);

`method` – метод передачі даних протоколу HTTP (значеннями, які набуває атрибут є get та post);

`name` – задає ім'я форми;

`novalidate` – скасовує вбудовану перевірку даних форми на коректність введення;

`target` – ім'я вікна або фрейму, куди скрипт буде завантажувати необхідний опрацьований результат.

Серед них найчастіше використовуються атрибути `action`, `method` та `name`.

Розглянемо більш детально значення атрибута `method`.

Метод передачі `get` є одним з тих, що найчастіше використовуються і призначений для отримання необхідної інформації та передачі даних у адресному рядку. Представлення даних відбувається у вигляді пар – «ім'я = значення», які приєднуються до адреси після знаку питання і розділяються між собою символом &. Передача даних обмежується обсягом 4 Кбайт.

Якщо вказується значення post, то тоді браузер надсилає на сервер дані у самому запиті. Таких підхід дозволяє відправляти більшу кількість даних, ніж за допомогою метода get. Великі обсяги даних зазвичай використовуються при створенні форумів, поштових служб, заповненні бази даних, при пересиланні файлів тощо.

Формування основних полів, кнопок, чекбоксів та інших елементів, а також їх підписів, відбувається за допомогою тегів: теги, які створюють елементи форми чи спеціальні позначення (<button> … </button>, <input>, <label> … </label>, <legend> … </legend>, <option> … </option>, <output> … </output>, <select> … </select>, <textarea> … </textarea>), групують їх (<fieldset> … </fieldset> та <optgroup> … </optgroup>) та виконують спеціальне шифрування (<keygen> … </keygen>). Розкриємо значення кожного із них.

<button> … </button> – створює на web-сторінці кнопки. В середині цього тега дозволяється розміщувати будь-які елементи HTML, у тому числі зображення. Використовуючи CSS стилі, дозволяється змінити зовнішній вигляд кнопки шляхом зміни шрифту, кольору фону, розмірів або інших параметрів.

Серед важливих атрибутів тега треба виділити такі.

accesskey – доступ до елементів форми за допомогою гарячих клавіш;

autofocus – вказує, що кнопка отримує фокус після завантаження сторінки;

disabled – блокує доступ і зміни елемента;

form – зв’язує кнопку із формою за її ідентифікатором (задається атрибутом name у формі). Такий зв'язок необхідна тоді, коли кнопки не розташовується всередині тега <form>, наприклад, при створенні її програмно.

formaction, formenctype, formmethod, formnovalidate, formtarget – діють аналогічно до атрибутів форми action, enctype, method, novalidate, target. Якщо одночасно вказати для атрибутів форми і їм відповідним у кнопці, то при натисканні на кнопку аналогічні атрибути форми

ігнорується, а пріоритетними залишаються ті, які були задані до тега <button> … </button>.

name – задає унікальне ім'я для кнопки;

type – визначає тип кнопки (button –звичайна кнопка, reset – кнопка для очищення введених даних форми та повернення значень в первісний стан, submit – кнопка для відправки даних форми на сервер). За замовчуванням визначається значення submit.

value – значення кнопки, яке буде передано на сервер або прочитано за допомогою скриптів.

Тег <input> призначений для створення текстових полів, різних кнопок, перемикачів і чекбоксів. Основний атрибут тега <input>, що визначає зовнішній вигляд компонента форми є type. Він дозволяє задавати такі елементи форми як: текстове поле (text), поле з паролем (password), перемикач (radio), прапорець (checkbox), приховане поле (hidden), кнопка (button), кнопка для відправки форми (submit), кнопка для очищення форми (reset), поле для відправки файлу (file) і кнопка із зображенням (image).

Серед важливих атрибутів для цього тега варто назвати такі.

accept – встановлює фільтр на типи файлів, які ви можете відправити через поле завантаження файлів.

accesskey – перехід до елементу за допомогою комбінації клавіш.

autofocus – встановлює фокус в поле форми.

max – верхнє значення для введення числа або дати.

min – нижнє значення для введення числа або дати.

maxlength – максимальна кількість символів дозволених в тексті.

multiple – дозволяє завантажити кілька файлів одночасно.

name – ім'я поля, призначене для того, щоб скрипт, який опрацьовує форму, міг його ідентифікувати.

pattern – встановлює шаблон введення даних у полі.

placeholder – виводить фонову підказку у полі.

readonly – встановлює, що поле не може змінюватися користувачем.

`required` – указує браузеру, що поле обов'язкове для заповнення.

`size` – задає ширину текстового поля.

`step` – крок збільшення або зменшення значень для числових полів.

`tabindex` – визначає порядок переходу між елементами форми за допомогою клавіші Tab.

`value` – значення компонента форми.

`<label> … </label>` – задає підпис елемента форми та дозволяє зв’язати його із текстом підпису. Основними атрибутами для цієї інструкції HTML є: `accesskey` – задає комбінацію клавіш для активації елемента форми із клавіатури та `for` – дозволяє зв’язати підпис із компонентом форми, задавши значення його `id`.

`<legend> … </legend>` –застосовується для створення заголовка групи елементів форми, яка визначається за допомогою тегу `<fieldset> … </fieldset>`.

`<option> … </option>` – визначає окремі пункти списку, що створюється за допомогою контейнера `<select>`. Ширина списку задається найбільш довгим текстовим рядком, зазначеним у тегу `<option>`, а також може змінюватися за допомогою каскадних таблиць стилів.

`<output> … </output>` – визначає область сторінки, у яку виводиться інформація після опрацювання даних із форми, переважно керується за допомогою скриптів.

`<select> … </select>` – дозволяє створити список, який розкривається, та список з одним або множинним вибором.

`<textarea> … </textarea>` – створює область форми, в яку можна вводити кілька рядків тексту. На відміну від тега `<input>` в текстовому полі дозволяється робити переноси рядків і вони зберігатимуться при відправці даних на сервер.

Окремо розглянемо теги, які дозволяють групувати теги елементів форми. Першим і одним із найпоширеніших у застосуванні є тег `<fieldset> … </fieldset>` . Це полегшує роботу з формами, які містять велику кількість різного типу полів. Наприклад,

один блок може бути призначений для введення текстової інформації, а інший – для чекбоксів або радіокнопок.

<optgroup> … </optgroup> – представляє собою контейнер, усередині якого розташовуються теги <option> об'єднані в одну спільну групу. Особливістю тега є те, що він не є звичайним елементом списку, а виділяється за допомогою жирного накреслення, при цьому всі елементи, що входять до цей контейнер, зміщуються вправо від свого початкового положення.

Тепер наведемо загальні рекомендації, що до коректного оформлення форм для сучасних web-сторінок.

1. Обов’язкова відповідність між вмістом, яке буде вводиться, та типом поля, що буде використано для отримання інформації від користувача.

2. Для тега <form> обов’язковим є наявність атрибутів action, method та name.

3. Для підвищення доступності компонентів форми необхідно описувати поля за допомогою атрибута placeholder.

4. Для підпису поля варто використовувати тег <label> … </label>, який обов’язково прив’язується до поля за допомогою атрибута for. Це необхідно для того, щоб при кліку кнопкою миші по назві відбувалася активація власне самого поля.

5. Для кожного поля id має бути унікальним.

6. Всі радіокнопки та прапорці мають бути названі унікальним іменем.

7. Групи полів мають об’єднуватися за допомогою тега <fieldset> … </fieldset> та містити її назву, що задається <legend> … </legend>.

ЧАСТИНА 2.
КАСКАДНІ ТАБЛИЦІ СТИЛІВ

Cascade Style Sheets

CSS (скорочення від англійських слів Cascade Style Sheets або каскадні таблиці стилів) властивості задають для web-документа його оформлення і поведінку блоків. Саме форматування сторінки за допомогою каскадних таблиць стилів забирає левову частку процесу верстки.

Властивості CSS умовно можна поділити на такі великі групи властивостей:

- задають позиціонування об’єктів;
- пов’язані із встановленням розмірів і відступів;
- керуванням «потоком» документа;
- форматування тексту;
- декоративне оформлення елементів;
- елементи анімації та динамічних ефектів.

Окремо варто виділити псевдоелементи та псевдокласи.

Синтаксис CSS є достатньо простим і його умовно можна подати так, як показано у прикладі 4.

Приклад 4.

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

Селектором є певне правило, яке визначає зовнішній вигляд та поведінку тега HTML, до якого застосовується. Опис, зрозумілий для браузера, подається між знаками відкритої і закритої фігурних дужок за допомогою запису пар `властивість – значення`. Кожен опис властивості завершується знаком крапкою із комою, а властивість і її значення розділяється двокрапкою.

Для коментування частини коду використовується наступний запис (Приклад 5).

Приклад 5.

```
/* Це коментар у CSS*/
```

Селектори можуть задаватися наступним чином.

Селектор, який задається іменем тега, визначає правило, яке буде застосовуватися для всіх тегів відповідної назви. Запис `p{…}` означає, що всі налаштування оформлення будуть застосовані до всіх тегів `p` заданої сторінки.

Селектор, запис якого починається із точки (клас), застосовується до всіх тегів HTML документу, які мають у описі атрибут class = ″назва_класу″ (приклад 6).

Приклад 6.

```
CSS
.redtext{
    color: red;
}
HTML
<p class = "redtext">Text</p>
```

Селектор id (задається через із використанням символу # на початку) може використовуватися лише один раз у рамках однієї web-сторінки для форматування або позначення певного унікального елемента HTML документа (приклад 7).

Приклад 7.

```
CSS
#uniq_id{
    text-align: center;
}
HTML
<div id = "uniq_id"></div>
```

Окрім стандартних записів правил оформлення зовнішнього вигляду документу в CSS є можливість комбінувати їх застосування. Розберемо їх докладніше на прикладах.

`a.help{…}` – правило буде застосовуватися до всіх тегів `<a></a>`, які у той же час матимуть атрибут `class = "help"`. Тобто буде виконуватися оформлення всіх тегів із назвою класу `help`.

Схожим чином буде працювати поєднання атрибуту та селектора `id – h2#chp_3{…}`. Стилі будуть застосовуватися до всіх тегів `h2`, у яких буде оголошено `id = "chp_3"`.

Важливою можливістю CSS є наявність вкладеного селектору або ж інша його назва контекстний селектор.

`card a{…}` дозволяє застосовувати правила оформлення до посилання a, що розміщення тега, який має клас `card`.

Аналогічно, може працювати вкладеність і для класів.

`.card .price{…}` виконує оформлення тегу із класом `price`, що розміщений у теге із класом `card`.

Псевдокласи CSS

Псевдоклас CSS – це ключове слово, яке додається до селектора та дозволяє визначити поведінку тега у певних ситуаціях (наприклад наведення вказівника миші та елемент сторінки). Серед ключових їх завдань є підсилення роботи звичних селекторів, дозволяють вибирати компоненти Web-документа із урахуванням стану і розмірів по відношенню до інших елементів.

Синтаксично псевдокласи задаються через двокрапку, після запису основного селектора (приклад 8).

Приклад 8.

```
.red_link:hover{…}
```

До найбільш вживаних псевдокласів варто віднести такі: `:link`, `:hover`, `:active`, `:focuse`, `:visited`, `:root`, `:first-child`, `:last-child`, `:nth-child`, `:valid`, `:invalid`, `:required`, `:optional`, `:checked`, `:disable`, `:enabled`.

Розглянемо докладніше кожен із них.

`:link` – дозволяє застосовувати набір правил до елементів, які виконують роль посилань на інші ресурси.

Приклад 9.

```
a:link {color: red;}
.sample: {font-size: 16px;}
```

`:hover` – елемент змінює свій зовнішній вигляд, тоді коли користувач наводить курсор миші на заданий елемент.

`:active` – застосовує оформлення до HTML елемента у той момент, коли він активується користувачем.

`:focus` – спрацьовує тоді, коли елемент активується для внесення певної інформації. Досить часто такі налаштування застосовуються до полів форми.

:visited – вибирає всі посилання, які були відвідані користувачем.

:root – визначає кореневий елемент дерева документа. Для web-сторінки це буде тег html. Цей псевдоклас часто використовують для задання CSS змінних.

:first-child – застосовує правила оформлення до першого елемента у заданому контейнері.

:last-child – працює аналогічно до попереднього, але у цьому випадку визначається останній елемент у заданому контейнері.

:nth-child – дозволяє знаходити один або більше елементів у контейнері.

:valid – застосовує описані правила оформлення до всіх полів введення, які проходять валідацію.

:invalid – визначає всі поля, які не проходять валідацію.

:required – вибирає всі теги <input>, які мають атрибут required.

:optional – вибирає всі теги <input>, які не мають атрибут required. Він дозволяє формам легко відзначати необов'язкові поля, і надавати їм відповідні стилі.

:checked – дозволяє застосовувати стилі оформлення до всіх елементів форми, які мають тип radio, checkbox або option.

:disabled – визначає всі елементи Web-сторінки, які є вимкнутими. Вимкнутими елементами вважаються ті, які не можуть бути активованими (їх не вибрати або натиснути на них кнопкою миші).

:enabled – діє аналогічно до попереднього, але у цьому випадку вибирає на сторінці чи у заданому контейнері всі активні елементи.

Окрім описаних вище існують такі як: :any, :any-link, :default, :defined, :dir(), :empty, :first, :first-of-type, :fullscreen, :indeterminate, :in-range, :lang(), :last-of-type, :left, :not(), :nth-last-child(), :nth-last-of-type(), :nth-of-type(), :only-child, :only-of-type, :out-of-range, :read-only, :read-write, :right, :scope, :target. Проте вони не часто використовуються для верстки сучасних web-документів.

Псевдоелементи CSS та їх застосування

Окрім псевдокласів у специфікації CSS присутнє таке поняття як псевдоелементи. Вони дозволяють утворювати нові віртуальні теги та стилізувати їх. Для їх оголошення використовують подвійну двокрапку. Серед найбільш використовуваних із них варто назвати такі як: `::after`, `::before`, `::first-letter`, `::first-line`, `::selection` тощо. Розглянемо їх особливості докладніше

`::after` – дозволяє створити псевдоелемент, який є останнім нащадком вибраного елемента. За замовчуванням є рядковим елементом.

`::before` – створює псевдоелемент, який є першим нащадком обраного елемента. Часто використовується для додавання додаткового контенту в елемент за допомогою властивості content. За замовчування є рядковим.

`::first-letter` – надає можливість застосовувати стилі до першої літери першого рядка контейнерного елемента, але тільки якщо немає іншого попередньо розміщеного зображення або таблиці.

`::first-line` – схожий за дією до попереднього, але дозволяє застосовувати стилі до першого рядка тексту контейнерного елемента. Варто зауважити, що довжина першого рядка залежить від багатьох факторів, включаючи ширину елемента, ширину документа і розмір шрифту тексту.

`::selection` – дозволяє застосувати стилі до частини документа, який був виділений користувачем (наприклад, за допомогою миші).

CSS властивості

У сучасних стандартах CSS3 існує дуже багато властивостей, які дозволяють форматувати та оформляти web-сторінки. Їх умовно можна розділити на такі великі групи: CSS властивості для встановлення розмірів та відступів; інструментарій для керуванням потоком; засоби форматування та декорування тексту; властивості для задання анімації

та динамічних ефектів тощо. Звичайно їх значно більше, проте вони діють за одним тими ж принципами.

Тому, перш ніж описати їх, розкриємо основні рекомендації, які варто дотримуватися при верстці HTML документів.

1. Якщо до одного і того ж тегу застосовуються декілька CSS правил (можуть бути різні класи), то у такому випадку властивості комбінуються.

2. У випадку конфлікту селекторів (наприклад у двох різних класах присутня однакова властивість із різними значеннями) браузер застосовує правила пріоритетів для визначення селектора, які будуть застосовані для оформлення тегу. Пріоритетність селекторів можна представити таким чином.

Найбільший пріоритет має селектор #id.

Вкладені класи завжди є важливішим за звичний клас

`.main.bage → .bage`

Вкладені теги важливіші за просто описаний тег у CSS

`.main a → a`

Тег із класом є пріоритетнішим за звичний клас

`p.title → .title`

Клас завжди є важливішим за звичний тег, означений у CSS

`.title → p`

Орієнтуючись на них завжди можна точно спрогнозувати поведінку браузера, а отже чітко форматувати саму сторінку сайта.

Основні типи значень у CSS

Важливим є розуміння, які значення можуть набувати властивості у селекторах. Загалом їх ділять на такі основні групи: абсолютні і відносні числові одиниці; ключові слова; кольори; функції та рядки. Дамо коротку характеристику для кожного із них.

Абсолютні величини задають точно значення властивостей, для яких вони є характері, у пікселях

Приклад 10.

```
height: 100px;
```

У цьому випадку вказується висота елемента у 100 пікселів.

Відносні одиниці вимірювання дозволяють вказати значення розмірів по відношенню до батьківського елемента. До них відносять: відсотки (розмір розраховується відносно батьківського елемента, наприклад для тега `body` таким буде `html`, а для нього власне сам браузер – точніше та частина, у якій відображається сам сайт); `vw` – розраховується як одна сота від ширини екрана; `vh` – аналогічна до попередньої відносна одиниця, тільки у цьому випадку за основу береться висота екрана; `em` – одиниця вимірювання для шрифтів, яка задає розмір тексту по відношенню до розміру шрифту батьківського елементу; `rem` – працює аналогічно, але розрахунок відбувається на основі базового розміру шрифту, який встановлений для тегу `body`.

Ключові слова задають певне значення, яке характеризує стан певної властивості (наприклад `display: flex;`).

Кольори у CSS можна описати трьома способами: за допомогою ключових слів, які позначають колір (наприклад `color:red;` – задає червоний колір тексту); у вигляді шістнадцяткового значення (`color:#23aaff;`); за допомогою ключого слова `rgb(255,40,50)`, де у дужках подаються значення трьох кольорів (червоного, зеленого і синього) за допомогою числа із інтервалу від 0 до 255 включно. Більш розширений формат `rgba`, дозволяє задати також прозорість четвертим числом із значенням від 0 до 1. Нуль відповідає відсутністю кольору, а 1 – повному його відображенню.

Також є можливість задати колір за допомогою моделі `hsl` або `hsla`. Ключовою відмінністю є те, що колір задається трьома значеннями у відсотках.

Функції CSS є спеціалізованими підпрограмами, які дозволяють генерувати та обраховувати значення. Найбільш застосовуваним із них є: `attr()`, `calc()`, `linear-gradient()`, `radial-gradient()`, `repeating-linear-gradient()`, `repeating-radial-gradient()` тощо.

Рядкові значення найчастіше зустрічаються при заданні сімейства шрифтів або у властивості content. Наприклад `font-family: ”Arial”, sans serif;` або ж content: `“Hello World!!!”`.

Спеціалізовані директиви CSS

Окремо розглянемо директиви CSS. Вони являються спеціалізованими правилами, які дозволяють змінити відображення або ж поведінку елементів сторінки у певній, завчасно визначеній ситуації.

Директиви розпочинаються із запису символа @, після якого записується одне із ключових слів. Їх ділять на стандартні правила та вкладені. Коротко охарактеризуємо перші.

`@charset` – задає кодування, яке у подальшому буде використовуватися браузером. Наприклад правило `@charset "UTF-8";` вказує, що сторінка використовує кодування символів UTF-8.

`@import` – дозволяє завантажити та використовувати зовнішній CSS файл (`@import "new.css";`).

`@namespace` дозволяє застосовувати CSS для таких мов розмітки як XML та XHTML.

Вкладені директиви містять у собі додаткові оголошення, які можуть бути описом певних завчасно визначених ситуацій.

`@document` – спеціалізована директива, яка визначає умови застосування стилей для всієї сторінки

Приклад 11.

```
@document(
    url(https://samplpe.com/)
    regexp("https:// ….”))
    {
        body{font-family: Arial;}
    }
```

`@font-face` – дозволяє завантажувати користувацькі шрифти.

Приклад 12.

```
@font-face{
    font-family:'MyWebFont';
    url('Fonts/MyWebFont.woff2')
}
```

`@media` – містить умовні інструкції, які застосовують задані стилі, у залежності від певної умови.

Приклад 13.

```
@media (max-width: 600px){
    .screen{
        display:none;
    }
}
```

Останній приклад скрипту дозволяє вимикати властивість `display` для значень ширини екрану 600px і менше.

Властивості керування потоком документу, заданням розмірів елементів

Поток документу HTML називається процес відображення елементів Web-сторінки у вікні браузера. Керування ним відбувається тільки за допомогою каскадних таблиць стилів. Основною одиницею для відображення матеріалу (текстового, графічного, відео та аудіо) є бокс – прямокутна область сторінки, у якій розміщується контент.

Поведінку блока на сторінці задається ключовою властивістю `display`. Серед основних значень, яких може набувати ця характеристика, є такі: `block, inline-block, inline, flex, grid, table`. Розглянемо їх детальніше.

Значення `block`. За замовчуванням браузер висловлює таке значення для тегів `header, section, footer, div, h1, …, h6, p, ul, ol` тощо. Для всіх тегів HTML, до яких застосовується властивість із описаним значенням характерні такі особливості.

1. Висота боксу підлаштовується під висоту контенту.
2. Займає весь доступний простір по ширині.
3. Для такого тегу є можливість задати значення висоти, ширини, внутрішні та зовнішні відступи.
4. Обов’язковим є примусове перенесення слів на новий рядок, якщо у попередньому речення не поміщається повністю.

Основні властивості, які задають параметри бокса, якщо для нього встановлено значення властивості `display: block`.

`width` – задає ширину контенту (зазвичай використовуються такі одиниці як px, %, auto).

`height` – висота контенту (px, auto).

padding – внутрішні відступи у боксі від меж до контенту. На рівні із цією властивістю також часто використовуються властивості padding-left, padding-right, padding-top, padding-bottom. Вони дозволяють задати внутрішній відступ з однієї сторони (зліва, справа, зверху, знизу).

margin – задає зовнішні відступи. Аналогічно дозволяється використовувати і такі властивості як: margin-left, margin-right, margin-top, margin-bottom. Їх дія аналогічна до попередньої властивості, з тією різницею, що у цьому випадку задається зовнішній відступ із певної сторони.

Властивість border задає для боксу рамки. Їх основними компонентами є: товщина, стиль накреслення і колір. Вони задаються за допомогою властивостей border-width, border-style та border-color. Окремі сторони дозволяється модифікувати за допомогою команд border-left, border-right, border-top, border-bottom.

Значення inline характерне для тегів, які використовуються для форматування фразового контенту. Типовими інструкціями HTML є: span, a, strong, em, b, i, time тощо.

Значення inline-block. Це гібридне значення, яке дозволяє розміщувати елементи із значенням block у рядок. Фактично поєднує особливості застосування першого та другого.

Flex – це більше ніж окреме значення для властивості display. Це взагалі окрема технологія, яка спростила створення гнучких та адаптивних сторінок для сайтів. Вона була розроблена як модель однорядкового макетування елементів інтерфейсу і, що також важливо, як один з методів розподілу простору між елементами у рамках контейнеру flexbox. Для неї характерно гнучкість у розміщенні компонентів на сторінці.

При верстці за допомогою цієї технології варто оперувати елементами із точки зору двох осей – основної та поперечної осей (рис. 2). Основна вісь визначається властивістю flex-direction, а поперечна вісь проходить перпендикулярно їй. Все, що ми робимо із компенентами flexbox, відноситься до цих осей, тому варто з самого початку зрозуміти, як вони працюють.

До основних властивостей, які дозволяють налаштувати розміщення елементів у flex-контейнері, відносять: `flex-direction`, `justify-content`, `flex-wrap`, `order`, `align-items`, `align-self`, `align-content`, `flex-grow`, `flex-shrink` і `flex-basis`.

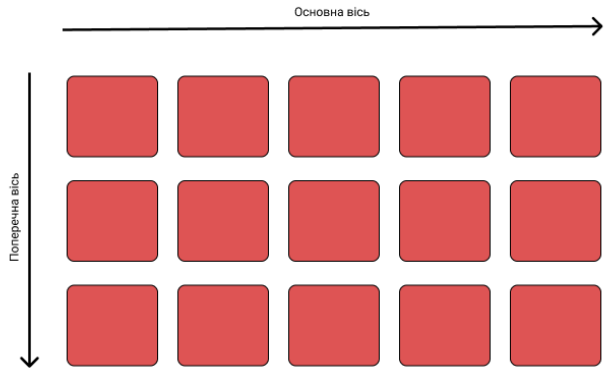


Рис. 2. Основна та поперечна вісь технології Flex

Значення `grid` є прикладом ще однієї інтегрованої технології, яка дозволяє виконувати позиціонування об’єктів на сторінці як по горизонталі так і по вертикалі. По своїй суті вона дуже схожа із таблицею, але має значно гнучкіший та простіший по свої суті функціонал.

Серед основних властивостей, які використовуються у рамках CSS Grid варто назвати: `grid-template-columns`, `grid-template-rows`, `grid-template-areas`, `grid-template`, `grid-column-gap`, `grid-row-gap`, `grid-gap`, `justify-items`, `align-items`, `justify-content`, `align-content`, `grid-auto-columns`, `grid-auto-rows`, `grid-auto-flow`, `grid`, `grid-column-start`, `grid-column-end`, `grid-row-start`, `grid-row-end`, `grid-column`, `grid-row`, `grid-area`, `justify-self`, `align-self`.

Властивості для форматування тексту

Форматування тексту є важливою частиною верстки сучасних web-сайтів. Для налаштування зовнішнього вигляду символів використовують такі властивості.

`font-family` – задає зовнішній вигляд та вказується сімейство шрифтів.

`font-size` – вказує для вибраного елемента розмір шрифту.

`font-style` – задає накреслення для шрифту (може мати такі значення `normal` – звичне накреслення, `italic` – курсив, `oblique` – похилий, `inherit` – наслідування шрифту батьківського елемента).

`font-variant` – вказує на представлення малих літер.

`font-weight` – дозволяє задати насиченість для шрифту (bold – напівжирний та значення від 100 до 900).

`letter-spacing` – відстань між символами.

`word-spacing` – задає додаткову відстань між словами.

`color` – вказується колір тексту.

`text-indent` – встановлюється відступ першого рядка для абзаца.

`text-decoration` – дозволяє представити текст наступним чином: `underline` – підкреслення, `overline` – лінія над текстом, `line-through` – перекреслений текст, `capitalize` – текст записується за допомогою великих літер, але розміром маленьких, `lowercase` – нижній регістр, `uppercase` – верхній регістр.

`vertical-align` – надає можливість записати текст у: `super` – верхньому та `sub` – нижньому індексі.

Властивості CSS для анімації та ефектів

CSS анімації дозволяє реалізувати анімацію переходів (transitions) від однєї конфігурації CSS стилю до іншої. Їх реалізація складаються з двох базових компонентів: стилю, котрий описує суть стилю та набору ключових кадрів (keyframes), які задають початковий та кінцевий стан стилю анімації. Присутня також можливість визначення проміжних точок анімації.

Анімація за елементів сторінки за допомого CSS має ряд важливих переваг над процесом її програмування за допомогою JavaScript.

1. Легкість створення простих анімацій, які при цьому не навантажують систему додатковими скриптами.

2. Анімації добре функціонують, навіть при помірному навантаженні на систему.

3. Процес анімації повністю контролюється браузером, а отже таким чином дозволяєте йому оптимізувати його роботу.

Властивості, які дозволяють описати анімацію у CSS, є наступними: `animation`, `animation-delay`, `animation-direction`, `animation-duration`, `animation-iteration-count`, `animation-name`, `animation-play-state`, `animation-timing-function`, `animation-fill-mode`.

Стилі кодування

При написанні будь-якої програми важливо дотримуватися певного стилю написання коду, який спрощує його розуміння як самим розробником так і сторонніми фахівцями. Стиль кодування включає стандартизацію форматування тексту програми, назви властивостей, порядок їх оголошення тощо.

Варто зауважити, що кожен фахівець самостійно вибирає способи оформлення власного коду програми, яку він пише, проте у рамках роботи команди програмістів, цей стиль узгоджується та стандартизується, з метою уникнення неузгодженостей.

Серед основних кодгайдів, які часто наслідують та дотримуються при розробці, варто вказати на такі: MDO, Google та Idiomatic CSS.

ДОДАТОК 1

Перелік літератури та корисних посилань із верстки сайтів

Література

1. Пасічник В. В., Пасічник О. В., Угрин Д. І. Веб-технології: підручник. Львів : Магнолія 2013. 335 с.

2. Романюк О. Н., Катєльніков Д. І., Косовець О. П. Веб-дизайн і комп'ютерна графіка: навчальний посібник. Вінниця : ВНТУ, 2007. 142 с.

3. Самсонов В. В., Єрохін А. Л. Методи та засоби Інтернет-технологій. Х. : СМІТ, 2008. 264 с.

4. Duckett J. HTML and CSS: Design and Build Websites. Wiley, 2014. 512 p.

5. Freeman A. Pro AngularJS (Expert's Voice in Web Development). Apress, 2014. 688 p.

6. McGrath M. HTML, CSS & JavaScript in easy steps. In Easy Steps Limited, 2020. 480 p.

7. Minnick J. Responsive Web Design with HTML 5 & CSS (MindTap Course List). Cengage Learning, 2020. 640 p.

8. Stefanov S. React: Up & Running: Building Web Applications. O'Reilly Media, 2016. 222 p.

Корисні посилання

1. Офіційна сторінка Codecademy. URL: https://www.codecademy.com/

2. Офіційна сторінка W3C. URL: https://www.w3.org.

3. HTML 5.2 W3C Recommendation. URL: https://www.w3.org/TR/2017/REC-html52-20171214/.

4. Основи Web UI розробки. URL: https://courses.prometheus.org.ua/ courses/course-v1:LITS+114+2017_T4/about.

ДОДАТОК 2

Лабораторна робота № 1.

Тема. Поняття боксу. Семантична структура. Позиціонування елементів сторінки.

Мета. Набути умінь та навичок створення семантичної структури документів

Програмне забезпечення. Текстові редактори Notepad++, Atom, Sublime Text, графічні редактори GIMP, Inkscape.

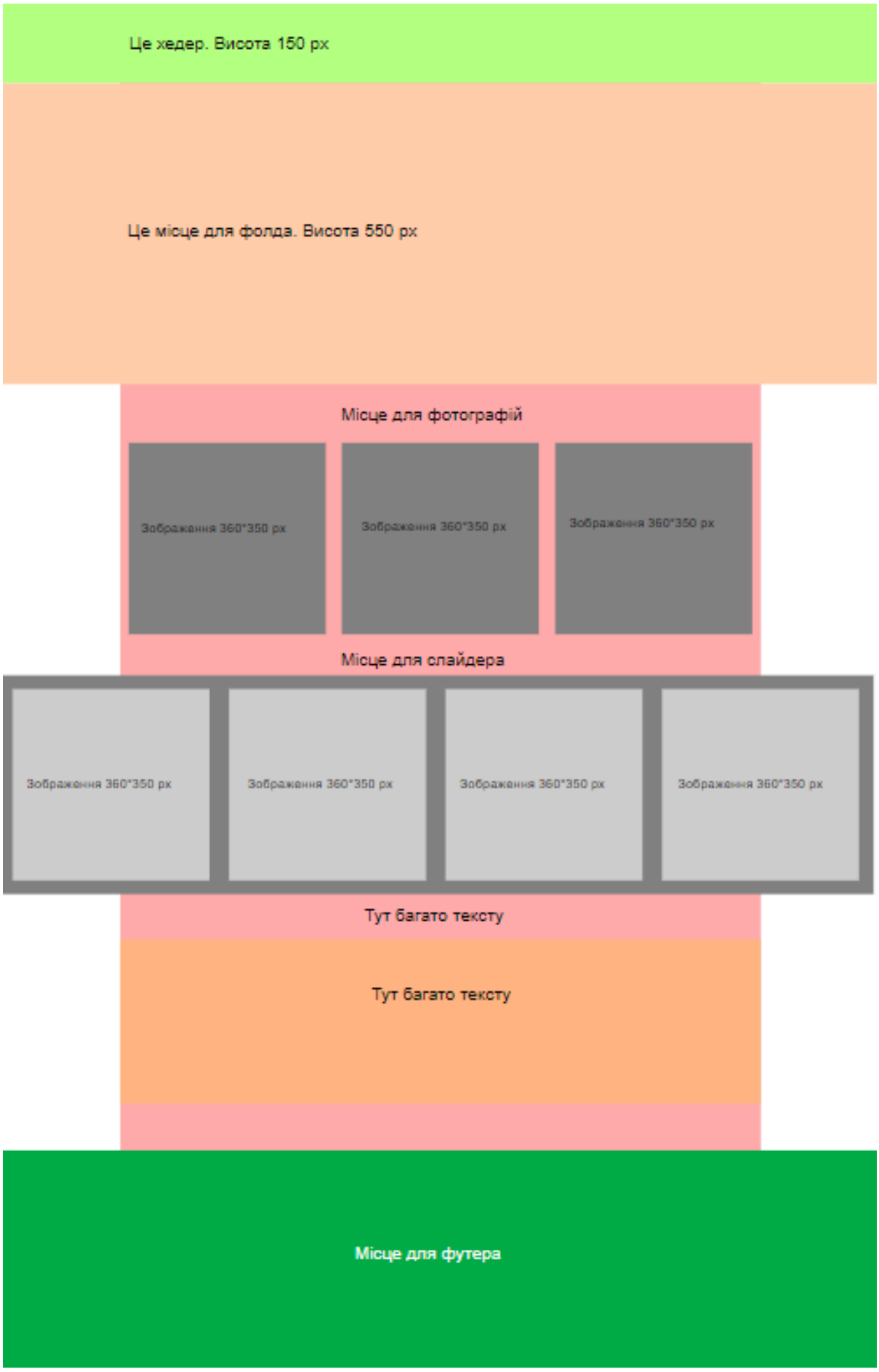
Контрольні запитання.

1. Основні теги HTML.
2. Поняття CSS та способи підключення каскадних таблиць стилів.
3. Поняття блокової верстки сайту.
4. Властивість CSS float.

Завдання до лабораторного заняття

1. Зверстати HTML сторінку відповідно до Вашого варіанту.
2. Структура блоків, їх розміри та розміщення мають відповідати зразку.
3. Колір блоків має відповідати кольору цих елементів на зразка.
4. Текст повинен бути таким, як представлено на рисунку (у блоці, де необхідно ввести багато тексту слід розмістити великий текст).
5. Каскадна таблиця стилів має бути розміщена в окремому файлі та підключатися до HTML файлу.
6. Всі фали мають бути у репозиторії на сайті github.com або gitlab.com із назвою Прізвище_Lab1.
7. Посилання на репозиторій слід надіслати на пошту викладача.

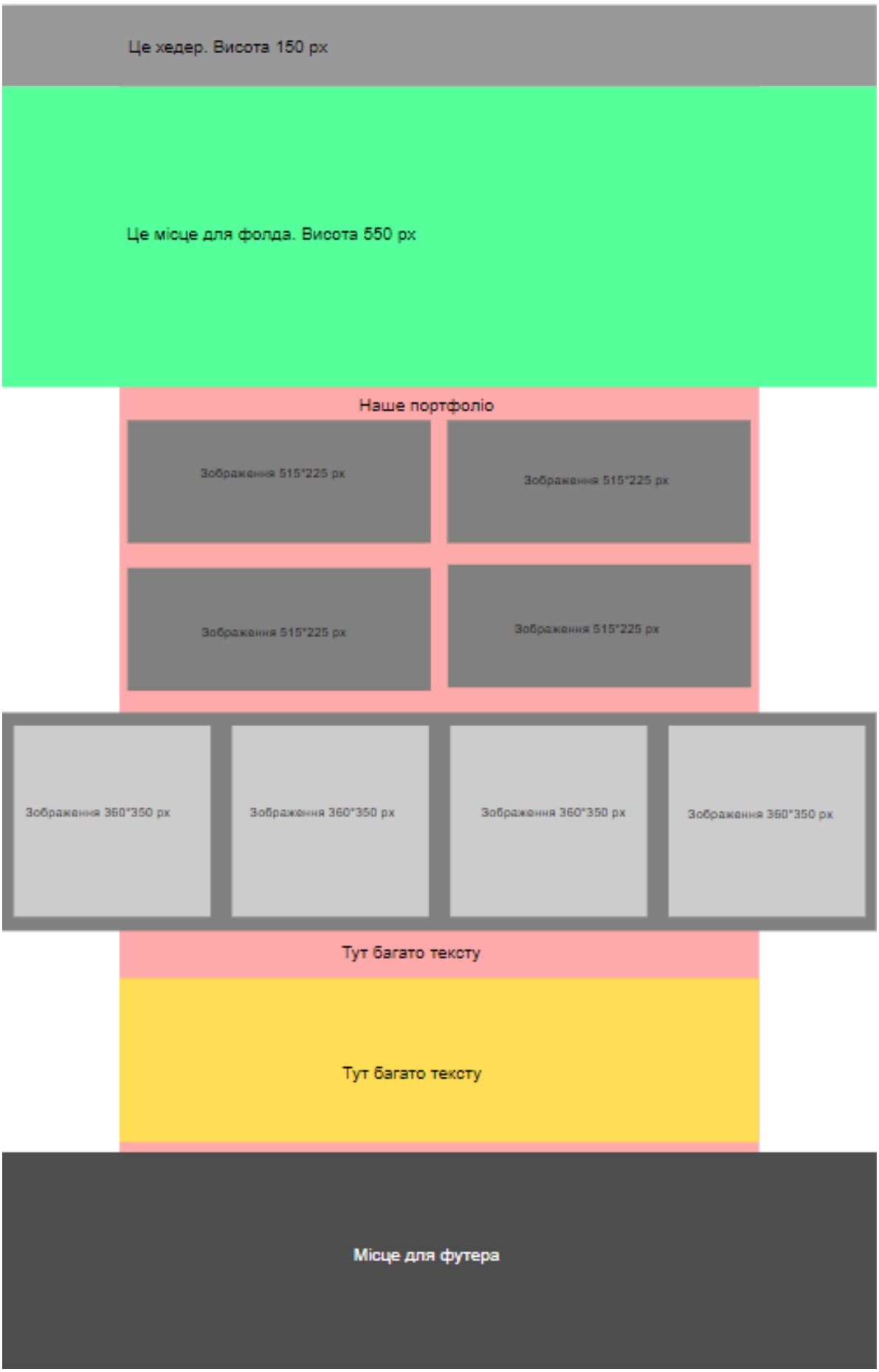
Варіант 1


Це хедер. Висота 150 px
Це місце для фолда. Висота 550 px
Місце для фотографій
Зображення 360*350 px
Зображення 360*350 px
Зображення 360*350 px
Місце для слайдера
Зображення 360*350 px
Зображення 360*350 px
Зображення 360*350 px
Зображення 360*350 px
Тут багато тексту
Тут багато тексту
Місце для футера

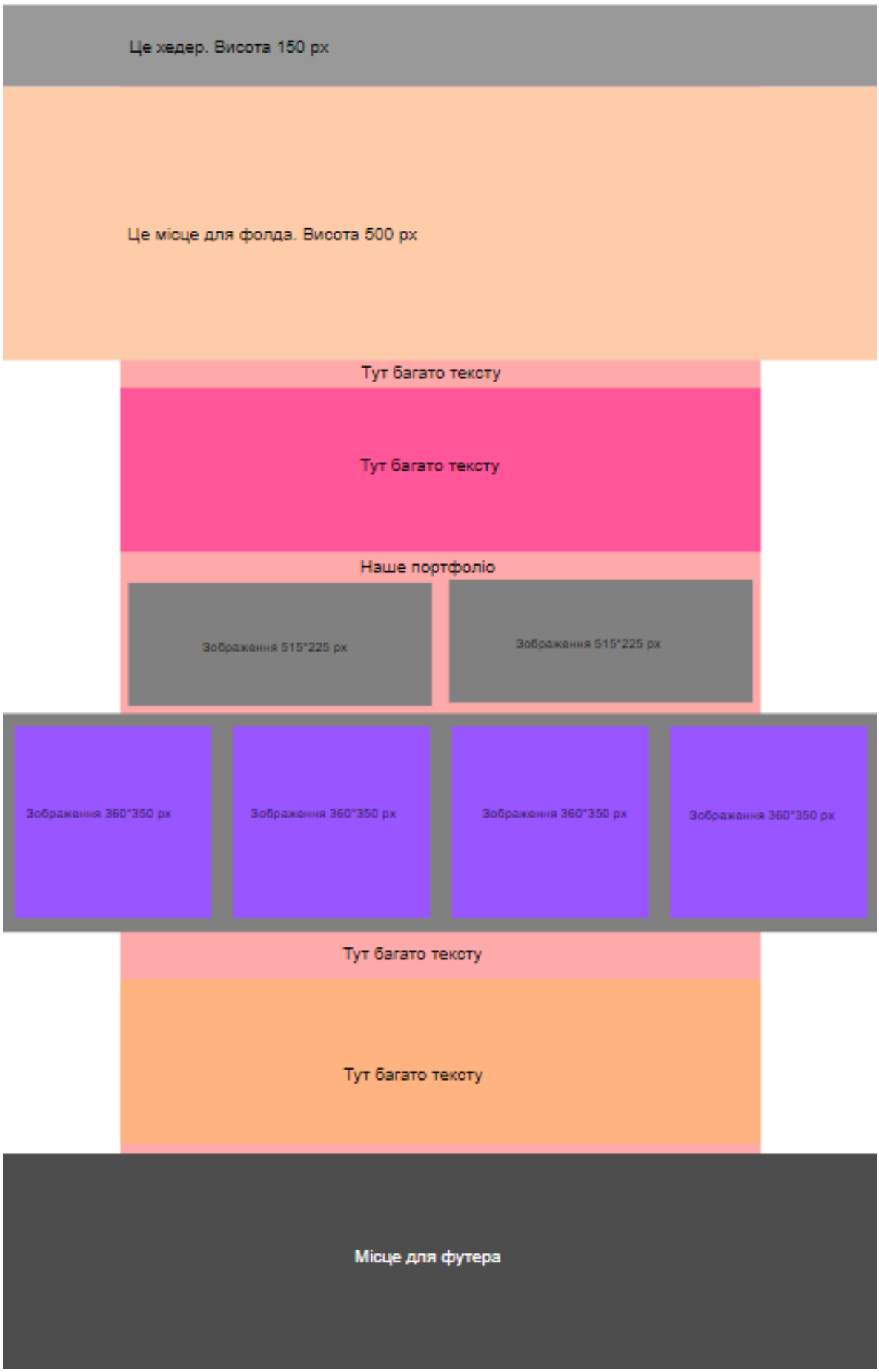
Варіант 2


Це хедер. Висота 150 px
Це місце для фолда. Висота 550 px
Тут багато тексту
Тут багато тексту
Зображення 360*350 px
Опис зображення
Тут багато тексту
Тут багато тексту
Місце для футера

Варіант 3


Це хедер. Висота 150 px
Це місце для фолда. Висота 550 px
Наше портфоліо
Зображення 515*225 px
Зображення 515*225 px
Зображення 515*225 px
Зображення 515*225 px
Зображення 360*350 px
Зображення 360*350 px
Зображення 360*350 px
Зображення 360*350 px
Тут багато тексту
Тут багато тексту
Місце для футера

Варіант 4


Це хедер. Висота 150 px
Це місце для фолда. Висота 500 px
Тут багато тексту
Тут багато тексту
Наше портфоліо
Зображення 515*225 px
Зображення 515*225 px
Зображення 360*350 px
Зображення 360*350 px
Зображення 360*350 px
Зображення 360*350 px
Тут багато тексту
Тут багато тексту
Місце для футера

ДОДАТОК 3

Лабораторна робота № 2.

Тема. Форматування тексту. Оформлення зображення. Фіксована верстка. Створення меню. Псевдокласи.

Мета. Набути умінь та навичок створення семантичної структури документів

Програмне забезпечення. Текстові редактори Notepad++, Atom, Sublime Text, графічні редактори GIMP, Inkscape.

Контрольні запитання.

1. Теги для завантаження зображення в HTML5.
2. Форматування тексту за допомогою CSS3.
3. Створення меню за допомогою HTML5 і CSS3.
4. Псевдоклас .hover.
5. Посилання.

Завдання до лабораторного заняття

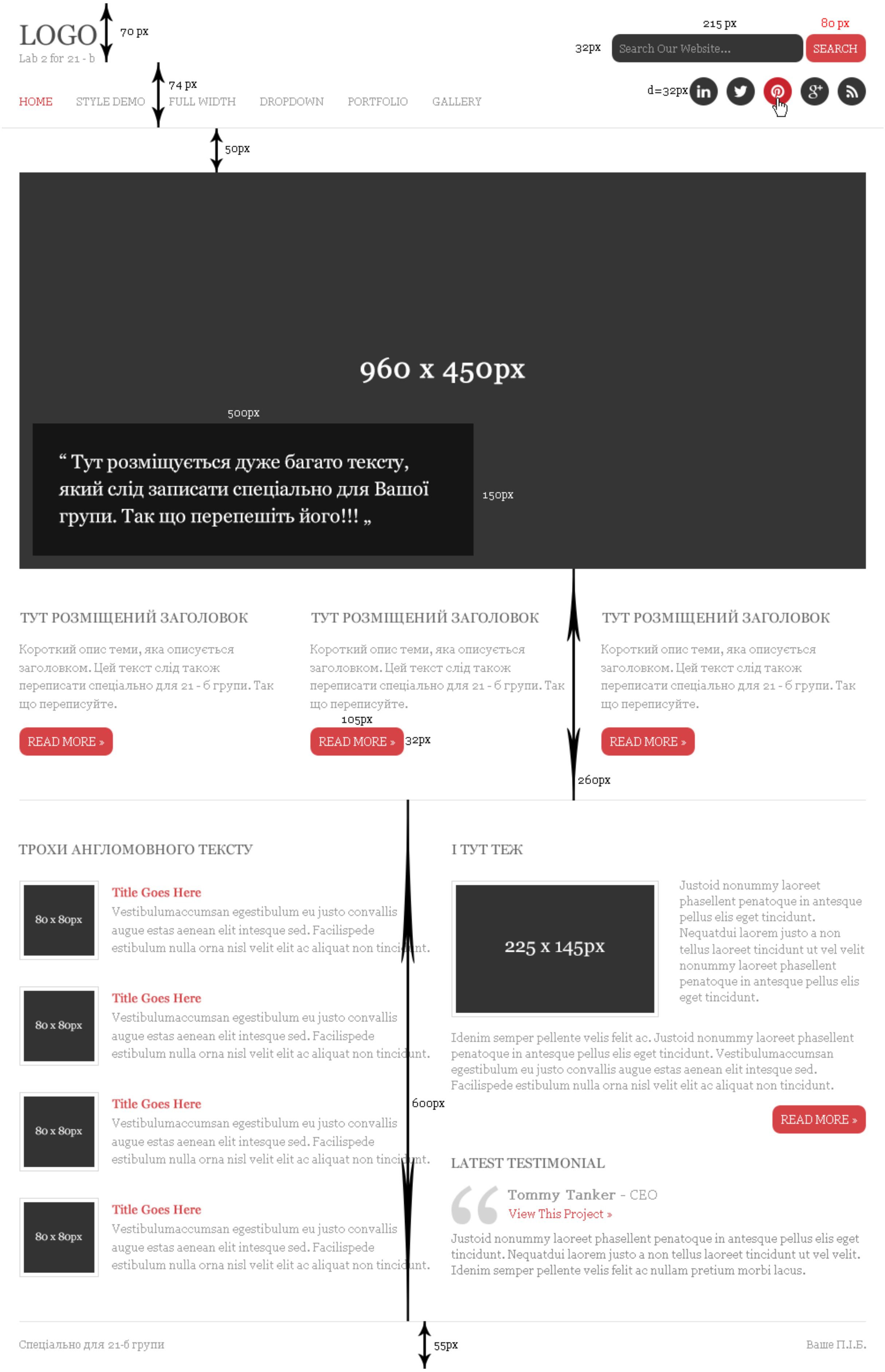
Звертстати основну сторінку сайта у відповідності до запропонованої схеми (варіанту). Він повинен відповідати наступним вимогам.

1. Розміри блоків та відступів мають бути такими, як на схемі.
2. Ширина основного блока 980px.
3. Необхідно підібрати зображення у відповідності до тематики сайта.
4. Елементи меню мають змінюватися при наведенні на нього кнопки миші.
5. Форми введення мають бути замінені на візуально подібні блоки.
6. Колір шрифту, його розмір має наближатися до оригіналу, шрифт - Georgia.
7. Забороняється використовувати таблиці.
8. Всі фали мають бути у репозиторії на сайті github.com або gitlab.com із назвою **Прізвище_Lab2.**
9. Посилання на репозиторій слід надіслати на пошту викладача.

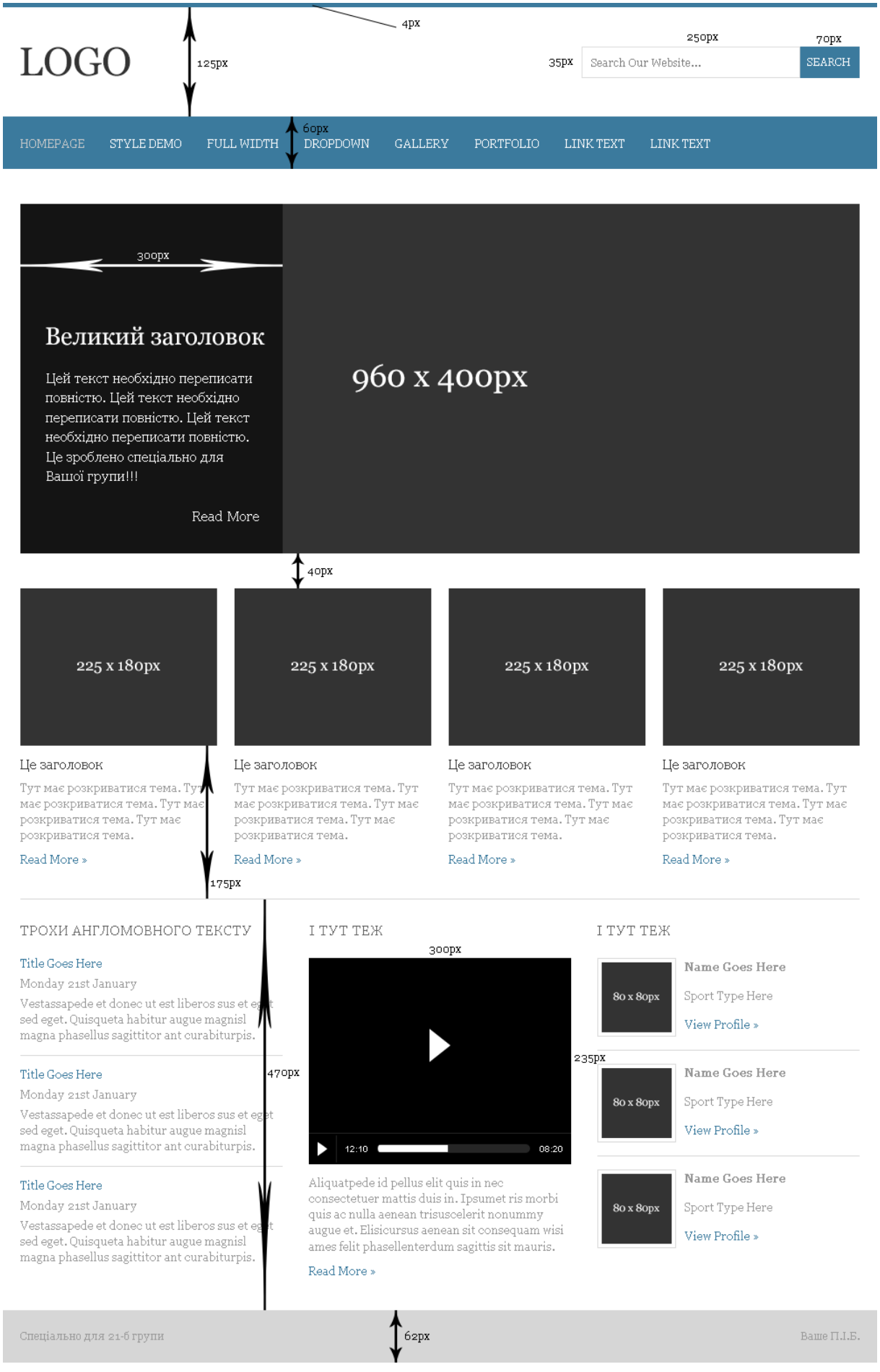
Тематика варіантів

Варіант 1. Спорт.
Варіант 2. Освіта.
Варіант 3. Захист природи.
Варіант 4. Заклад швидкого харчування.

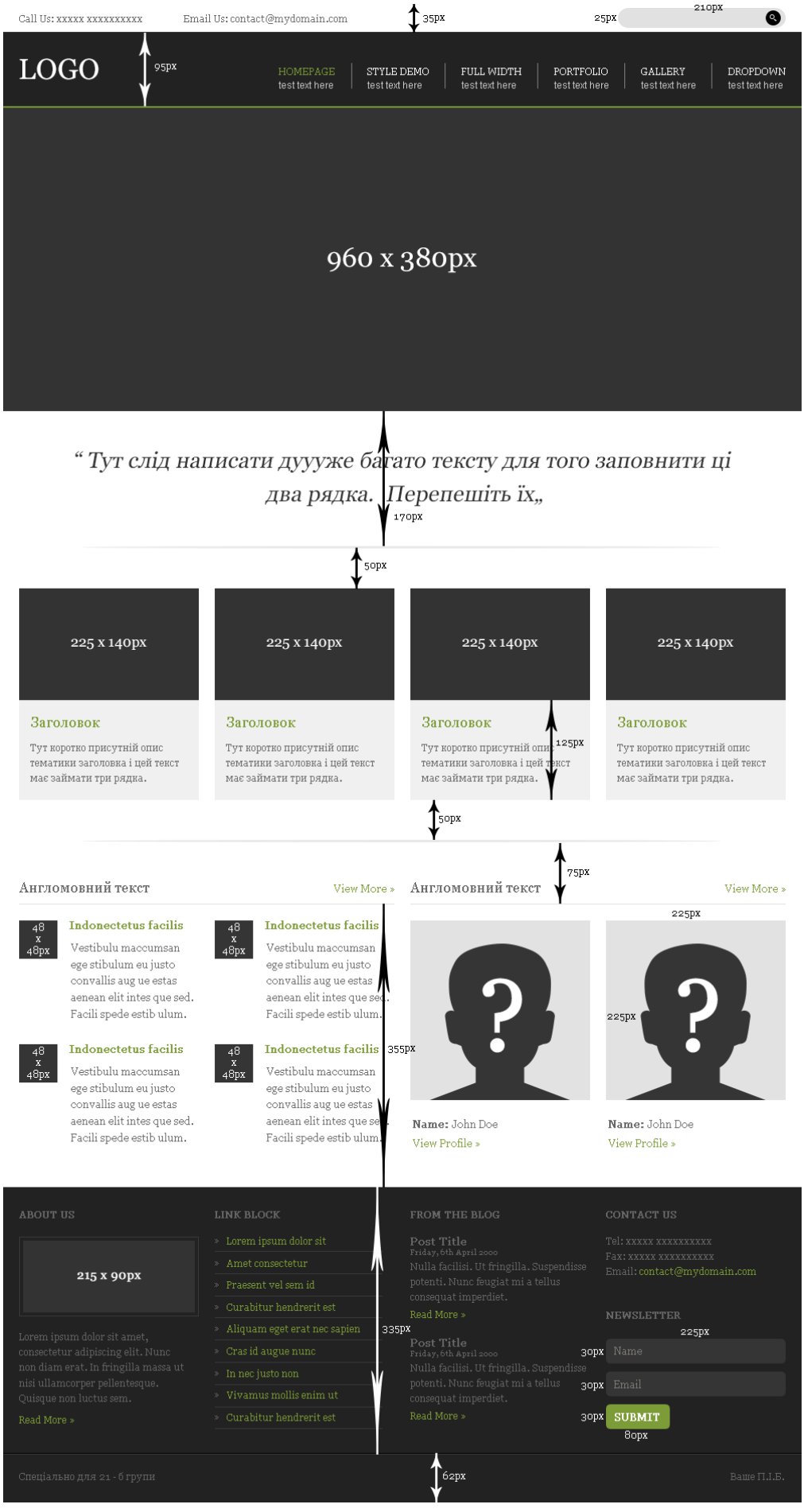
Варіант 1


LOGO
Lab 2 for 21 - b
70 px
74 px
215 px
80 px
32px
Search Our Website...
SEARCH
d=32px
HOME
STYLE DEMO
FULL WIDTH
DROPDOWN
PORTFOLIO
GALLERY
50px
960 x 450px
500px
“ Тут розміщується дуже багато тексту, який слід записати спеціально для Вашої групи. Так що перепешіть його!!! „
150px
ТУТ РОЗМІЩЕНИЙ ЗАГОЛОВОК
Короткий опис теми, яка описується заголовком. Цей текст слід також переписати спеціально для 21 - б групи. Так що переписуйте.
READ MORE »
ТУТ РОЗМІЩЕНИЙ ЗАГОЛОВОК
Короткий опис теми, яка описується заголовком. Цей текст слід також переписати спеціально для 21 - б групи. Так що переписуйте.
105px
READ MORE »
32px
ТУТ РОЗМІЩЕНИЙ ЗАГОЛОВОК
Короткий опис теми, яка описується заголовком. Цей текст слід також переписати спеціально для 21 - б групи. Так що переписуйте.
READ MORE »
260px
ТРОХИ АНГЛОМОВНОГО ТЕКСТУ
80 x 80px
Title Goes Here
Vestibulumaccumsan egestibulum eu justo convallis augue estas aenean elit intesque sed. Facilispede estibulum nulla orna nisl velit elit ac aliquat non tincidunt.
80 x 80px
Title Goes Here
Vestibulumaccumsan egestibulum eu justo convallis augue estas aenean elit intesque sed. Facilispede estibulum nulla orna nisl velit elit ac aliquat non tincidunt.
80 x 80px
Title Goes Here
Vestibulumaccumsan egestibulum eu justo convallis augue estas aenean elit intesque sed. Facilispede estibulum nulla orna nisl velit elit ac aliquat non tincidunt.
80 x 80px
Title Goes Here
Vestibulumaccumsan egestibulum eu justo convallis augue estas aenean elit intesque sed. Facilispede estibulum nulla orna nisl velit elit ac aliquat non tincidunt.
600px
І ТУТ ТЕЖ
225 x 145px
Justoid nonummy laoreet phasellent penatoque in antesque pellus elis eget tincidunt. Nequatdui laorem justo a non tellus laoreet tincidunt ut vel velit nonummy laoreet phasellent penatoque in antesque pellus elis eget tincidunt.
Idenim semper pellente velis felit ac. Justoid nonummy laoreet phasellent penatoque in antesque pellus elis eget tincidunt. Vestibulumaccumsan egestibulum eu justo convallis augue estas aenean elit intesque sed. Facilispede estibulum nulla orna nisl velit elit ac aliquat non tincidunt.
READ MORE »
LATEST TESTIMONIAL
Tommy Tanker - CEO
View This Project »
Justoid nonummy laoreet phasellent penatoque in antesque pellus elis eget tincidunt. Nequatdui laorem justo a non tellus laoreet tincidunt ut vel velit. Idenim semper pellente velis felit ac nullam pretium morbi lacus.
Спеціально для 21-б групи
55px
Ваше П.І.Б.

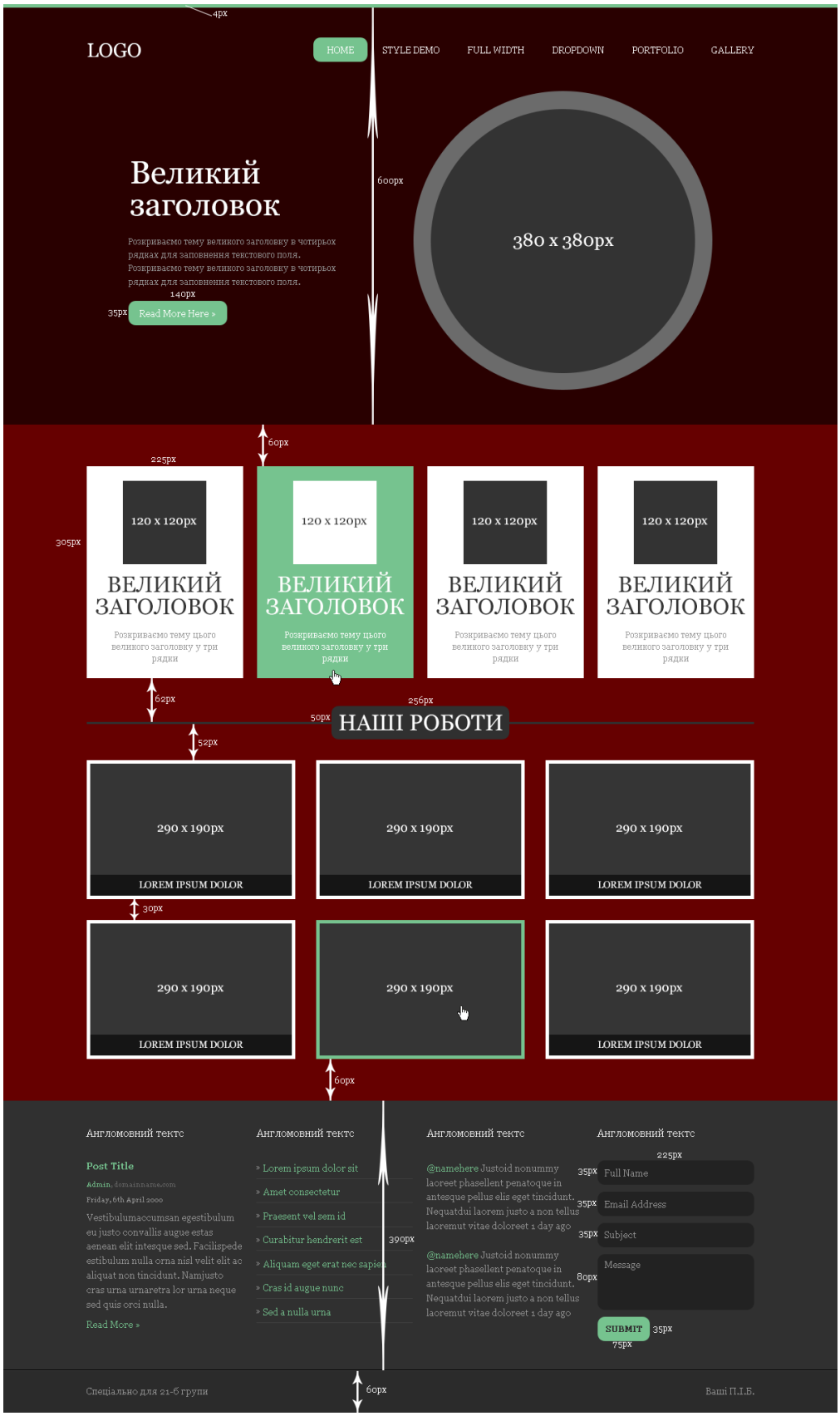
Варіант 2


LOGO
4px
125px
250px
70px
35px
Search Our Website...
SEARCH
HOMEPAGE
STYLE DEMO
FULL WIDTH
60px
DROPDOWN
GALLERY
PORTFOLIO
LINK TEXT
LINK TEXT
300px
Великий заголовок
Цей текст необхідно переписати повністю. Цей текст необхідно переписати повністю. Цей текст необхідно переписати повністю. Це зроблено спеціально для Вашої групи!!!
Read More
960 x 400px
40px
225 x 180px
225 x 180px
225 x 180px
225 x 180px
Це заголовок
Тут має розкриватися тема. Тут має розкриватися тема. Тут має розкриватися тема. Тут має розкриватися тема.
Read More »
Це заголовок
Тут має розкриватися тема. Тут має розкриватися тема. Тут має розкриватися тема. Тут має розкриватися тема.
Read More »
Це заголовок
Тут має розкриватися тема. Тут має розкриватися тема. Тут має розкриватися тема. Тут має розкриватися тема.
Read More »
Це заголовок
Тут має розкриватися тема. Тут має розкриватися тема. Тут має розкриватися тема. Тут має розкриватися тема.
Read More »
175px
ТРОХИ АНГЛОМОВНОГО ТЕКСТУ
Title Goes Here
Monday 21st January
Vestassapede et donec ut est liberos sus et eget sed eget. Quisqueta habitur augue magnisl magna phasellus sagittitor ant curabiturpis.
Title Goes Here
Monday 21st January
Vestassapede et donec ut est liberos sus et eget sed eget. Quisqueta habitur augue magnisl magna phasellus sagittitor ant curabiturpis.
Title Goes Here
Monday 21st January
Vestassapede et donec ut est liberos sus et eget sed eget. Quisqueta habitur augue magnisl magna phasellus sagittitor ant curabiturpis.
470px
І ТУТ ТЕЖ
300px
12:10
08:20
Aliquatpede id pellus elit quis in nec consectetuer mattis duis in. Ipsumet ris morbi quis ac nulla aenean trisuscelerit nonummy augue et. Elisicursus aenean sit consequam wisi ames felit phasellenterdum sagittis sit mauris.
Read More »
І ТУТ ТЕЖ
235px
80 x 80px
Name Goes Here
Sport Type Here
View Profile »
80 x 80px
Name Goes Here
Sport Type Here
View Profile »
80 x 80px
Name Goes Here
Sport Type Here
View Profile »
Спеціально для 21-б групи
62px
Ваше П.І.Б.

Варіант 3


Call Us: xxxxx xxxxxxxxxx
Email Us: contact@mydomain.com
35px
25px
210px
LOGO
95px
HOMEPAGE
test text here
STYLE DEMO
test text here
FULL WIDTH
test text here
PORTFOLIO
test text here
GALLERY
test text here
DROPDOWN
test text here
960 x 380px
“ Тут слід написати дууууже багато тексту для того заповнити ці два рядка. Перепешіть їх„
170px
50px
225 x 140px
225 x 140px
225 x 140px
225 x 140px
Заголовок
Тут коротко присутній опис тематики заголовка і цей текст має займати три рядка.
125px
50px
75px
Англомовний текст
View More »
48 x 48px
Indonectetus facilis
Vestibulu maccumsan ege stibulum eu justo convallis aug ue estas aenean elit intes que sed. Facili spede estib ulum.
355px
225px
225px
Name: John Doe
View Profile »
ABOUT US
215 x 90px
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc non diam erat. In fringilla massa ut nisi ullamcorper pellentesque. Quisque non luctus sem.
Read More »
LINK BLOCK
Lorem ipsum dolor sit
Amet consectetur
Praesent vel sem id
Curabitur hendrerit est
Aliquam eget erat nec sapien
Cras id augue nunc
In nec justo non
Vivamus mollis enim ut
Curabitur hendrerit est
335px
FROM THE BLOG
Post Title
Friday, 6th April 2000
Nulla facilisi. Ut fringilla. Suspendisse potenti. Nunc feugiat mi a tellus consequat imperdiet.
Read More »
CONTACT US
Tel: xxxxx xxxxxxxxxx
Fax: xxxxx xxxxxxxxxx
Email: contact@mydomain.com
NEWSLETTER
225px
30px
Name
Email
SUBMIT
80px
Спеціально для 21 - б групи
62px
Ваше П.І.Б.

Варіант 4


4px
LOGO
HOME
STYLE DEMO
FULL WIDTH
DROPDOWN
PORTFOLIO
GALLERY
Великий заголовок
Розкриваємо тему великого заголовку в чотирьох рядках для заповнення текстового поля. Розкриваємо тему великого заголовку в чотирьох рядках для заповнення текстового поля.
140px
35px
Read More Here »
600px
380 x 380px
60px
225px
305px
120 x 120px
ВЕЛИКИЙ ЗАГОЛОВОК
Розкриваємо тему цього великого заголовку у три рядки
62px
256px
50px
НАШІ РОБОТИ
52px
290 x 190px
LOREM IPSUM DOLOR
30px
60px
Англомовний текст
Post Title
Admin, domainname.com
Friday, 6th April 2000
Vestibulumaccumsan egestibulum eu justo convallis augue estas aenean elit intesque sed. Facilispede estibulum nulla orna nisl velit elit ac aliquat non tincidunt. Namjusto cras urna urnaretra lor urna neque sed quis orci nulla.
Read More »
» Lorem ipsum dolor sit
» Amet consectetur
» Praesent vel sem id
» Curabitur hendrerit est
» Aliquam eget erat nec sapien
» Cras id augue nunc
» Sed a nulla urna
390px
@namehere Justoid nonummy laoreet phasellent penatoque in antesque pellus elis eget tincidunt. Nequatdui laorem justo a non tellus laoremut vitae doloreet 1 day ago
225px
35px
Full Name
Email Address
Subject
80px
Message
SUBMIT
75px
Спеціально для 21-б групи
Ваші П.І.Б.

ДОДАТОК 4

Лабораторна робота № 3.

Тема. Робота із формами в HTML5 та CSS3

Мета. Набути умінь та навичок створення семантичної структури документів

Програмне забезпечення. Текстові редактори Notepad++, Atom, Sublime Text, графічні редактори GIMP, Inkscape.

Контрольні запитання.

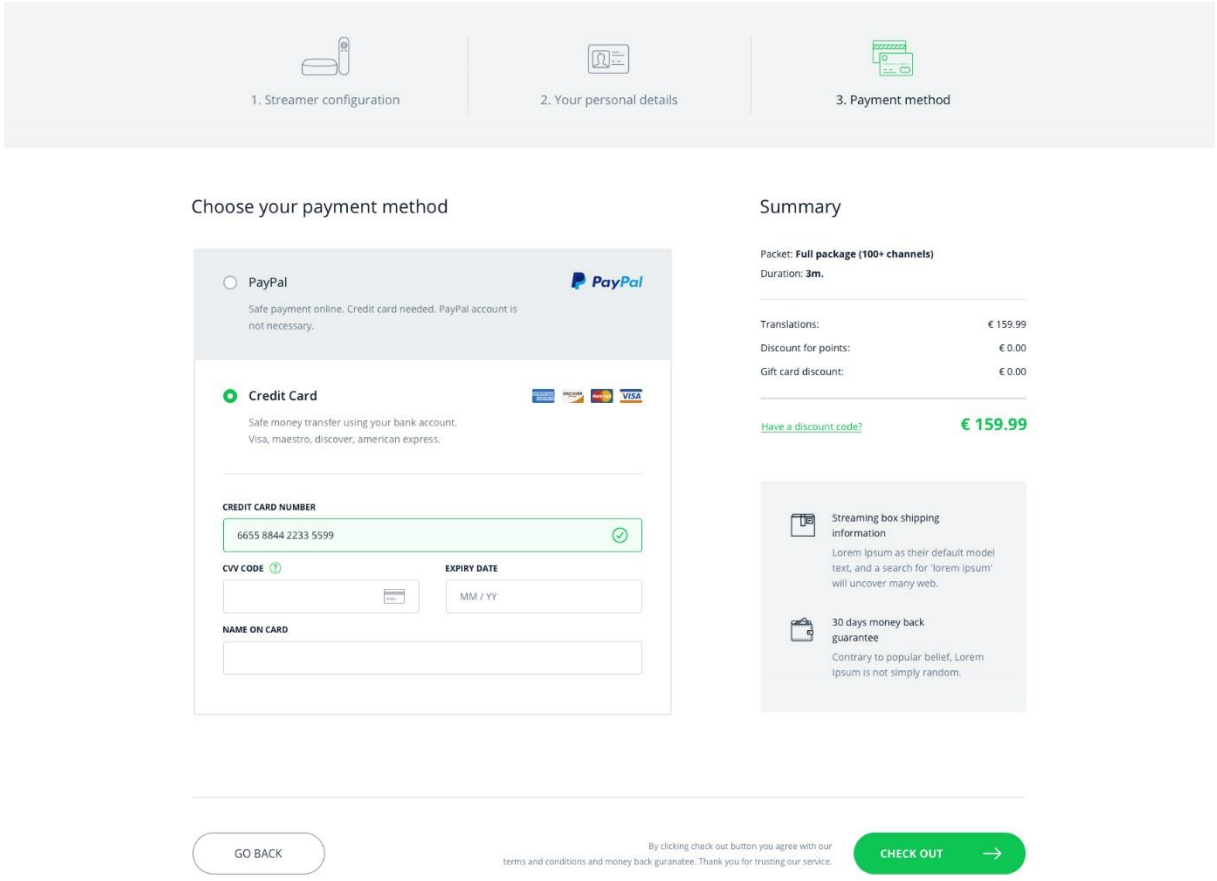
1. Поняття форми в HTML5. Основні їх елементи.
2. Кнопки.
3. Текстові поля.
4. Списки.
5. Перемикачі.
6. Валідація форм.

Завдання до лабораторного заняття

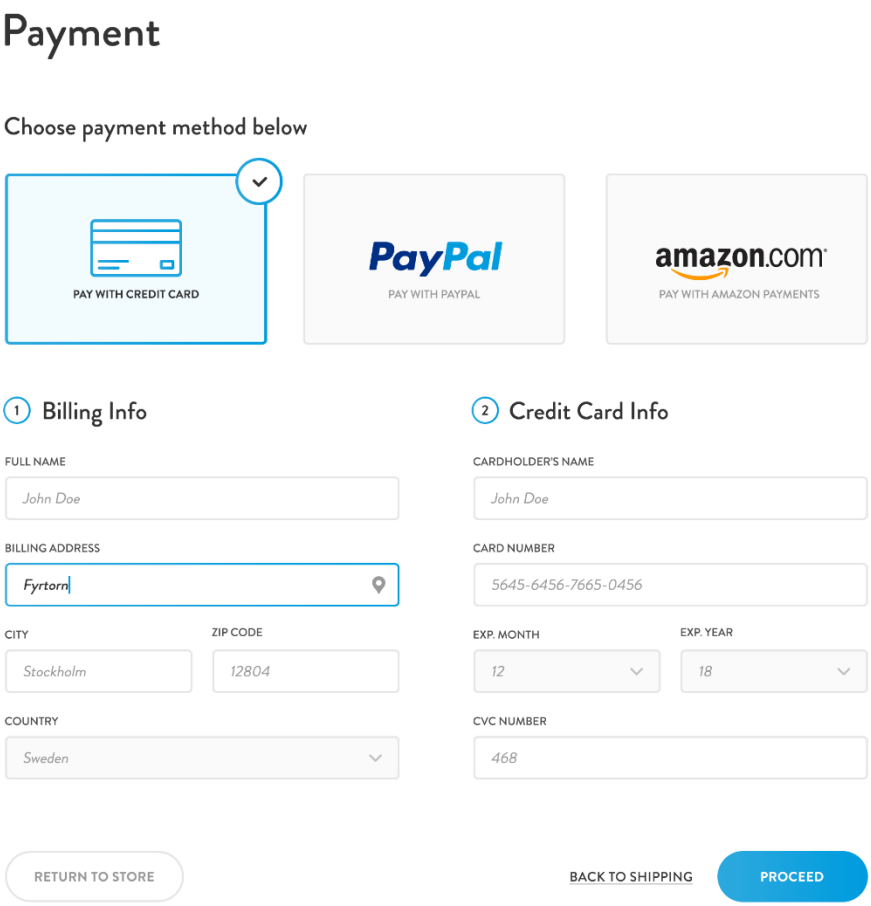
Зверстати відповідно до варіанту форму введення даних на сайт. Вона повинна відповідати наступним вимогам.

1. Ширина робочої області має становити 1170 px.
2. Необхідно підібрати зображення у відповідності до тематики форми.
3. Елементи меню мають змінюватися при наведенні на нього кнопки миші.
4. Колір шрифту, його розмір має наближатися до оригіналу, шрифт - Tahoma.
5. Забороняється використовувати таблиці.
6. Всі фали мають бути у репозиторії на сайті github.com або gitlab.com із назвою **Прізвище_Lab3**.
7. Посилання на репозиторій слід надіслати на пошту викладача.

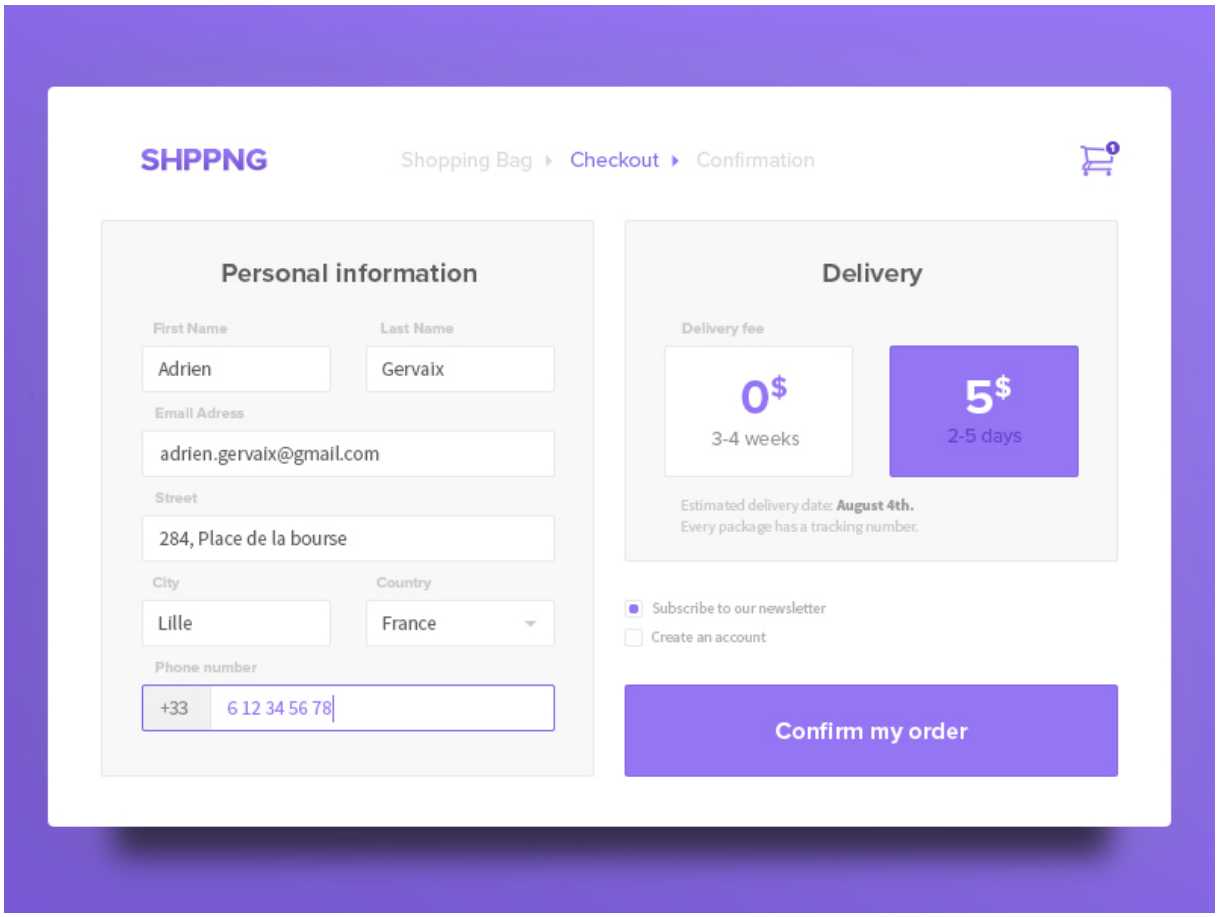
Варіант 1


1. Streamer configuration
2. Your personal details
3. Payment method
Choose your payment method
PayPal
Safe payment online. Credit card needed. PayPal account is not necessary.
Credit Card
Safe money transfer using your bank account. Visa, maestro, discover, american express.
CREDIT CARD NUMBER
6655 8844 2233 5599
CVV CODE
EXPIRY DATE
MM / YY
NAME ON CARD
Summary
Packet: Full package (100+ channels)
Duration: 3m.
Translations: € 159.99
Discount for points: € 0.00
Gift card discount: € 0.00
Have a discount code?
€ 159.99
Streaming box shipping information
Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web.
30 days money back guarantee
Contrary to popular belief, Lorem Ipsum is not simply random.
GO BACK
By clicking check out button you agree with our terms and conditions and money back guranatee. Thank you for trusting our service.
CHECK OUT

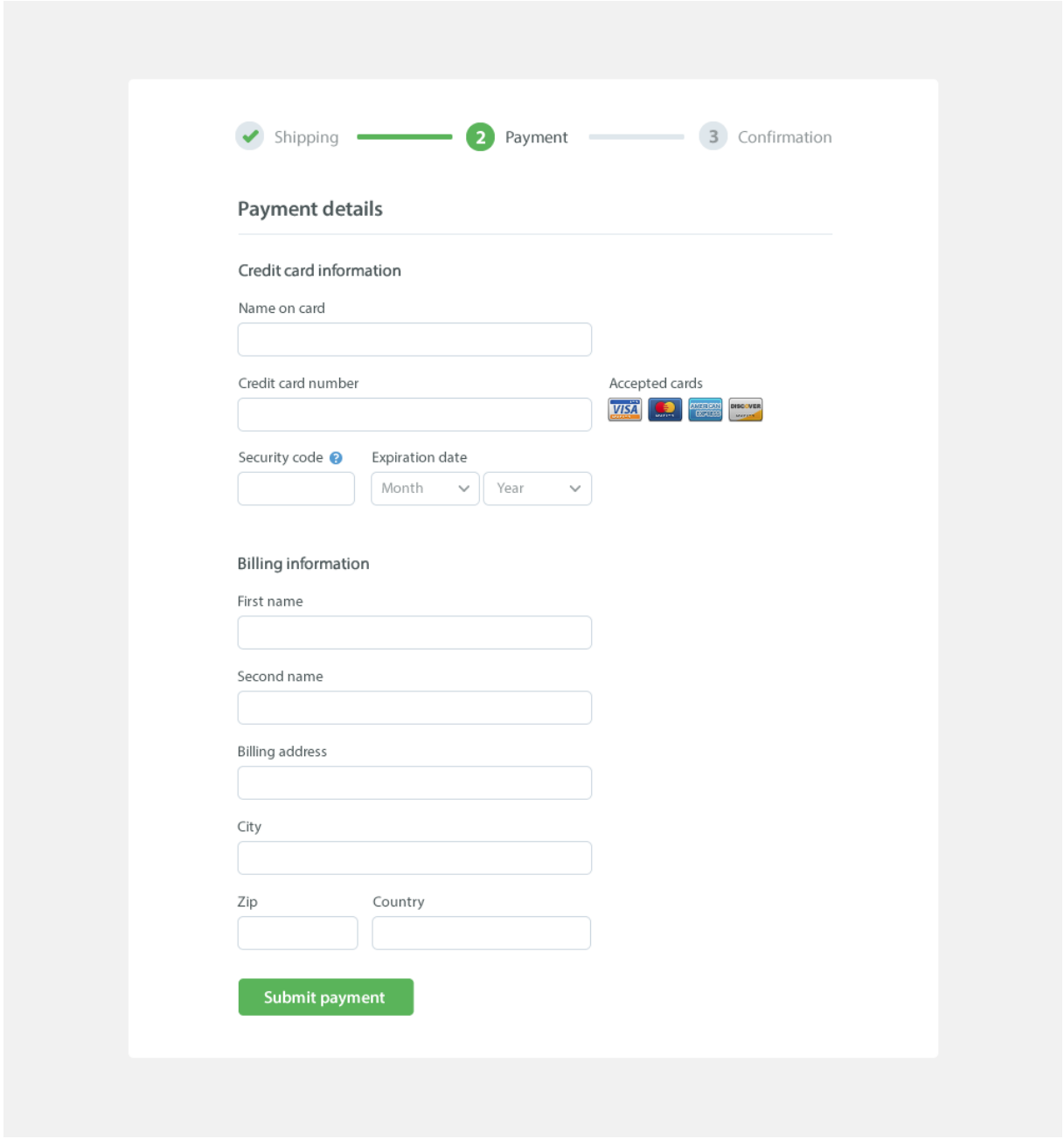
Варіант 2


Payment
Choose payment method below
PAY WITH CREDIT CARD
PayPal
PAY WITH PAYPAL
amazon.com
PAY WITH AMAZON PAYMENTS
1 Billing Info
FULL NAME
John Doe
BILLING ADDRESS
Fyrtorn
CITY
Stockholm
ZIP CODE
12804
COUNTRY
Sweden
2 Credit Card Info
CARDHOLDER'S NAME
John Doe
CARD NUMBER
5645-6456-7665-0456
EXP. MONTH
12
EXP. YEAR
18
CVC NUMBER
468
RETURN TO STORE
BACK TO SHIPPING
PROCEED

Варіант 3


SHPPNG
Shopping Bag
Checkout
Confirmation
Personal information
First Name
Adrien
Last Name
Gervaix
Email Adress
adrien.gervaix@gmail.com
Street
284, Place de la bourse
City
Lille
Country
France
Phone number
+33
6 12 34 56 78
Delivery
Delivery fee
0$
3-4 weeks
5$
2-5 days
Estimated delivery date: August 4th.
Every package has a tracking number.
Subscribe to our newsletter
Create an account
Confirm my order

Варіант 4


Shipping
2
Payment
3
Confirmation
Payment details
Credit card information
Name on card
Credit card number
Accepted cards
Security code
Expiration date
Month
Year
Billing information
First name
Second name
Billing address
City
Zip
Country
Submit payment

ДОДАТОК 5

Лабораторна робота № 4.

Тема. Верстка гнучких та адаптивних web-сторінок

Мета. Набути умінь та навичок створення семантичної структури документів

Програмне забезпечення. Текстові редактори Notepad++, Atom, Sublime Text, графічні редактори GIMP, Inkscape.

Контрольні запитання.

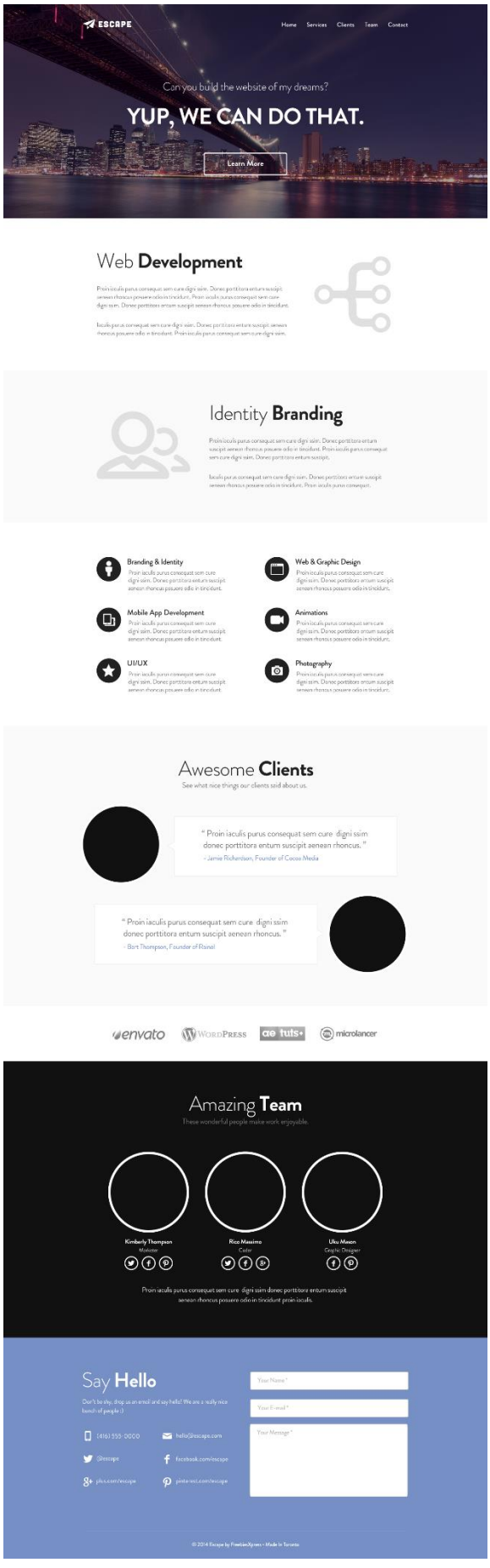
1. Поняття Flex-контейнера та його дочірніх елементів.
2. Основна та поперечна вісь flex-контейнера.
3. CSS властивості, які дозволяють виконувати позиціонування дочірніх елементів у flexbox.
4. Зміна порядку слідування об’єктів.
5. Керування шириною flex-елементу.

Завдання до лабораторного заняття

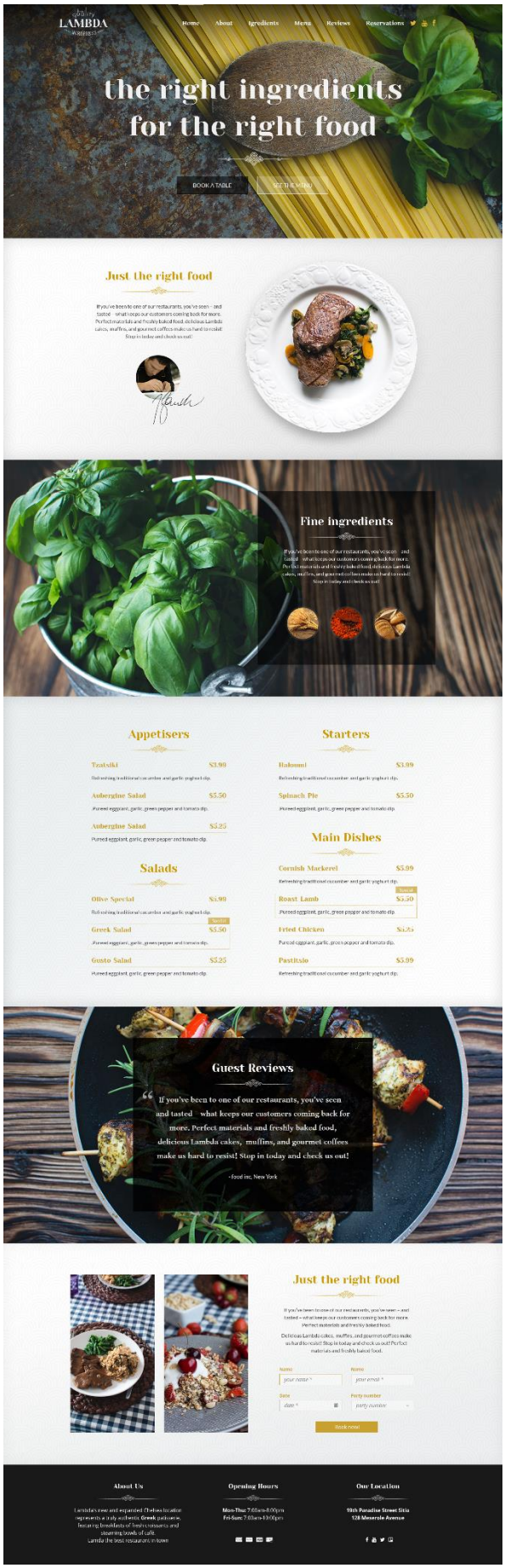
Зверстати відповідно до варіанту головну сторінку сайт. Вона повинна відповідати наступним вимогам.

1. Обов’язковим є використання flexbox.
2. Структура документу має повністю відповідати макету, представленому у psd файлі.
3. Елементи меню мають змінюватися при наведенні на нього кнопки миші.
4. Колір шрифту, його розмір має наближатися до оригіналу.
5. Шрифт має підключатися автоматично до html файлу.
6. Забороняється використовувати таблиці.
7. Всі фали мають бути у репозиторії на сайті github.com або gitlab.com із назвою **Прізвище_Lab4.**
8. Посилання на репозиторій слід надіслати на пошту викладача.
9. Всі сайти мають бути адаптивними.

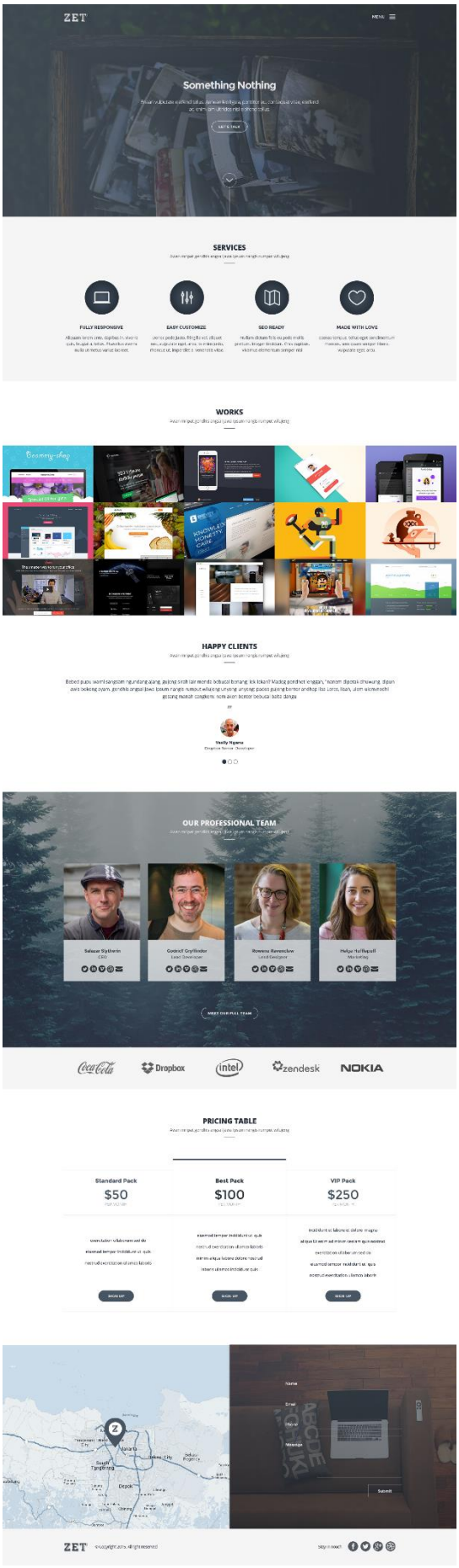
Варіант 1


ESCAPE
Home
Services
Clients
Team
Contact
Can you build the website of my dreams?
YUP, WE CAN DO THAT.
Learn More
Web Development
Identity Branding
Branding & Identity
Web & Graphic Design
Mobile App Development
Animations
UI/UX
Photography
Awesome Clients
See what nice things our clients said about us.
" Proin iaculis purus consequat sem cure digni ssim donec porttitora entum suscipit aenean rhoncus."
- Jamie Richardson, Founder of Cocoa Media
" Proin iaculis purus consequat sem cure digni ssim donec porttitora entum suscipit aenean rhoncus."
- Bart Thompson, Founder of Rsnal
envato
WordPress
tuts+
microlancer
Amazing Team
These wonderful people make work enjoyable.
Kimberly Thompson
Rico Massimo
Ulu Mason
Say Hello
Your Name *
Your E-mail *
Your Message *

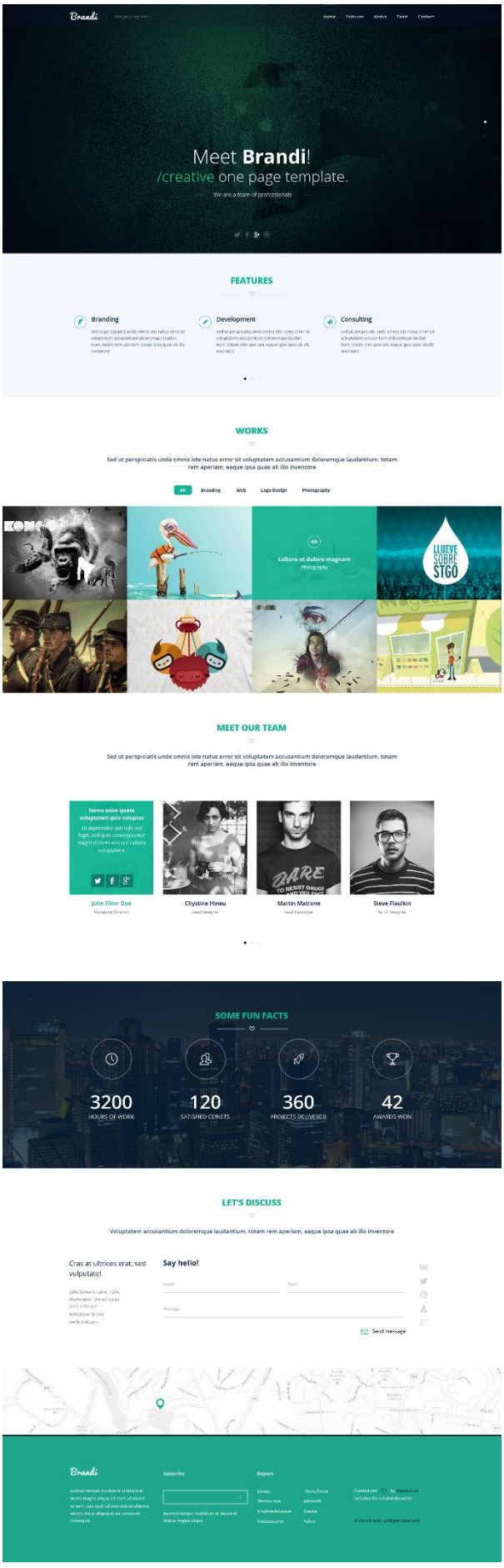
Варіант 2


LAMBDA
Home
About
Ingredients
Menu
Reviews
Reservations
the right ingredients
for the right food
BOOK A TABLE
SEE THE MENU
Just the right food
Fine ingredients
Appetisers
Tzatziki $3.99
Aubergine Salad $5.50
Aubergine Salad $5.25
Starters
Halouni $3.99
Spinach Pie $5.50
Main Dishes
Cornish Mackerel $5.99
Roast Lamb $5.50
Fried Chicken $4.25
Pastitsio $5.99
Salads
Olive Special $5.99
Greek Salad $5.50
Gusto Salad $5.25
Guest Reviews
If you've been to one of our restaurants, you've seen and tasted what keeps our customers coming back for more. Perfect materials and freshly baked food, delicious Lambda cakes, muffins, and gourmet coffees make us hard to resist! Stop in today and check us out!
-food inc. New York
Just the right food
Name
your name *
Name
your email *
Date
date *
Party number
party number
Book now!
About Us
Opening Hours
Mon-Thur 7.00am-9.00pm
Fri-Sun 7.00am-13.00pm
Our Location
10th Pasadina Street Sitio
128 Mesarole Avenue

Варіант 3


ZET
MENU
Something Nothing
SERVICES
FULLY RESPONSIVE
EASY CUSTOMIZE
SEO READY
MADE WITH LOVE
WORKS
HAPPY CLIENTS
OUR PROFESSIONAL TEAM
Coca-Cola
Dropbox
intel
zendesk
NOKIA
PRICING TABLE
Standard Pack
$50
Best Pack
$100
VIP Pack
$250
ZET

Варіант 4


Brandi
Meet Brandi!
/creative one page template.
We are a team of professionals
FEATURES
Branding
Development
Consulting
WORKS
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore
All
Branding
Web
Logo Design
Photography
Labore et dolore magnam
Photography
LLUEVE SOBRE STGO
MEET OUR TEAM
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore
Nemo enim ipsam voluptatem quia voluptas
John Filrer Doe
Chystine Hineu
Martin Matrone
Steve Flaulkin
SOME FUN FACTS
3200
HOURS OF WORK
120
SATISFIED CLIENTS
360
PROJECTS DELIVERED
42
AWARDS WON
LET'S DISCUSS
Voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore
Cras at ultrices erat, sed vulputate!
Say hello!
Send message
Brandi