

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ І ЗВ'ЯЗКУ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Данилець Є.В.

Методичні вказівки для
виконання лабораторних робіт з дисципліни

Веб-технології та веб-дизайн

Частина 1

Одеса – 2021

Рецензент: Т. О. Калініна, к.т.н., доцент кафедри будівельної механіки Одеської державної академії будівництва та архітектури.

Данилець Є.В. Методичні вказівки для виконання лабораторних робіт з дисципліни «Веб-технології та веб-дизайн». Частина 1. для студентів другого курсу спеціальності 121 – Інженерія програмного забезпечення / Є.В. Данилець. – Одеса: ДУІТЗ, 2021. – 77 с.

Методичні вказівки для виконання лабораторних робіт з дисципліни «Веб-технології та веб-дизайн» присвячені практичним аспектам проектування та побудови сучасних веб-сайтів. Рекомендації містять приклад практичного створення сучасного веб-сайту з використанням HTML5 та CSS3 . Зокрема, розглядаються такі питання як позиціонування елементів та блокова модель; типографіка та списки; додавання медіа-контенту та створення таблиць тощо. Методичні вказівки пропонуються студентам, які навчаються за спеціальністю 122 – Інженерія програмного забезпечення.

Ухвалено на засіданні кафедри інформаційних технологій і рекомендовано до друку.
(Протокол №6 від 22 січня 2021 року).

Затверджено методичною радою ДУІТЗ (протокол №1 від 11 березня 2021 року).

© Данилець Є.В., 2021.

© ДУІТЗ, 2021.

Зміст

Вступ.....	4
Тема 1. Створення першої Web-сторінки	5
Тема 2. Знайомство з HTML	7
Тема 3. Блокова модель.....	12
Тема 4. Позиціонування вмісту.....	15
Лабораторна робота № 1.....	21
Тема 5. Робота с типографікою.....	21
Тема 6. Встановлення фону і градієнту	31
Лабораторна робота № 2.....	37
Тема 7. Створення списків	37
Тема 8. Додавання медіа-контенту	43
Лабораторна робота № 3.....	50
Тема 9. Організація даних за допомогою таблиць	50
Лабораторна робота № 4.....	59
Тема 10. Побудова форм	60
Лабораторна робота № 5.....	66
Варіанти індивідуальних завдань.....	66
Література	77
Інформаційні ресурси.....	77

Вступ

Наша цивілізація нестримно прагне до комп'ютерної ери. У виробництві, керуванні, науці, торгівлі, банківській справі й в інших галузях широко застосовуються сучасні веб-технології. Не секрет, що ключем до оволодіння багатьма сучасними спеціальностями є вміння користуватися інтернет-технологіями.

Мета виконання даних практичних робіт полягає в засвоєнні основних принципів створення web-сайтів, опису і розмітки web-сторінок.

Кожну роботу слід виконувати під керівництвом викладача або самостійно. Завдання до робіт індивідуальні. Номером варіанта є його порядковий номер у списку студентів в журналі групи. Багато завдань потребують творчої активності виконавця, оскільки деякі умови завдань потрібно конкретизувати самостійно.

У складних випадках рекомендується звертатися до викладача або довідкових служб програм. Саме там, перегортаючи зміст довідок, можна знайти інформацію, теоретичні матеріали й приклади, що відсутні в книгах.

Успіхів!

Тема 1. Створення першої Web-сторінки

HTML

Щороку на кафедрі інформаційних технологій проводиться міжнародна науково-практична конференція молодих вчених і студентів. У цих методичних вказівках ми маємо організувати інформаційну підтримку конференції «Молодь у світі сучасних технологій» і створити для неї сайт протягом наступних занять.

Відкрийте текстовий редактор, створіть новий файл з ім'ям `index.html` і збережіть його в папці `conference` (в цій папці будуть знаходитися всі файли нашого майбутнього сайту) на робочому столі (або в іншому доступному для вас місці).

У файлі `index.html` додамо структуру документа, в тому числі тип документа `<!DOCTYPE html>` і елементи `<html>`, `<head>` і `<body>`.

```
<!DOCTYPE html>
<html lang="ua">
  <head>
  </head>
  <body>
  </body>
</html>
```

Всередину елемента `<head>` додамо елементи `<meta>` і `<title>`. Елемент `<meta>` повинен включати в себе правильні атрибути `charset` і значення, в той час як елемент `<title>` повинен містити назву сторінки, скажімо, Конференція "Молодь у світі сучасних технологій".

```
<head>
  <meta charset="utf-8">
  <title> Конференція "Молодь у світі сучасних технологій"</title>
</head>
```

Всередину елемента `<body>` додамо елементи `<h1>` і `<p>`. Елемент `<h1>` повинен включати в себе бажаний заголовок - Міжнародна конференція, а елемент `<p>` повинен включати в себе простий абзац для представлення нашої конференції.

```
<body>
  <h1> Міжнародна конференція</h1>
  <p>
    Шановні колеги! Запрошуємо вас взяти участь у V Міжнародній науково-
    практичній конференції молодих вчених та студентів «Молодь у світі сучасних
    технологій», за тематикою «Методологічні аспекти управління проектами
    розвитку регіону».
  </p>
</body>
```

Настав час подивитися на те, що було зроблено. Знайдемо файл `index.html`. Двічі клацнувши по цьому файлу або перетягнувши його в браузер, він відкриється для перегляду (рис. 1.1).

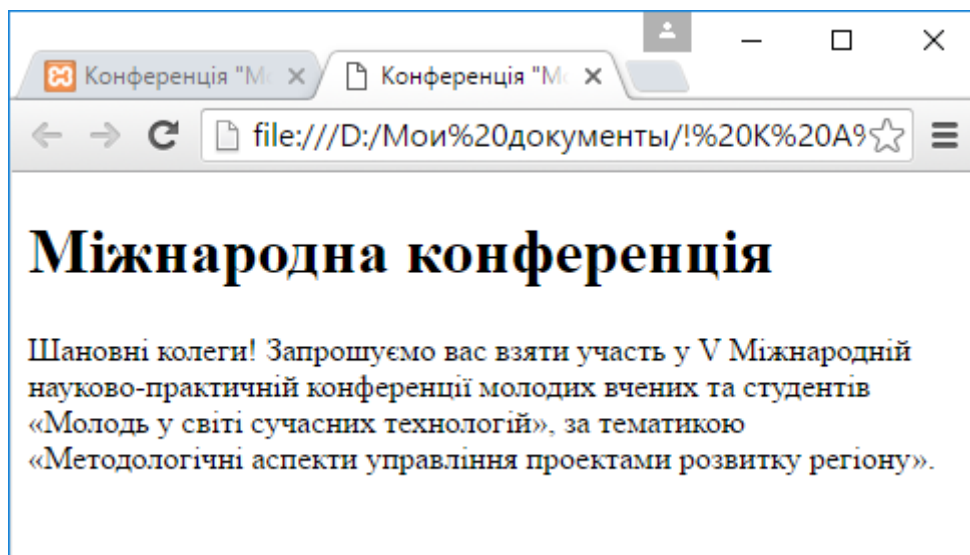


Рис. 1.1. Перша веб-сторінка

CSS

Усередині папки `conference` створимо нову папку з ім'ям `files`. У ній будуть зберігатися всі ресурси для веб-сайту, такі як стилі, зображення, відео тощо. Для стилів додамо ще одну папку `css` всередині папки `files`.

Використовуючи текстовий редактор створимо новий файл з ім'ям `main.css` і збережемо його в папці `css`, яка була щойно створена.

Переглядаючи файл `index.html` в браузері, ми можемо бачити, що елементи `<h1>` і `<p>` вже містять стиль за замовчуванням. Зокрема, у них встановлено унікальний розмір шрифту і простір навколо них. Використовуючи скидання Еріка Мейєра можна пом'якшити ці стилі, що дозволить кожному з них починати з однакової бази. Для цього перейдіть на його сайт (<https://meyerweb.com/eric/tools/css/reset/>), скопіюйте код і вставте його у верхню частину файлу `main.css`.

Файл `main.css` починає набувати форми, тому підключимо його до файлу `index.html`. Відкрийте `index.html` в текстовому редакторі і додайте елемент `<link>` в `<head>` відразу після елемента `<title>`.

Оскільки на стилі вказано через елемент `<link>`, додайте атрибут `rel` зі значенням `stylesheet`.

Також включимо посилання на файл `main.css` використовуючи атрибут `href`. Пам'ятайте, що файл `main.css` збережений в папці `css`, який знаходиться всередині папки `files`. Таким чином, значення атрибута `href`, який є шляхом до файлу `main.css`, має бути наступним: `files/css/main.css`.

```
<head>
  <meta charset="utf-8">
  <title>Конференція "Молодь у світі сучасних технологій"</title>
  <link rel="stylesheet" type="text/css" href="files/css/main.css">
</head>
```

Настав час для перевірки роботи і перегляду, як співіснують разом [HTML](#) і [CSS](#). Відкриття файлу [index.html](#) (або оновлення сторінки, якщо вона вже відкрита) в браузері має показати трохи інший результат, ніж раніше (рис. 1.2).

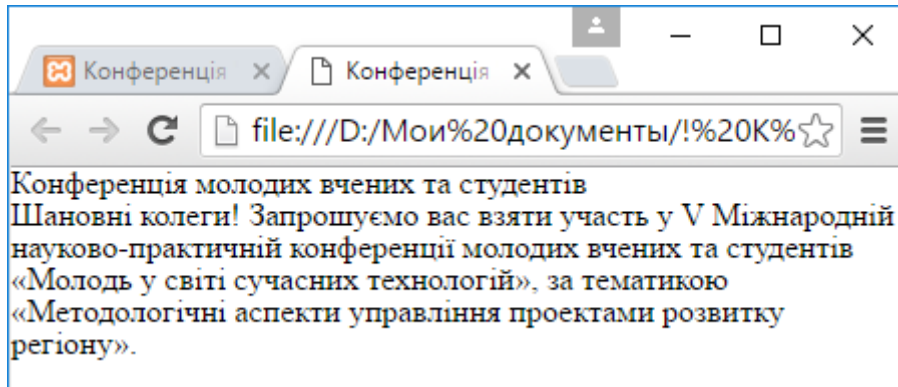


Рис. 1. 2. Перша веб-сторінка після використання скидання стилів

Тема 2. Знайомство з HTML

Створення структури сторінки

На даний момент сайту не вистачає реальної структури і вмісту для нього. Витратимо деякий час, щоб розширити вміст нашої головної сторінки.

Використовуючи існуючий файл [index.html](#), додамо в нього елемент `<header>`. Він повинен включати в себе існуючий елемент `<h1>`; також додамо елемент `<h3>` як слоган, щоб підтримати елемент `<h1>`.

```
<header>
  <h1> Міжнародна конференція </h1>
  <h3>2&ndash;3 червня &mdash; ХНТУ, Херсон, Україна</h3>
</header>
```

Після елементу `<header>` додамо нову групу, використовуючи елемент `<section>`, який презентує конференцію. Почнемо цей розділ з нового елемента `<h2>` і закінчимо абзацем.

```
<section>
  <h2> Шановні колеги!</h2>
  <p>
    Запрошуємо вас взяти участь у V Міжнародній науково-практичній
    конференції молодих вчених та студентів «Молодь у світі сучасних
    технологій», за тематикою «Методологічні аспекти управління
    проектами розвитку регіону».
  </p>
</section>
```

Після презентації конференції, включимо ще одну групу вмісту - кілька тизерів, зокрема, Тематика, Розклад і Місце проведення. Кожен із цих тизерів повинен також знаходитися у власному розділі і включати додатковий текст.

Згрупуємо всі тизери всередині елемента `<section>` і кожен окремий тизер буде також обгорнутий елементом `<section>`. В решті-решт, три елементи `<section>` будуть знаходитись всередині іншого `<section>`, це нормально.

```
<section>
  ...
  <section>
    <h5>Тематика</h5>
    <h3>Питання, що потребують негайного вирішення</h3>
    <p>Будуть розглянуті важливі питання, що стосуються методологічних
      аспектів управління проектами розвитку регіону.</p>
  </section>
  ...
</section>
```

На закінчення, додамо копірайт в елемент `<footer>` в кінці сторінки. Для цього використовуємо елемент `<small>`, який семантично представляє побічні коментарі і дрібний шрифт, який буде ідеально підходити для авторських прав.

Як правило, вміст всередині елемента `<small>` буде відображатися маленькими літерами, але наше скидання **CSS** цьому стане на заваді.

```
<footer>
  <small> &copy; кафедра ЕК та УП ХНТУ </small>
</footer>
```

Тепер головна сторінка, після додавання нового вмісту та структури, буде мати вигляд, що зображений на рис. 2.1.

Елемент `<h3>` всередині `<header>`, а також `<small>` всередині `<footer>` містить дещо цікаве, зокрема, кілька спеціальних символів.

Спеціальні символи включають у себе різні розділові знаки, букви з діакритичними знаками та символи. Коли вони містяться безпосередньо в **HTML**, то можуть бути неправильно зрозумілі або помилково прийняті за іншу букву, тому повинні бути кодовані.

Кожен спеціальний символ починається з амперсанда (`&`) і закінчується крапкою з комою. Все, що знаходиться між ними, є унікальним кодом символу, будь це ім'я або число.

Наприклад, можна було б кодувати слово «*résumé*» як «`resumé`». У нашому заголовку середнє і довге тире було закодовано, а в підвалі – символ авторського права. Для довідки, повний список кодів символів можна знайти, наприклад, тут <http://htmlweb.ru/html/symbols.php> або тут <http://frontender.com.ua/manual/html-symbols/>.

Головна сторінка починає складатися воедино, і тепер додамо декілька посилань, щоб створити додаткові сторінки і, тим самим, побудувати іншу частину нашого сайту.

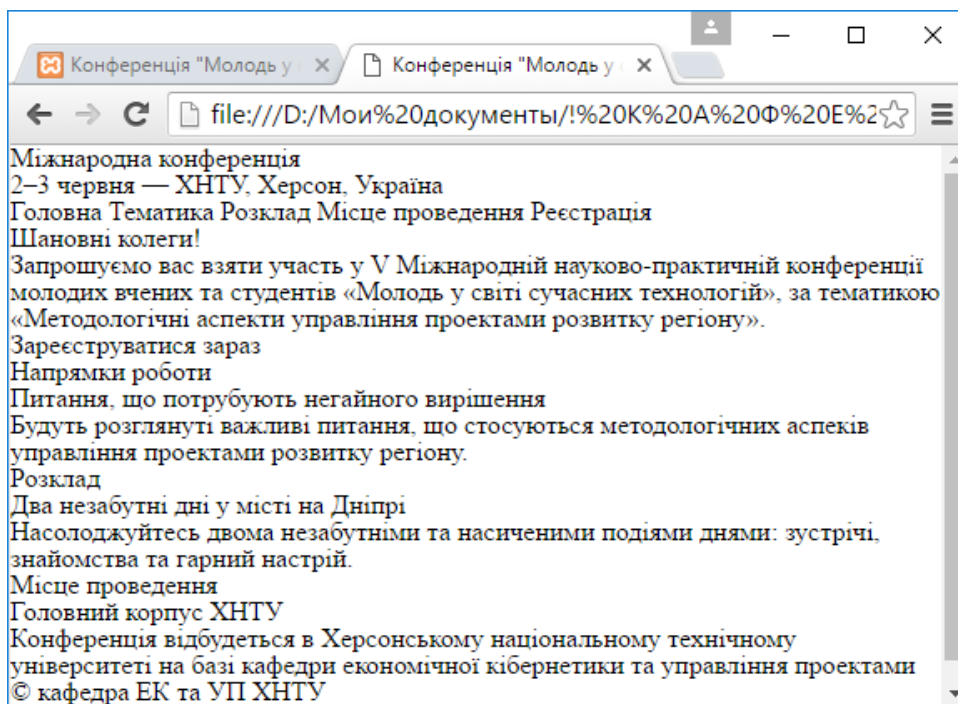


Рис. 2.1. Вигляд головної сторінки сайту, що має структуру

Додавання гіперпосилань

Прийшов час перетворити сайт конференції, що складається лише з однієї сторінки в повномасштабний сайт з декількома сторінками, де всі вони будуть пов'язані одна з одною за допомогою гіперпосилань.

Почнемо з перетворення тексту «Конференція молодих вчених та студентів» всередині `<h1>` в елемент `<header>` на посилання на сторінку `index.html`. Оскільки ми вже і так знаходимось на сторінці `index.html`, це може здатися трохи дивним, що справедливо, але завдяки тому, що заголовок повторюється на інших сторінках, зворотнє посилання на головну сторінку має сенс.

```
<h1>
    <a href="index.html"> Міжнародна конференція </a>
</h1>
```

Щоб рухатись по всіх інших сторінках додамо меню навігації за допомогою елемента `<nav>` всередину елемента `<header>`. Створимо сторінки Тематика, Розклад, Місце проведення та Реєстрація. Щоб переходити з головної сторінки на них, повинні бути додані відповідні посилання.

```
<header>
    ...
    <nav>
        <a href="index.html">Головна</a>
        <a href="direction.html">Напрямки роботи</a>
        <a href="schedule.html">Розклад</a>
        <a href="venue.html">Місце проведення</a>
        <a href="register.html">Реєстрація </a>
```

```

        </nav>
        ...
</header>

```

Також для зручності додамо це ж навігаційне меню з елемента `<header>` в елемент `<footer>`.

```

<footer>
    ...
    <nav>
        <a href="index.html">Головна</a>
        <a href="direction.html">Тематика</a>
        <a href="schedule.html">Розклад</a>
        <a href="venue.html">Місце проведення</a>
        <a href="register.html">Реєстрація</a>
    </nav>
    ...
</footer>

```

Усередині елемента `<section>`, який описує конференцію, трохи нижче заголовка необхідно також додати посилання для реєстрації на конференцію.

```

<section>
    ...
    <a href="register.html">Зареєструватися зараз</a>
</section>

```

Також треба додати посилання на інші сторінки в усі розділи з тизерами. Усередині кожного розділу, обгорнемо обидва елементи `<h3>` і `<h5>` посиланням, що веде на відповідну сторінку.

Необхідно переконатися, що це зроблено для кожного розділу.

```

<section>
    ...
    <a href="direction.html">
        <h5>Тематика</h5>
        <h3>Питання, що потребують негайного вирішення</h3>
    </a>
    <p>Будуть розглянуті важливі питання, що стосуються методологічних аспектів управління проектами розвитку регіону.</p>
    ...
</section>

```

Тепер потрібно створити кілька нових сторінок. Створимо файли `direction.html`, `schedule.html`, `venue.html` і `register.html`. Оскільки всі ці файли повинні розташовуватися в тій же папці, що і файл `index.html`, то всі наші посилання повинні працювати як і очікувалося.

Щоб гарантувати, що всі сторінки виглядають однаково, переконайтеся, що всі нові файли містять таку ж структуру документа що і файл `index.html`. Нижче представлений код

сторінки [direction.html](#), інші зробіть так само, а на рис. 2.2 зовнішній вигляд головної сторінки на даному етапі проектування веб-сайту конференції.

```
<!DOCTYPE html>
<html lang="ua">
<head>
  <meta charset="utf-8">
  <title>Конференція "Молодь у світі сучасних технологій"</title>
  <link rel="stylesheet" type="text/css" href="files/css/main.css">
</head>
<body>
  <header>
    <h1>
      <a href="index.html"> Конференція молодих вчених та
      студентів </a>
    </h1>
    <h3>2&ndash;3 червня &mdash; ХНТУ, Херсон, Україна</h3>
    <nav>
      <a href="index.html">Головна</a>
      <a href="direction.html">Тематика</a>
      <a href="schedule.html">Розклад</a>
      <a href="venue.html">Місце проведення</a>
      <a href="register.html">Реєстрація</a>
    </nav>
  </header>
  <section>
    <h1>Напрямки роботи</h1>
  </section>
  <footer>
    <small> &copy; кафедра ЕК та УП ХНТУ </small>
    <nav>
      <a href="index.html">Головна</a>
      <a href="direction.html">Тематика</a>
      <a href="schedule.html">Розклад</a>
      <a href="venue.html">Місце проведення</a>
      <a href="register.html">Реєстрація</a>
    </nav>
  </footer>
</body>
</html>
```

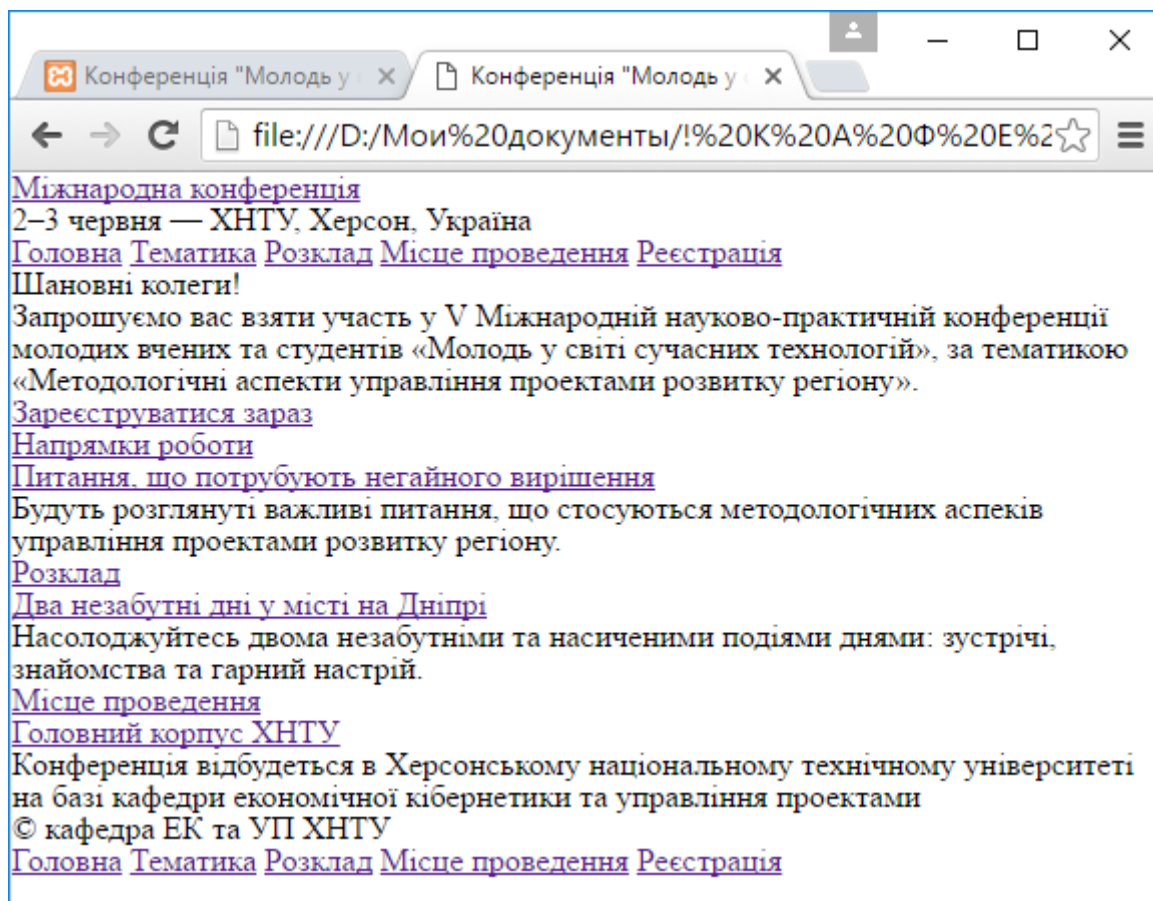


Рис. 2.2. Зовнішній вигляд головної сторінки з гіперпосиланнями

Тема 3. Блокова модель

Далі почнемо з перемикання моделі розміру блока з `content-box` у `border-box`, яка спростить установлення розмірів усіх елементів. У файлі `main.css`, трохи нижче скиду, додамо коментар, щоб визначити код для створення нашої сітки і макета сайту. Розташуємо все це нижче скиду, щоб цей код йшов в правильному порядку в каскаді.

Можна використовувати універсальний селектор `*`, поряд з універсальними псевдоелементами `*:before` і `*:after` для вибору всіх елементів і зміни `box-sizing` на `border-box`. Пам'ятайте, що необхідно включити вендорні префікси для властивості `box-sizing`, тому що це відносно нова властивість.

```
=====
Сітка
=====
*/
*,
*:before,
*:after {
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
```

```
box-sizing: border-box;
}
```

Далі створимо клас, який буде слугувати контейнером для наших елементів. Можна використовувати цей клас для різних елементів, щоб встановити загальну ширину, вирівняти елементи по центру сторінки і застосувати деякі загальні горизонтальні поля.

Трохи нижче набору універсальних селекторів створимо селектор класу `container`. У цьому селекторі встановимо значення властивості `width` як 1000 пікселів, `padding` зліва і справа по 30 пікселів, `margin` зверху і знизу як 0, а зліва і справа як `auto`.

Встановлення властивості `width` повідомляє браузеру яка ширина елемента з класом `container` однозначно повинна бути. Використання `margin` зліва і справа як `auto` в поєднанні з `width` дозволяє браузеру автоматично виставити рівні ліві і праві поля для елемента, таким чином, вирівняти його по центру на сторінці. Нарешті, `padding` зліва і справа гарантує, що наш вміст не розташований впритул до краю елемента і забезпечує невеликий простір між ними.

```
.container {
  margin: 0 auto;
  padding-left: 30px;
  padding-right: 30px;
  width: 1000px;
}
```

Тепер у нас є клас `container` доступний для використання і застосуємо цей клас в HTML коді до елементів `<header>` і `<footer>` на кожній сторінці, включаючи файли `index.html`, `speakers.html`, `schedule.html`, `venue.html` і `register.html`.

```
<header class="container">...</header>

<footer class="container">...</footer>
```

Тепер необхідно розташувати по центру решту вмісту наших сторінок. На головній сторінці (файл `index.html`) додамо клас `container` для `<section>` в розділ, який представляє конференцію і для розділу тизерів.

```
<section class="container">...</section>
```

Крім того, обернемо всі елементи `<h1>` на кожній сторінці елементом `<section>` з класом `container`.

```
<section class="container">
  <h1>...</h1>
</section>
```

Тепер, коли вміст вирівняно по центру, встановимо деякий вертикальний простір між елементами. Для початку розмістимо 22-піксельний нижній `margin` для деяких заголовків і абзаців. Додамо і прокоментуємо стилі нижче стилів сітки.

```
/*
=====
Типографіка
=====
*/

h1, h3, h4, h5, p {
  margin-bottom: 22px;
}
```

Заголовки `<h2>` і `<h6>` були пропущені навмисно, тому що по дизайну не вимагаються відступи для елементів `<h2>` і будь-які елементи `<h6>` не будуть використовуватись в даний момент.

Також створимо меж і деяких закруглених кутів. Почнемо з розміщення кнопки у верхній частині елемента `<section>` на головній сторінці, трохи нижче заголовка.

Раніше елемент `<a>` був доданий всередині `<section>`. Додамо класи `btn` і `btn-alt` до цих посилань.

```
<a class="btn btn-alt">...</a>
```

Тепер створимо деякі стилі для цих класів у нашому `CSS`. Нижче набору правил типографіки зробимо новий розділ для кнопок.

Для початку додамо клас `btn` і застосуємо деякі загальні стилі, які будуть використовуватися для всіх кнопок. Необхідно, щоб в усіх кнопок був `border-radius` 5 пікселів. Кнопки повинні відображатися як `inline-block`, тому можна додати поля з усіх чотирьох сторін і прибрати всі `margin`.

```
/*
=====
Кнопки
=====
*/

.btn {
  border-radius: 5px;
  display: inline-block;
  margin: 0;
}
```

Додамо стилі, характерні для кнопки реєстрації, за допомогою класу `btn-alt`. Сюди додамо однопіксельні рамки сірого кольору, `padding` 10 пікселів зверху і знизу і `padding` 30 пікселів зліва і праворуч від кнопки.

```
.btn-alt {
  border: 1px solid #dfe2e5;
  padding: 10px 30px;
}
```

Застосування двох класів `btn` і `btn-alt` до одного елемента `<a>` дозволяє накладати ці стилі, відображаючи їх для одного елемента.

Також додамо невеликий `padding` до елемента `<section>`, що містить наш елемент `<a>` з класами `btn` і `btn-alt`. Зробимо це шляхом додавання атрибута `class` зі значенням `hero` до елемента `<section>`, поряд з класом `container`, тому що це буде провідний розділ сайту.

```
<section class="hero container">
  ...
</section>
```

Далі створимо новий розділ у файлі `CSS` для стилів головної сторінки і після цього додамо клас `hero`, щоб встановити `padding` для всіх чотирьох сторін елемента `<section>`.

```
/*
=====
Головна сторінка
=====
*/

.hero {
  padding: 22px 80px 66px 80px;
}
```

Тема 4. Позиціонування вмісту

Створення обтікання в головній частині сторінки та в її підвалі

Перед застосуванням `float` до будь-якого елемента, необхідно забезпечити очистку обтічних елементів, додавши `clearfix` в наш `CSS`. У файлі `main.css`, трохи нижче стилів сітки, додамо `clearfix` з ім'ям класу `group`.

```
/*
=====
Clearfix
=====
*/

.group::before,
.group::after {
  content: "";
  display: table;
}
.group::after {
  clear: both;
}
.group {
  clear: both;
  *zoom: 1;
}
```

Тепер використаємо `float`, для основного `<h1>` всередині елемента `<header>` як `left` і таким чином дозволимо решті вмісту в заголовку обтікати його праворуч.

Для цього додамо клас `logo` до елементів `<h1>`. Потім всередині `CSS` додамо новий розділ стилів для основного заголовка. У цьому розділі виберемо елемент `<h1>` з класом `logo` і встановимо `float` як `left`.

HTML

```
<h1 class="logo">
  <a href="index.html">Styles Conference</a>
</h1>
```

CSS

```
/*
=====
Основний заголовок
=====
*/

.logo {
  float: left;
}
```

Додамо трохи більше деталей до нашого логотипа. Почнемо з розміщення елемента `<br>` для розриву рядка між словами «Міжнародна» і «Конференція», щоб змусити текст логотипа розташовуватися на двох рядках.

В `CSS` додамо границю зверху логотипа і невеликий вертикальний `padding`.

HTML

```
<h1 class="logo">
  <a href="index.html">Styles <br> Conference</a>
</h1>
```

CSS

```
.logo {
  border-top: 4px solid #648880;
  padding: 40px 0 22px 0;
  float: left;
}
```

Оскільки елемент `<h1>` став обтічним, то встановимо вміст `float`. Найближчим батьком для `<h1>` є елемент `<header>`, тому додамо до нього клас `group`. Це застосує до нього стилі `clearfix`, які були встановлені раніше.

```
<header class="container group">
  ...
</header>
```

Елемент `<header>` набуває форми, тому перейдемо до елементу `<footer>`. Подібно до того, що ми зробили з `<header>`, встановимо `float` для авторських прав як `left` усередині `<small>` і нехай всі інші елементи обтікають його праворуч.

Почнемо з додавання класу `primary-footer` до елементу `<footer>`. Оскільки ми знаємо, що у нас будуть обтічні елементи всередині `<footer>`, то повинні додати клас `group`.

```
<footer class="primary-footer container group">
  ...
</footer>
```

Тепер, коли клас `primary-footer` встановлений для елемента `<footer>`, можна використовувати цей клас, щоб цілеспрямовано вибрати елемент `<small>` за допомогою `CSS`. Необхідно задати йому `float` як `left`. Створимо новий розділ в файлі `main.css` для стилю основного підвалу.

```
/*
=====
Основний підвал
=====
*/

.primary-footer small {
  float: left;
}
```

Для перевірки - тут ми вибираємо елемент `<small>`, який повинен знаходитися всередині елемента з атрибутом `class` зі значенням `primary-footer`, в даному випадку це наш елемент `<footer>`.

Нарешті, додамо невеликий `padding` зверху і знизу елемента `<footer>`, це допоможе трохи відокремити його від іншої частини сторінки. Це можна зробити безпосередньо за допомогою класу `primary-footer`.

```
.primary-footer {
  padding-bottom: 44px;
  padding-top: 44px;
}
```

З урахуванням усіх цих змін в елементах `<header>` і `<footer>`, ми повинні внести їх на кожній сторінці, а не тільки на сторінці `index.html`.

Створення триколонкового макета

Для нашого сайту створимо триколонковий макет використовуючи рядково-блокові елементи. Зробимо це так, щоб отримати три колонки однакової ширини або дві колонки із загальною шириною, розділеної між ними як 2/3 для однієї і 1/3 для іншої (комбінація стилів 2/3 і 1/3 буде використана далі при створенні сторінки з тематикою конференції).

Для початку створимо класи, що визначають ширину цих колонок. Ці два класи назвемо `col-1-3` для однієї третини і `col-2-3` для двох третин відповідно. У розділі «Сітка» файлу `main.css` перейдемо вперед і визначимо ці класи і відповідну їм ширину.

```
.col-1-3 {
  width: 33.33%;
}
.col-2-3 {
  width: 66.66%;
}
```

Необхідно, щоб обидві колонки відображалися як рядково-блокові елементи. Також треба переконатися, що вертикальне вирівнювання у них задано по верхній частині кожної колонки.

Тепер створимо два нових селектори, які спільно використовуватимуть `display` і `vertical-align`.

```
.col-1-3,
.col-2-3 {
  display: inline-block;
  vertical-align: top;
}
```

Подивимось на `CSS` знову. Ми створили два селектори класу `col-1-3` і `col-2-3` розділених комою. Кома в кінці першого селектора означає, що за нею слідує інший селектор. Після другого селектора йде відкриваюча фігурна дужка, яка повідомляє що починається опис стилю. За допомогою коми, що розділяє селектори, можна прив'язати один стиль до кількох селекторів одночасно.

Далі необхідно встановити деякий простір між колонками, який допоможе розбити вміст. Це можна зробити, додавши горизонтальний `padding` до кожної колонки.

Це працює, проте, коли дві колонки розташовуються поряд одна з одною, ширина простору між ними буде вдвічі більше, ніж простір від зовнішнього краю. Щоб урівноважити це, помістимо всі колонки в сітку і додамо до неї аналогічний `padding`.

Скористаємося класом `grid`, щоб визначити сітку, а потім поставимо однаковий горизонтальний `padding` для класів `grid`, `col-1-3` і `col-2-3`. З комами, що розділяють селектори, `CSS` буде виглядати наступним чином:

```
.grid,
.col-1-3,
.col-2-3 {
  padding-left: 15px;
  padding-right: 15px;
}
```

Коли встановлюємо горизонтальний `padding`, потрібно бути обережними. Раніше було створено контейнер з класом `container` по центру всього контенту на сторінці з шириною 1000 пікселів. В даний момент, якщо б ми поставили елемент з класом `grid` всередину елемента з

класом `container`, то їх горизонтальні `padding` склалися б разом і наші колонки не відображалися б пропорційно ширині решти сторінки.

Щоб цього не сталося, необхідно використати деякі стилі із правил `container` з набором правил `grid`. Зокрема, необхідно використати властивість `width` (щоб переконатися, що наша сторінка залишається фіксованою на 1000 пікселів в ширину) і властивість `margin` (щоб вирівняти по центру сторінки елемент з класом `grid`).

Здійснимо це, розбиваючи набір старих правил `container` на наступне:

```
.container,
.grid {
  margin: 0 auto;
  width: 1000px;
}
.container {
  padding-left: 15px;
  padding-right: 150px;
}
```

Тепер будь-який елемент з класом `container` або `grid` буде 1000 пікселів у ширину і розташовуватися по центру сторінки. Крім того, був збережений існуючий горизонтальний `padding` для будь-якого елемента з класом `container` шляхом переміщення його в новий, окремий набір правил.

Вся складна і необхідна частина з налаштування сітки завершена. Тепер прийшов час для роботи з `HTML` і перегляду, як ці класи діють.

Почнемо з тизерів на головній сторінці, у файлі `index.html`, вирівнявши їх по трьох колонках. В даний час тизери обгорнуті елементом `<section>` з класом `container`. Замінімо клас `container` на `grid` так, щоб усередині можна було почати розміщення колонок.

```
<section class="grid">
  ...
</section>
```

Далі додамо клас `col-1-3` для кожного елемента `<section>` усередині `<section>` з класом `grid`.

```
<section class="grid">
  <section class="col-1-3">
    ...
  </section>

  <section class="col-1-3">
    ...
  </section>

  <section class="col-1-3">
    ...
  </section>
</section>
```

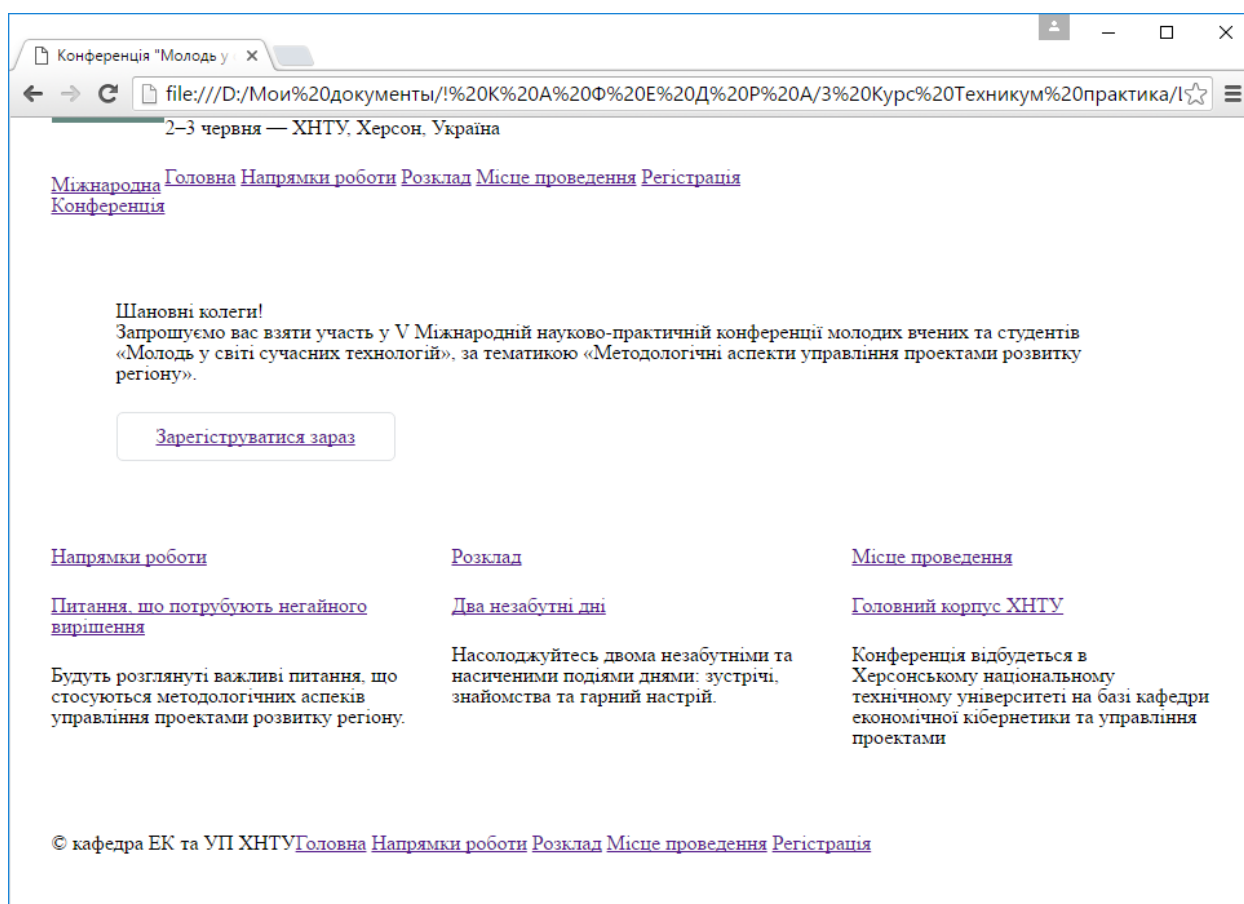


Рис 4.1. Головна сторінка сайту з позиційованим змістом

І, нарешті, оскільки кожна колонка є рядково-блоковим елементом, слід переконатися, що видалено порожній простір між ними. Щоб зробити це скористаємося коментарями і додамо трохи документації до кожного розділу, щоб краще організувати свій код.

```
<section class="grid"><!--
  Напрямки роботи
  --><section class="col-1-3">
  ...
</section><!--
  Розклад
  --><section class="col-1-3">
  ...
</section><!--
  Місце проведення
  --><section class="col-1-3">
  ...
</section>
</section>
```

Для перевірки: в рядку 3 ми залишили коментар, що ідентифікує розділ «Спікери». В кінці рядка 7 ми відкриваємо коментар відразу після закриваючого тегу `</section>`. Усередині цього коментаря, в рядку 9 ми визначаємо наступний розділ «Розклад». Потім закриваємо коментар на початку рядка 11, відразу перед відкриваючим тегом `<section>`. Аналогічна

структура коментарів з'являється в рядках з 13 до 17 між двома елементами `<section>`, прямо перед розділом «Місце проведення». В цілому, ми закоментували будь-який потенційний порожній простір між колонками, одночасно використовуючи ті ж коментарі для ідентифікації наших розділів.

Тепер у нас є повторно використовувана триколонкова сітка, яка підтримує різні розстановки, з використанням ширини колонок як 1/3 і 2/3. Головна сторінка тепер містить три колонки, розділяючи всі тизери (див. рис. 4.1).

Лабораторна робота № 1

Завдання

1. Створити `html` файл головної сторінки вашого сайту (візуальний шаблон див. у Варіантах індивідуальних завдань).
2. Створити таблицю стилів вашого сайту і підключити її до головної сторінки.
3. Створити основну структуру сайту.
4. Заповнити сайт текстовою інформацією. Малюнки, фон, градієнти та інший візуальний контент немає необхідності додавати на цій стадії, він буде доданий пізніше.
5. Створити область навігації (вгорі і внизу сторінки).
6. Додати ще кілька (не менше 3-х) сторінок на сайт.
7. Створити гіперпосилання для навігації між сторінками сайту.
8. Дайте властивості `box-sizing` значення `border-box`.
9. Встановіть необхідні розміри і відступи блоків вашого сайту відповідно до індивідуального шаблону.
10. Здійсніть позиціонування вмісту сайту, тобто створіть макет, як головної, так й інших сторінок сайту. При цьому в різних частинах сайту необхідно скористатися основними трьома технологіями: `inline-block`, `float` і з використанням `flexbox` (самостійне опрацювання останньої технології).

Тема 5. Робота с типографікою

Зміна властивостей шрифту

Настав час додати деякі властивості шрифту. Почнемо з поновлення шрифту для всього нашого тексту. Щоб зробити це, застосуємо стилі до елемента `<body>`. Почнемо з `color`, а також додамо значення `font-weight`, `font-size`, `line-height` і `font-family` через властивість `font`.

Щоб зберегти файл `main.css` організованим, наскільки це можливо, створимо новий розділ для цих користувальницьких стилів, помістивши його трохи нижче скидання і вище стилів сітки.

Нам потрібно додати наступне:

```

/*
=====
Стилі користувача
=====
*/
body {
  color: #888;
  font: 300 16px/22px "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
}

```

Раніше, в темі 4, було розпочато додавання деяких друкарських стилів, зокрема, нижній [margin](#) для заголовків різного рівня й абзаців. У цьому ж розділі файлу [main.css](#) додамо колір заголовків від першого до четвертого рівнів.

```

h1, h2, h3, h4 {
  color: #648880;
}

```

Також додамо розмір шрифту для цих заголовків. Елементи `<h1>` і `<h2>` використовуватимуть досить великі значення [font-size](#), отже, також необхідно збільшити у них значення [line-height](#), щоб зберегти текст усередині цих елементів розбірливим. Для довідки, ми встановимо значення [line-height](#) як 44 пікселя, подвоївши значення основного [line-height](#), встановленого в правилах для елемента `<body>`.

```

h1 {
  font-size: 36px;
  line-height: 44px;
}
h2 {
  font-size: 24px;
  line-height: 44px;
}
h3 {
  font-size: 21px;
}
h4 {
  font-size: 18px;
}

```

Елементи `<h5>` планується зробити більш унікальними, ніж інші заголовки. Відповідно, необхідно змінити їх стилі.

Застосуємо інше значення властивості [color](#) і трохи зменшимо [font-size](#) для цих елементів, а також змінимо [font-weight](#) на **400** або **normal**.

За замовчуванням, браузері відображають заголовки зі значенням [font-weight](#) як **bold**. Проте для наших заголовків в даний момент усі [font-weight](#) задані як **300**. Скидання **CSS** у верхній частині файлу `main.css` змінив [font-weight](#) на **normal**, а потім [font-weight](#) всередині елемента `<body>` задав значення **300** для всіх заголовків.

Значення `font-weight` як `400` для елемента `<h5>` насправді зробить це трохи товще, ніж інші заголовки і текст.

```
h5 {
  color: #a9b2b9;
  font-size: 14px;
  font-weight: 400;
}
```

Скидання на початку стилів також скине стилі браузера за замовчуванням для елементів `<strong>`, `<cite>` і `<em>`, які необхідно повернути. Для елементів `<strong>` необхідно встановити `font-weight` як `400`, який, насправді, прирівнюється до `normal`, а не `bold`, оскільки гарнітура, що використовується, товща, ніж більшість інших гарнітур. Потім для елементів `<cite>` і `<em>` необхідно встановити `font-style` як `italic`.

```
strong {
  font-weight: 400;
}
cite, em {
  font-style: italic;
}
```

Далі необхідно додати деякі стилі для посилань. На даний час вони сині, так встановлено в браузері за замовчуванням. Зробимо їх того ж кольору, що і заголовки від `<h1>` до `<h4>`. Крім того, можна застосувати псевдоклас `:hover`, щоб змінити колір на світло-сірий, коли користувач наводить курсор на посилання.

```
/*
=====
Посилання
=====
*/
a:hover {
  color: #a9b2b9;
}
a {
  color: #648880;
}
```

Тепер поглянемо на елемент `<header>` й оновимо для нього стилі. Почнемо з логотипа, додавши властивості `font-size` і `line-height` в набір правил для `logo`. Після додавання до існуючих властивостей `border-top`, `float` і `padding` новий набір правил повинен виглядати наступним чином:

```
.logo {
  border-top: 4px solid #648880;
  float: left;
  font-size: 48px;
  line-height: 44px;
```

```
padding: 40px 0 22px 0;
}
```

Оскільки ширину логотипа було збільшено, додамо `margin` до елемента `<h3>` всередині `<header>`, щоб це збалансувати. Зробимо це шляхом розміщення значення `tagline` атрибута `class` для елемента `<h3>`, а потім використовуючи цей клас в `CSS`, щоб застосувати відповідні поля. Важливо пам'ятати, що зміни в елементі `<h3>` повинні відбутися на кожній сторінці.

HTML

```
<h3 class="tagline">2 червня &mdash; ХНТУ, Херсон, Україна</h3>
```

CSS

```
.tagline {
  margin: 66px 0 22px 0;
}
```

Після елемента `<h3>` з класом `tagline` настає черга модифікувати елемент `<nav>`. Додамо значення `primary-nav` до атрибута `class` елемента `<nav>`, а також властивості `font-size` і `font-weight`, щоб навігація виділялася в порівнянні з іншою частиною заголовка.

HTML

```
<nav class="primary-nav">
  ...
</nav>
```

CSS

```
.primary-nav {
  font-size: 14px;
  font-weight: 400;
}
```

Оскільки тепер елемент `<header>` має набагато кращий вигляд, настав час поглянути на елемент `<footer>`. Використовуючи клас `primary-footer` змінимо `color` і `font-size` для всього тексту всередині `<footer>`. Крім того, піднімемо `font-weight` для `<small>` до 400.

З огляду на існуючі стилі, стиль для нашого підвалу повинен виглядати наступним чином:

```
.primary-footer {
  color: #648880;
  font-size: 14px;
  padding-bottom: 44px;
  padding-top: 44px;
}
.primary-footer small {
  float: left;
  font-weight: 400;
}
```

Також оновимо головну сторінку. Почнемо з розділу `hero` і збільшимо загальний `line-height` в розділі до 44 пікселів. Також зробимо текст у цьому розділі більше, збільшивши `font-size` для елемента `<h2>` до 36 пікселів і `font-size` для елемента `<p>` до 24 пікселів.

Можна внести всі ці зміни, використовуючи існуючий клас `hero` і створивши нові селектори для елементів `<h2>` і `<p>`. Наші стилі для розділу `hero` перероблюються тепер таким чином:

```
.hero {
  line-height: 44px;
  padding: 22px 80px 66px 80px;
}
.hero h2 {
  font-size: 36px;
}
.hero p {
  font-size: 24px;
}
```

На завершення, на головній сторінці з'явилась проблема, яку треба виправити. Раніше для всіх посилань був заданий світло-сірий колір при наведенні на них. Це прекрасно працює, за винятком трьох тизерів на головній сторінці, де посилання охоплює обидва елементи `<h3>` і `<h5>`. Оскільки у цих елементів є своє власне визначення `color`, псевдоклас `:hover`, описаний раніше, на них не впливає.

На щастя, це можна виправити, хоча для цього буде потрібно створити досить складний селектор. Почнемо з додавання класу `teaser` до всіх трьох колонок на головній сторінці. Будемо використовувати цей клас в якості відбіркового селектора.

```
<section class="grid"><!--
Напрямки роботи
--><section class=" teaser col-1-3">
<a href="direction.html">
  <h5>Напрямки роботи</h5>
  <h3>Питання, що потребують негайного вирішення</h3>
</a>
... ..
```

Почнемо створювати селектор з цього класу, оскільки необхідно націлитися тільки на елементи всередині елемента з класом `teaser`. Застосуємо стилі до елементів, які знаходяться всередині посилань і на які навели курсор миші. Таким чином, додамо селектор типу спільно з псевдокласом `:hover`. Нарешті, додамо селектор типу `h3`, щоб вибрати елементи `<h3>`, до яких ми хотіли б застосувати стилі.

В цілому, наш селектор і стилі для цих елементів `<h3>` будуть виглядати наступним чином:

```
.teaser a:hover h3 {
  color: #a9b2b9;
}
```

Застосування текстових властивостей

На даний момент кожне посилання на сторінці підкреслюється, що є стилем за замовчуванням для посилань, але його необхідно змінити.

Почнемо додавання до розділу з посиланнями у файлі `main.css` з видалення підкреслення для всіх посилань через властивість `text-decoration`. Далі виберемо всі посилання, які з'являються в абзацах, і задамо їм нижню межу.

Можна було б використовувати властивість `text-decoration` замість `border-bottom` для підкреслення посилань всередині кожного абзацу, однак за допомогою властивості `border-bottom` можна отримати більше контролю над зовнішнім виглядом підкреслення. Тут, наприклад, підкреслення буде іншого кольору, ніж сам текст.

Наш розділ з посиланнями, який включає в себе попередні стилі при наведенні, тепер повинен виглядати наступним чином:

```
a {
  color: #648880;
  text-decoration: none;
}
a:hover {
  color: #a9b2b9;
}
p a {
  border-bottom: 1px solid #dfe2e5;
}
```

Повертаючись до елементів `<h5>`, у яких стиль трохи відрізняється від інших заголовків, встановимо їх у верхній регістр за допомогою властивості `text-transform`. Новий набір стилів елементів `<h5>` повинен виглядати наступним чином:

```
h5 {
  color: #a9b2b9;
  font-size: 14px;
  font-weight: 400;
  text-transform: uppercase;
}
```

Повернемося до елементу `<header>` і застосуємо додаткові стилі до навігаційного меню (до якого раніше був доданий клас `primary-nav`). Після існуючих властивостей `font-size` та `font-weight` додамо `letter-spacing` і змінимо текст за допомогою властивості `text-transform`.

Стилі для елемента `<nav>` з класом `primary-nav` тепер повинні виглядати наступним чином:

```
.primary-nav {
  font-size: 14px;
  font-weight: 400;
  letter-spacing: .5px;
  text-transform: uppercase;
}
```

Раніше для логотипу був заданий `float` як `left` всередині елемента `<header>`. Тепер слоган знаходиться безпосередньо праворуч від логотипа, проте, необхідно, щоб він був видимим в правій частині елемента `<header>`.

Для цього потрібно додати властивість `text-align` зі значенням `right` для елемента `<h3>` з класом `tagline`, щоб отримати слоган, який розташовується праворуч.

При додаванні до існуючої властивості `margin` нові стилі для елемента `<h3>` з класом `tagline` будуть виглядати наступним чином:

```
.tagline {
  margin: 66px 0 22px 0;
  text-align: right;
}
```

Також необхідно, щоб навігаційні меню в елементах `<header>` і `<footer>` розташовувалися по правому краю. Оскільки обидва елементи `<header>` і `<footer>` містять дочірні обтічні елементи, можна використати той самий підхід, що і для слогана.

Обтічні елементи всередині `<header>` і `<footer>` прибираються зі звичайного потоку сторінки, і це змушує інші елементи обтікати їх. В даному конкретному випадку навігаційні меню – це елементи, які обтікають обтічні елементи.

Оскільки застосовуються одні і ті ж самі стилі відразу до двох навігаційних меню, то дамо кожному клас `nav`. Елемент `<header>` буде виглядати наступним чином:

```
<header class="container group">
  <h1 class="logo">...</h1>
  <h3 class="tagline">...</h3>
  <nav class="nav primary-nav">
    ...
  </nav>
</header>
```

А елемент `<footer>` буде виглядати так:

```
<footer class="primary-footer container group">
  <small>...</small>
  <nav class="nav">
    ...
  </nav>
</footer>
```

Необхідно пам'ятати, що зміни в наших елементах `<header>` і `<footer>` повинні бути виконані на кожній сторінці.

З класом `nav` для двох навігаційних меню створимо новий розділ у рамках файлу `main.css`, щоб додати загальні стилі навігації. Почнемо з додавання властивості `text-align` зі значенням `right` до класу `nav`. Розширимо ці стилі згодом, але поки це послужить основою.

```
/*
```

```

=====
Навігація
=====
*/
.nav {
  text-align: right;
}

```

Властивість `text-align` було вже додано до кількох різних елементів, також додамо ще `text-align` зі значенням `center` до набору правил з класом `hero`. Для довідки, ці стилі, в тому числі, що містять властивості `line-height` і `padding`, знаходяться в розділі головної сторінки нашого файлу `main.css`.

```

.hero {
  line-height: 44px;
  padding: 22px 80px 66px 80px;
  text-align: center;
}

```

Сайт конференції тепер має серйозний стиль – діловий стиль: всі стилі разом поєднуються гармонічно.

Вбудовування веб-шрифтів

Щоб додати трохи характеру до сайту конференції, спробуємо використати [Google Font](#). Зайдемо на сайт [Google Fonts](#) і пошукаємо шрифт [Roboto](#). Після того як його буде знайдено, додамо цей шрифт до колекції, згідно з інструкціями на сайті [Google Font](#).

Коли прийде час вибору, яке накреслення ми хотіли б використовувати, переконаємося, що вказали [300](#) і [400](#), які вже були застосовувані. Також додамо накреслення [100](#) до колекції теж – воно згодом стане у пригоді.

[Google Font](#) згенерує додатковий елемент `<link>` для включення в `<head>` на всіх сторінках. Розташуємо цей елемент безпосередньо нижче існуючого `<link>`. Новий елемент буде включати в себе правильний покажчик на стилі [Google](#), який в тому числі включає новий файл CSS з правилом `@font-face`, необхідним для використання шрифту [Roboto](#).

Після додавання нового елемента `<link>`, елемент `<head>` буде виглядати наступним чином:

```

<head>
<meta charset="utf-8">
<title>Конференція "Молодь у світі сучасних технологій"</title>
<link rel="stylesheet" type="text/css" href="files/css/main.css">
<link href="https://fonts.googleapis.com/css?family=Roboto:100,300,400&subset=cyrillic"
rel="stylesheet">
</head>

```

Після того, як було додано новий елемент `<link>` для всіх сторінок можна використовувати шрифт [Roboto](#). Це можна зробити, додавши його в основний набір шрифтів у властивості `font` всередині стилю для елемента `<body>`.

Додамо **Roboto** на початку набору шрифтів так: "**Roboto**", "**Open Sans**", "**Helvetica Neue**", **Helvetica**, **Arial**, **sans-serif**.

Хоча **Roboto** це одне слово, але оскільки це вбудований веб-шрифт, то візьмемо його в лапки всередині CSS. Наш новий стиль для елемента `<body>` буде виглядати наступним чином:

```
body {
  color: #888;
  font: 300 16px/22px "Roboto", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
}
```

Тепер **Roboto** повинен завантажитися і відображатися по всьому тексту на сайті. Оновимо трохи логотип сайту. Всередині набору правил з класом **logo** почнемо з додавання властивості **font-weight** зі значенням **100**, щоб зробити текст досить тонким. Також використовуємо властивість **text-transform** зі значенням **uppercase**, щоб зробити всі букви великими і властивість **letter-spacing** зі значенням **.5** пікселів, щоб додати трохи простору між буквами всередині логотипа.

Всі стилі для нашого логотипа будуть виглядати наступним чином:

```
.logo {
  float: left;
  font-size: 48px;
  font-weight: 100;
  letter-spacing: .5px;
  line-height: 48px;
  border-top: 4px solid #648880;
  padding: 40px 0 22px 0;
  text-transform: uppercase;
}
```

Встановимо **font-weight** значенням **100** для абзацу розділу **hero**. Можна використовувати існуючий селектор щоб зробити це і оновлений набір правил буде виглядати наступним чином:

```
.hero p {
  font-size: 24px;
  font-weight: 100;
}
```

При автономній роботі (відсутність інтернету) можна завчасно завантажити файли шрифту з сайту **Google**. Файли завантажуються у вигляді **zip** архіву. Цей архів буде потрібно розпакувати в директорію для шрифтів (за аналогією зі стилями): `\files\fonts`, яку необхідно буде завчасно створити. Далі в файлі **main.css** в блоці призначених для користувача стилів необхідно додати підключення трьох файлів шрифтів для жирності **100**, **300** і **400**:

```
@font-face {
  font-family: "Roboto";
  src: url("../fonts/roboto/Roboto-Thin.ttf");
  font-weight: 100;
```

```

}
@font-face {
  font-family: "Roboto";
  src: url("../fonts/roboto/Roboto-Light.ttf");
  font-weight: 300;
}
@font-face {
  font-family: "Roboto";
  src: url("../fonts/roboto/Roboto-Regular.ttf");
  font-weight: 400;
}

```

Після стилізації сайту конференції за допомогою шрифтів, він буде виглядати як показано на рис. 5.1.

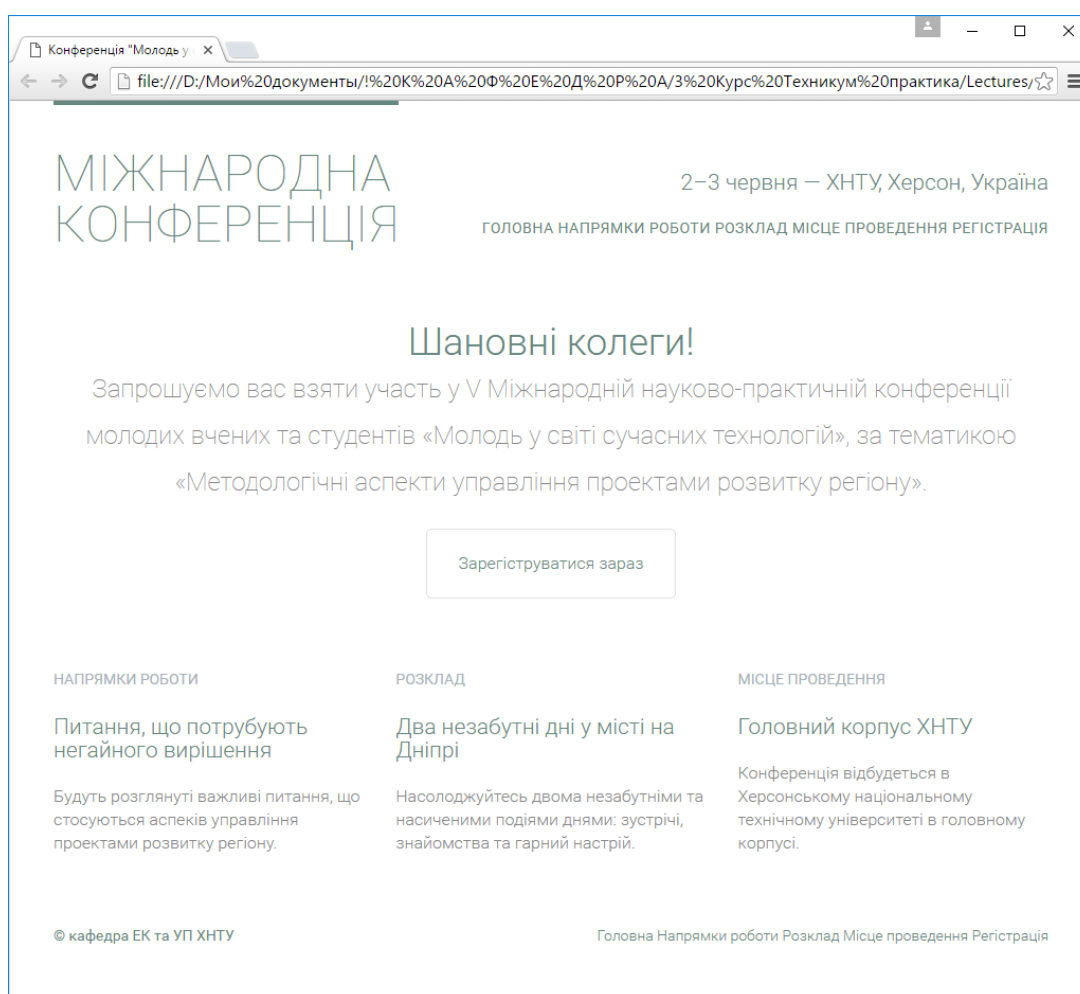


Рис. 5.1. Гловна сторінка сайту конференції

Тема 6. Встановлення фону і градієнту

Додавання фону

Настав час додати трохи фонового кольору. Почнемо із застосування синього фону до елемента `<body>` поряд з наявними властивостями `color` і `font`. Всі стильові правила для елемента `<body>`, встановлені на даний момент, включають наступне:

```
body {
  background: #293f50;
  color: #888;
  font: 300 16px/22px "Roboto", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
}
```

Синій фон для елемента `<body>` був встановлений навмисно, тому що сайт буде містити кілька різних рядів `background`, а найбільш частим кольором фону буде синій.

Тепер, коли кожна сторінка на включає в себе синій фон, очистимо кілька областей з ним. Зокрема, елементи `<header>` і `<footer>` залишаються синіми, як і розділ `hero` на головній сторінці.

Усередині елементів `<header>` і `<footer>` встановимо колір посилань білим, а потім при наведенні на них курсору миші тим же зеленим, що і заголовки.

Почнемо з елемента `<header>`. Щоб вибрати всі елементи `<a>` всередині елемента `<header>` додамо клас `primary-header` до `<header>` (на додаток до існуючих класів `container` і `group`). Необхідно пам'ятати, що необхідно додати цей клас до елементів `<header>` на всіх сторінках.

```
<header class="primary-header container group">
  ...
</header>
```

З класом `primary-header` у елемента `<header>` й існуючим класом `primary-footer` у `<footer>` можна додати два нових правила нижче розділу з посиланнями у файлі `main.css`.

Перший набір правил буде вибирати всі елементи `<a>` всередині елемента з класом `primary-header` або `primary-footer` і встановить їх колір білий, як це визначено комою, що розділяє два окремих селектори, в яких одна і та ж властивість і значення. Другий набір правил буде вибирати ті ж елементи `<a>` що і раніше, але вони змінять колір на зелений, коли користувач наводить на них покажчик миші.

```
.primary-header a,
.primary-footer a {
  color: #fff;
}
.primary-header a:hover,
.primary-footer a:hover {
  color: #648880;
}
```

Тепер встановимо колір тексту в розділі `hero` на головній сторінці теж білим, тому що він буде залишатися на синьому тлі. У нас є існуючий клас `hero`, до якого можна додати стилі, так що встановимо білий колір тексту в ньому. В цілому, набір правил для класу `hero` повинен включати в себе наступне:

```
.hero {
  color: #fff;
  line-height: 44px;
  padding: 22px 80px 66px 80px;
  text-align: center;
}
```

Також в розділі `hero` головної сторінки налаштуємо деякі стилі кнопок. Почнемо з додавання кількох нових властивостей для набору правил з класом `btn`, в розділі з кнопками нашого файлу `main.css`.

Зокрема, замінимо колір тексту кнопки на білий, задамо `cursor` як `pointer`, збільшимо `font-weight`, додамо невелику відстань у `letter-spacing` і змінимо `text-transform` на `uppercase`.

В цілому, новий набір правил для класу `btn` повинен виглядати наступним чином:

```
.btn {
  border-radius: 5px;
  color: #fff;
  cursor: pointer;
  display: inline-block;
  font-weight: 400;
  letter-spacing: .5px;
  margin: 0;
  text-transform: uppercase;
}
```

Також очистимо деякі стилі інших кнопок шляхом установки класу `btn-alt`. Зокрема, зробимо рамку для кнопки білою і додамо стиль для `:hover`, який включає білий фон і синій колір тексту.

З усіма доповненнями наш новий клас `btn-alt` повинен виглядати наступним чином:

```
.btn-alt {
  border: 1px solid #fff;
  padding: 10px 30px;
}
.btn-alt:hover {
  background: #fff;
  color: #648880;
}
```

Тепер додамо стилі для смуг з білим фоном. Вставимо новий розділ всередині файлу `main.css` для смуг, трохи нижче розділу `clearfix`. В цьому розділі створимо новий селектор класу з ім'ям `row`.

У новому класі `row` додамо білий фон, мінімальну ширину `960` пікселів (щоб переконатися, що `row` завжди більше, ніж `width` у елементів `container` або `grid`) і `padding` по вертикалі. В цілому новий розділ в файлі `main.css` повинен виглядати наступним чином:

```
/*
=====
Смуги
=====
*/

.row {
  background: #fff;
  min-width: 1000px;
  padding: 66px 0 44px 0;
}
```

З класом `row` додамо смугу з білим фоном на головній сторінці. Зробимо це для розділів з тизерами. В даний час ця область містить елемент `<section>` з класом `grid` який обгортали три додаткових `<section>` з класами `teaser` і `col-1-3`.

Щоб додати білий `background` до цього розділу, необхідно обернути всі ці елементи в елемент з класом `row`.

Оскільки необхідно обернути розділи з усіма тизерами елементом `<section>`, то додамо новий елемент `<section>` з класом `row`, який оточує існуючий `<section>` з класом `grid`.

Наявність двох елементів `<section>`, які обертають один вміст, знижує семантичний вплив. Щоб виправити це, змінимо другий елемент `<section>` з класом `grid` на елемент `<div>`. Зрештою, цей елемент тільки додає стилі, а не смислове значення, тому підходить елемент `<div>`.

Структура тизерів повинна виглядати наступним чином:

```
<section class="row">
  <div class="grid"><!--

    Напрямки роботи

  --><section class="teaser col-1-3">
    ...
  </section><!--

    Розклад

  --><section class="teaser col-1-3">
    ...
  </section><!--

    Місце проведення

  --><section class="teaser col-1-3">
    ...
  </section>
```

```
</div>
</section>
```

Дивно, як кілька фонових кольорів можуть впливати на дизайн веб-сайту. Наш сайт стає все кращим і головна сторінка тому доказ (рис. 6.1).

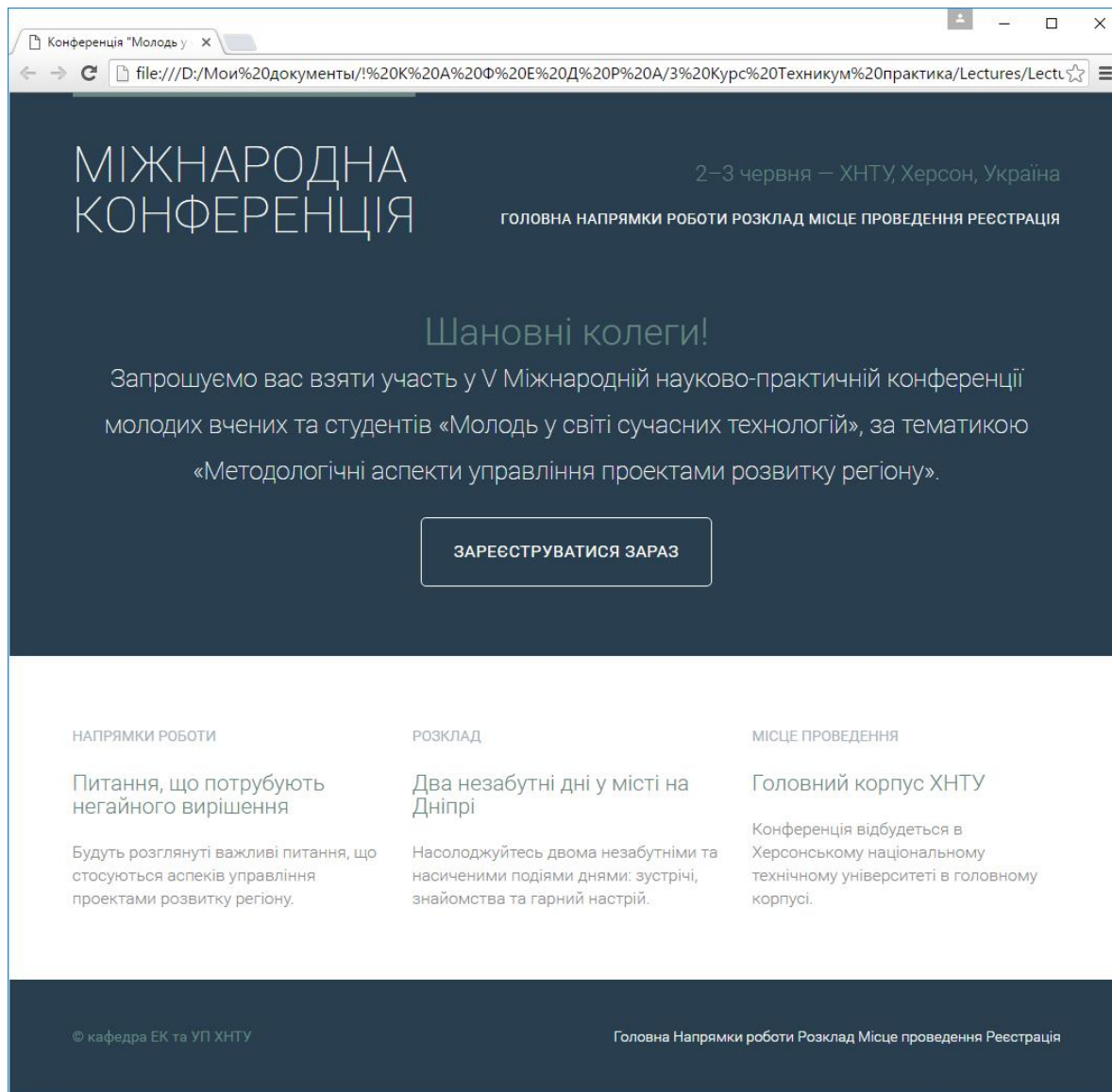


Рис. 6.1. Головна сторінка сайту після додавання декількох кольорів фону

Додавання градієнта

За допомогою градієнта створимо нову смугу для сайту конференції. Для цього використаємо клас `row-alt`. Оскільки для нової смуги задано ту саму властивість і значення `min-width` що і для класу `row`, об'єднаємо ці два селектори.

```
.row,
.row-alt {
  min-width: 1000px;
}
```

Далі створимо новий набір правил і застосуємо стилі спеціально до класу `row-alt`. Ці нові стилі включатимуть в себе градієнтний `background`, який починається з зеленого кольору і переходить до жовтого, зліва направо.

Використовуючи функцію `linear-gradient()` з відповідними значеннями і вендорними префіксами, додамо градієнтний `background` для класу `row-alt`. Також включимо один `background` перед градієнтним `background` як запасний варіант, на той випадок, якщо браузер не підтримує градієнти.

Нарешті, також додамо і деякий вертикальний `padding`. Оновлений розділ тепер виглядає наступним чином:

```
.row,
.row-alt{
  min-width: 960px;
}
.row {
  background: #fff;
  padding: 66px 0 44px 0;
}
.row-alt {
  background: #cbe2c1;
  background: -webkit-linear-gradient(to right, #a1d3b0, #f6f1d3);
  background: -moz-linear-gradient(to right, #a1d3b0, #f6f1d3);
  background: linear-gradient(to right, #a1d3b0, #f6f1d3);
  padding: 44px 0 22px 0;
}
```

Стилі `row-alt` на місці, скористаємося ними на всіх внутрішніх сторінках. В даний момент всі вони містять елемент `<section>` з класом `container`. Потім, всередині кожного `<section>` є елемент `<h1>`, що містить заголовок сторінки.

Необхідно змінити ці елементи `<section>` так само, як було зроблено з розділом з тизером на головній сторінці. Обернемо кожен елемент `<section>` з класом `container` елементом `<section>` з класом `row-alt`. Потім змінимо кожен елемент `<section>` з класом `container` на елемент `<div>` для кращої відповідності семантики.

Кожна з внутрішніх сторінок повинна тепер включати в себе наступне:

```
<section class="row-alt">
  <div class="container">

    <h1>...</h1>

  </div>
</section>
```

Зробимо вступ на сторінках трохи більш привабливим. Почнемо з додавання вступу на кожній сторінці трохи нижче `<h1>` в кожному елементі `<section>` з класом `row-alt`. Сторінка [direction.html](#), наприклад, тепер може включати наступний розділ:

```
<section class="row-alt">
```

```
<div class="container">
```

```
  <h1>Напрямки роботи</h1>
```

```
  <p>Інформаційні технології у науці, освіті, економіці, логістиці, туристичній сфері, морському транспорті; управління проектами регіонального розвитку; впровадження інновацій та сучасних технологій; прогнозування соціально-економічних процесів за умов невизначеності та ризику; математичні методи, моделі, інформаційні системи та технології в економіці; сучасні тенденції розвитку інформаційних технологій; системи підтримки прийняття рішень інноваційного розвитку; інвестиційне проектування в різних сферах суспільного життя.</p>
```

```
</div>
```

```
</section>
```

Додатково зі вставкою абзацу також змінимо деякі стилі в розділі зі вступом. Щоб зробити це, додамо клас `lead` до елемента `<div>`, який вже містить клас `container`. Він може бути знайдений усередині елемента `<section>` з класом `row-alt`. Вступ для кожної внутрішньої сторінки тепер буде виглядати так:

```
<section class="row-alt">
```

```
  <div class="lead container">
```

```
    ...
```

```
  </div>
```

```
</section>
```

Після цього вирівнюємо по центру весь текст всередині елементів `<div>`. Також збільшимо `font-size` і `line-height` всіх абзаців у цих `<div>`.

Створіть новий розділ для вступу в нашому файлі `main.css`, трохи нижче розділу з типографікою, і додайте наступні стилі:

```
/*
```

```
=====
```

```
Вступ
```

```
=====
```

```
*/
```

```
.lead {
```

```
  text-align: center;
```

```
}
```

```
.lead p {
```

```
  font-size: 21px;
```

```
  line-height: 33px;
```

```
}
```

Переконайтеся, код був оновлений для всіх внутрішніх сторінок (рис. 6.2).

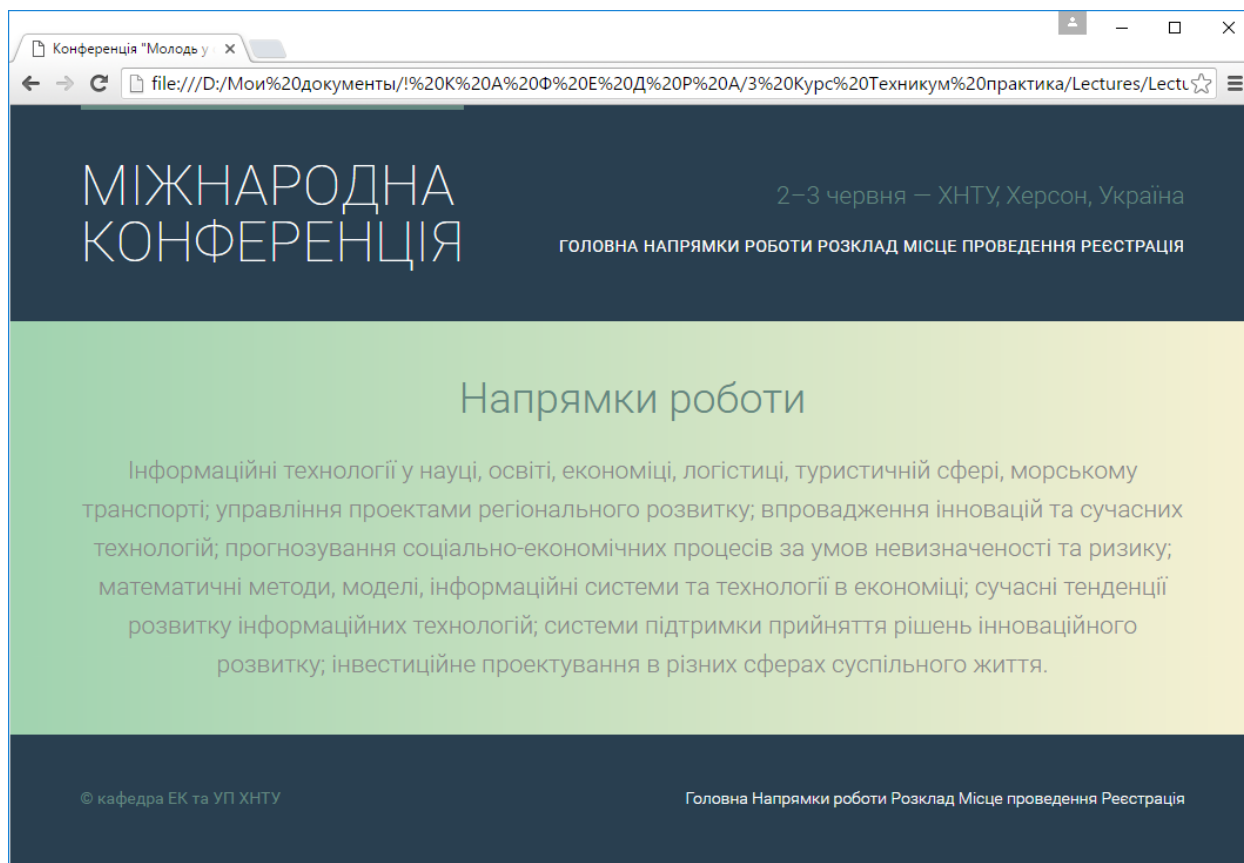


Рис. 6.2. Сторінка сайту напрямків роботи конференції

Лабораторна робота № 2

У вашому проєкті, створеному в попередній лабораторній роботі:

1. Змініть властивості шрифтів на необхідні.
2. Встановіть необхідний розмір шрифту, висоту рядка, накреслення, колір.
3. Застосуйте до тексту, там, де це необхідно, такі властивості як `text-decoration`, `text-transform`, `letter-spacing`.
4. Вбудуйте мінімум два веб-шрифту в ваш проєкт. Один з них повинен вбудовуватися як локальний шрифт, а другий підключатися за посиланням з сайту <https://fonts.google.com>.
5. Додайте фон, а також градієнт до сторінок вашого сайту в тих місцях, де це необхідно.

Тема 7. Створення списків

Зараз навігаційні меню в елементах `<header>` і `<footer>` на сторінках сайту складаються з декількох посилань. Ці елементи можуть бути краще організовані у вигляді невпорядкованого списку.

Використання невпорядкованого списку (через елемент `<ul>`) і пунктів списку (через елемент `<li>`) задасть структуру навігаційних меню.

Необхідно змінити значення `display` для елементів `<li>` на `inline-block`, щоб всі вони вирівнювалися в горизонтальний рядок. Після цього також необхідно врахувати порожній

простір зліва між кожним елементом `<li>`. Відомо, що відкриття коментаря HTML в кінці елемента `<li>` і закриття коментаря на початку елемента `<li>` видаляють цей порожній простір.

Пам'ятаючи про це, розмітка навігаційного меню всередині елемента `<header>` буде виглядати наступним чином:

```
<nav class="nav primary-nav">
  <ul>
    <li><a href="index.php">Головна</a></li><!--
    --><li><a href="direction.php">Тематика</a></li><!--
    --><li><a href="schedule.php">Розклад</a></li><!--
    --><li><a href="venue.php">Місце проведення</a></li><!--
    --><li><a href="register.php">Реєстрація</a></li>
  </ul>
</nav>
```

Аналогічно, розмітка для навігаційного меню всередині елемента `<footer>` буде мати такий вигляд:

```
<nav class="nav">
  <ul>
    <li><a href="index.php">Головна</a></li><!--
    --><li><a href="direction.php">Тематика</a></li><!--
    --><li><a href="schedule.php">Розклад</a></li><!--
    --><li><a href="venue.php">Місце проведення</a></li><!--
    --><li><a href="register.php">Реєстрація</a></li>
  </ul>
</nav>
```

З щойно створеним маркованим списком переконаємося, що пункти списку вирівняні по горизонталі й очистимо їх стилі. Будемо використовувати існуючий клас `nav` для нових стилів.

Почнемо з того, щоб всі елементи `<li>` всередині будь-якого елемента зі значенням `nav` атрибута `class` відображалися як `inline-block`, щоб включити горизонтальні `margin` і щоб можна було вирівнювати елементи по вертикалі.

Крім того, скористаємося псевдокласами: `last-child` щоб визначити останній елемент `<li>` і скинути його правий `margin` в 0. Це гарантує, що будь-який горизонтальний простір між елементами `<li>` і краєм його батька зникне.

У файл `main.css`, нижче стилів навігації додамо наступний `CSS`:

```
.nav li {
  display: inline-block;
  margin: 0 10px;
  vertical-align: top;
}
.nav li:last-child {
  margin-right: 0;
}
```

Меню навігації не єдине місце, де будуть використовуватись списки. Також можна застосувати їх на деяких внутрішніх сторінках, включаючи сторінку Тематика. Додамо кілька тематичних секцій в конференцію.

У файлі `direction.html`, трохи нижче розділу зі вступом, створимо новий розділ, де представимо всі наші секції. Повторно задіємо деякі існуючі стилі і скористаємося елементом `<section>` з класом `row`, щоб обернути всі секції і застосуємо білий фон і поля за ними. Всередині елемента `<section>` додамо елемент `<div>` з класом `grid` щоб вирівняти по центру сторінки всі наші секції, і це дозволить також застосувати кілька колонок.

Поки `HTML` нижче розділу зі вступом виглядає наступним чином:

```
<section class="row">
  <div class="grid">

  </div>
</section>
```

Усередині `grid` кожна секція буде розмічена з власним елементом `<section>`, який включає дві колонки. Перша колонка відміряє дві третини елемента `<section>` і буде розмічена за допомогою елемента `<div>`. Друга колонка відміряє третину елемента `<section>` і буде розмічена за допомогою елемента `<aside>`.

Використовуючи існуючі класи `col-2-3` і `col-1-3`, схема для розділу з секціями буде виглядати так:

```
<section id=" theme1">
  <div class="col-2-3">
    ...
  </div><!--

  --><aside class="col-1-3">
    ...
  </aside>

</section>
```

Тут є кілька пунктів, на які варто звернути увагу. Перше, кожен елемент `<section>` для кожного спікера включає в себе атрибут `id` з номером секції в якості значення. Пізніше, коли буде створено розклад для конференції, ці атрибути будуть слугувати якорями, що дозволять зв'язати розклад з секцією.

Крім того, що закриваючий тег елемента `<div>` супроводжується відкриттям коментаря в `HTML` і відкриваючим тегом елемента `<aside>`, якому передуює закриття коментаря. Оскільки класи для колонок будуть відображати ці елементи як рядково-блокові, видаляється порожній простір, який з'явиться між ними.

Усередині колонки `2/3` розмічений елементом `<div>` будемо використовувати кілька заголовків і абзаців, щоб показати ім'я секції, короткий опис і повний опис.

З огляду на це вміст та розділ секції буде виглядати наступним чином:

```

<section class="row">
  <div class="grid">
    <section id="theme1">
      <div class="col-2-3">
        <h2>Секція "Інформаційні технології"</h2>
        <h5>
          Інформаційні технології у науці, освіті, економіці
        </h5>
        <p>
          Чим далі, тим частіше ...
        </p>
      </div><!--
    --><aside class="col-1-3">
      . . .
    </aside>
  </section>

```

У колонку 1/3 розмічену елементом `<aside>` додамо елемент `<div>` з класом `curator`. Будемо використовувати елемент `<div>`, і додамо стилі до цього елементу найближчим часом.

Перш ніж перейти до будь-яких стилів, додамо невпорядкований список в елемент `<div>`, який включає в себе пункти списку, що мають відношення до секції.

Тепер **HTML** для секції буде виглядати наступним чином:

```

<section class="row">
  <div class="grid">
    <section id="theme1">
      <div class="col-2-3">
        <h2>Секція "Інформаційні технології"</h2>
        <h5>
          Інформаційні технології у науці, освіті, економіці
        </h5>
        <p>
          Чим далі, тим частіше ...
        </p>
      </div><!--
    --><aside class="col-1-3">
      <div class="curator">
        <ul>
          <li>Куратор секції</li>
          <li>Альберт Енштейн</li>
        </ul>
      </div>
    </aside>
  </section>

```

Коли елемент `<div>` з класом `curator` готовий, можна додати до нього деякі стилі. Почнемо з додавання нового розділу у файл `main.css` для стилів сторінки Тематики. В нього додамо однопиксельну суцільну рамку з радіусом 5 пікселів навколо будь-якого елементу з класом `curator`.

Далі додамо верхній `margin 88` пікселів, щоб помістити елемент на тому ж вертикальному рядку, що і перший абзац секції, а також додамо `22` пікселя по вертикалі для `padding`, щоб забезпечити місце для вкладеного маркованого списку.

Нарешті, вирівнюємо по центру весь текст всередині елемента. В цілому, CSS для класу `curator` виглядає наступним чином:

```
/*
=====
Тематика
=====
*/
.curator {
border: 1px solid #dfe2e5;
border-radius: 5px;
margin-top: 88px;
padding-bottom: 22px;
text-align: center;
}
```

Необхідно, щоб секції залишалися на рівній відстані одна від одної по вертикалі. Щоб зробити це, створимо набір правил для класу `theme`, який включає в себе нижній `margin 44` пікселя, таким чином:

```
. theme {
margin-bottom: 44px;
}
```

Можна застосувати цей клас до `<section>` для кожного спікера, за умови, що це не останній спікер. Ми опускаємо цей клас для останнього спікера, тому що не хочемо створити непотрібні відступи перед елементом `<footer>`. З більш ніж однією секцією розмітка буде виглядати наступним чином:

```
<section class="row">
  <div class="grid">

    <section class=" theme" id="theme1">

      <div class="col-2-3">
        ...
      </div><!--

    --><aside class="col-1-3">
      ...
    </aside>

  </section>

  ...    ...    ...
```

```

<section id=" theme5">

  <div class="col-2-3">
    ...
  </div><!--

--><aside class="col-1-3">
  ...
</aside>

</section>

</div>
</section>

```

Наше меню навігації в даний час завершено і сторінка Спікери почала набирати свою форму (рис. 7.1).

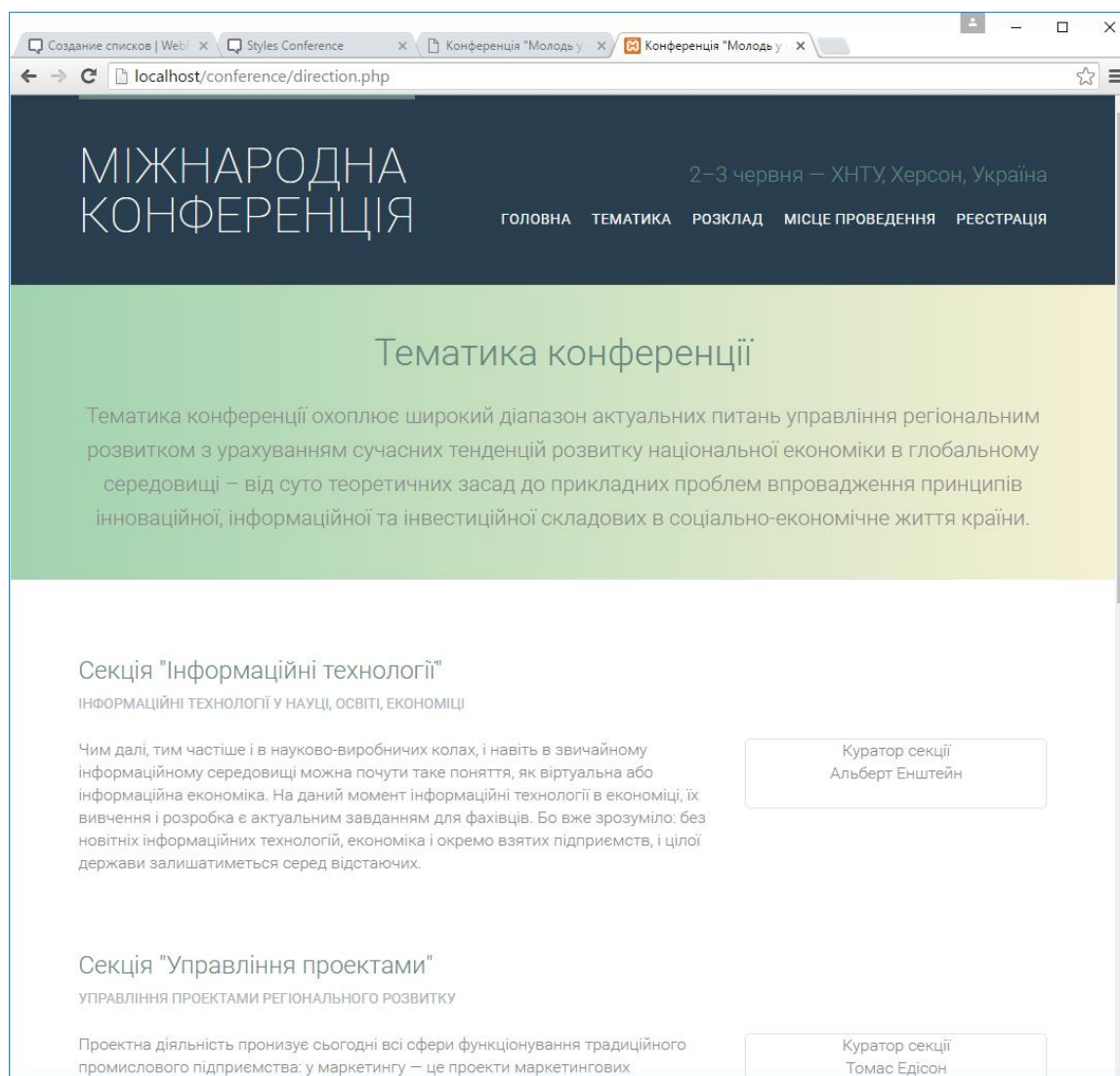


Рис. 7.1. Сторінка Спікери

Тема 8. Додавання медіа-контенту

Почнемо з додавання декількох зображень на головну сторінку. Зокрема, додамо зображення в кожен розділ з тизером, що просуває деякі сторінки.

Перш, ніж перейти до коду, створимо нову папку з назвою `images` в нашій `files`. Потім, в папці `images` створимо іншу папку з назвою `home` спеціально для зображень головної сторінки. В папку `home` додамо три зображення: `direction.jpg`, `schedule.jpg` і `kntu.jpg`.

Далі, у файлі `index.html`, кожен розділ з тизером містить елемент `<a>`, який одночасно обгортає елементи `<h3>` і `<h5>`. Перемістимо елемент `<h5>` вище елемента `<a>` і замінимо його елементом `<img>`. Значення атрибута `src` для кожного елемента `<img>` буде відповідати структурі папок і імені створеного файлу, а значення атрибута `alt` буде описувати зміст кожного зображення.

HTML для першого тизера зі спікерами буде виглядати наступним чином:

```
<section class="teaser col-1-3">
  <a href="direction.php">
    <h5>Тематика</h5>
    
    <h3>Питання, що потрубують негайного вирішення</h3>
  </a>
  <p>Будуть розглянуті важливі питання, що стосуються аспектів управління проектами розвитку регіону.</p>
</section>
```

Застосуємо цей шаблон також для тизерів розкладу і місця проведення.

Тепер, коли було додано кілька зображень на головну сторінку, потрібно трохи очистити їх стилі і переконаватися, що вони правильно вписуються в макет сторінки.

Зображення за замовчуванням є рядково-блоковими елементами, то давайте змінимо зображення в розділі з тизерами на блокові елементи. Також встановимо їх максимальну ширину як `100%`, щоб гарантувати, що вони не перевищують ширину відповідних їм колонок. Зміна значення цієї ширини важлива, оскільки вона дозволяє зображення підлаштовуватися під ширину колонок за необхідності.

Нарешті, трохи заокруглим кути зображень і застосуємо до них нижній `margin 22` пікселя, забезпечуючи невеликий вільний простір.

Після додавання цих нових стилів до існуючих головної сторінки (за допомогою класу `teaser` в якості відбіркового селектора для елементів `<img>`), CSS буде виглядати наступним чином:

```
.teaser img {
  border-radius: 5px;
  display: block;
  margin-bottom: 22px;
  max-width: 100%;
}
```

В результаті головна сторінка буде мати вигляд (рис. 8.1).

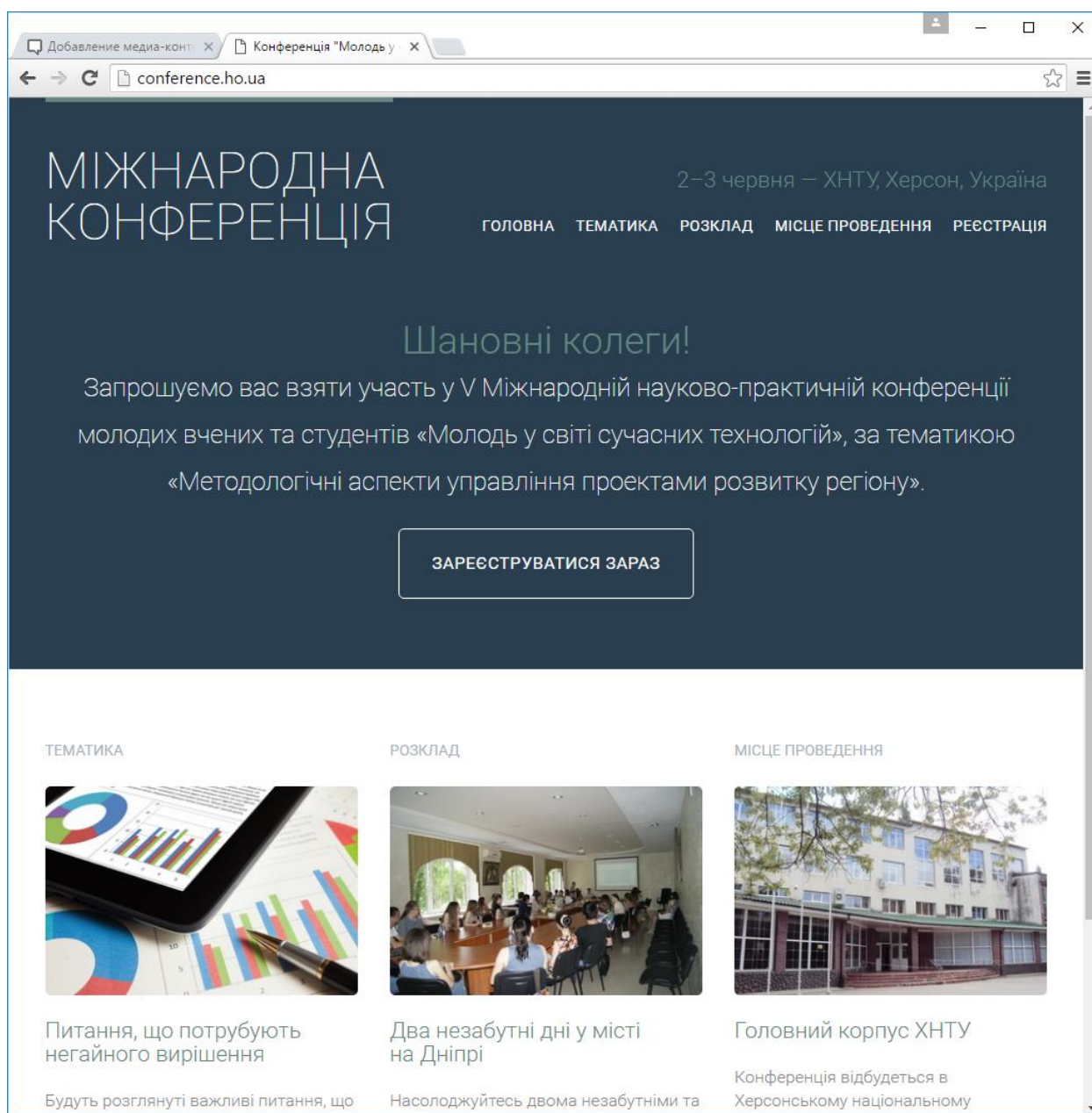


Рис. 8.1. Головна сторінка конференції після додавання зображень

Далі додамо зображення всіх кураторів секцій на сторінці Спікери. Почнемо зі створення папки `curators` в нашій папці `images` і помістимо туди зображення всіх кураторів.

У файлі `direction.html` додамо елемент `<img>` всередину кожного елемента `<aside>` з інформацією про куратора. Розмістимо кожен елемент `<img>` всередині елемента `<div>` зі значенням атрибута `class` вище елемента `<ul>`.

Значення атрибута `src` кожного зображення буде відповідати папці `curators`; значенням атрибута `alt` буде ім'я куратора.

Елемент `<aside>` для першого куратора буде виглядати наступним чином:

```
<aside class="col-1-3">
  <div class="curator">
    
    <ul>
```

```

        <li>Куратор секції</li>
        <li>Альберт Енштейн</li>
    </ul>
</div>
</aside>

```

Цей самий шаблон додавання зображення слід застосувати до всіх інших кураторів секцій.

Аналогічно із зображеннями на головній сторінці так само застосуємо деякі стилі для зображень на сторінці «Тематика конференції».

Почнемо з застосування властивості `border-radius` зі значенням `50%`, перетворюючи зображення в кола. Звідси встановимо фіксовану висоту `130` пікселів для кожного зображення і поставимо вертикальне вирівнювання по верху рядка, в якому вони знаходяться.

Після цього можна застосувати до зображень вертикальні відступи. Використовуючи негативний `margin` зверху в `66` пікселів, витягнемо трохи зображення з елемента `<aside>` і вирівняємо по центру верхньої межі елемента `<div>` з класом `curator`. Далі, застосування `margin` як `22` пікселя до нижньої частини зображення забезпечує простір між зображенням і елементом `<ul>` нижче нього.

Після того, як було додано ці нові стилі до існуючих на сторінці Спікери (за допомогою класу `curator` в якості відбіркового селектора для елементів `<img>`), CSS буде виглядати наступним чином:

```

.curator img {
  border-radius: 50%;
  height: 130px;
  margin: -66px 0 22px 0;
  vertical-align: top;
}

```

Оскільки використовується негативний `margin` для елемента `<img>` всередині елемента `<div>` з класом `curator`, необхідно прибрати `padding` зверху цього елемента `<div>`.

Раніше використовувалась властивість `padding` зі значенням `22px 0`. Замінімо її на властивість `padding-bottom`, тому що потрібно визначити тільки цей `padding` і використаємо значення `22` пікселя.

Новий набір правил для класу `curator` виглядає наступним чином:

```

.curator {
  border: 1px solid #dfe2e5;
  border-radius: 5px;
  margin-top: 88px;
  padding-bottom: 22px;
  text-align: center;
}

```

В результаті сторінка кураторів конференції буде мати вигляд (рис. 8.2).

Добавление медиа-конт... Конференція "Молодь у ...

conference.ho.ua/direction.php

МІЖНАРОДНА КОНФЕРЕНЦІЯ

2–3 червня — ХНТУ, Херсон, Україна

ГОЛОВНА ТЕМАТИКА РОЗКЛАД МІСЦЕ ПРОВЕДЕННЯ РЕЄСТРАЦІЯ

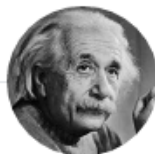
Тематика конференції

Тематика конференції охоплює широкий діапазон актуальних питань управління регіональним розвитком з урахуванням сучасних тенденцій розвитку національної економіки в глобальному середовищі – від суто теоретичних засад до прикладних проблем впровадження принципів інноваційної, інформаційної та інвестиційної складових в соціально-економічне життя країни.

Секція "Інформаційні технології"

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ У НАУЦІ, ОСВІТІ, ЕКОНОМІЦІ

Чим далі, тим частіше і в науково-виробничих колах, і навіть в звичайному інформаційному середовищі можна почути таке поняття, як віртуальна або інформаційна економіка. На даний момент інформаційні технології в економіці, їх вивчення і розробка є актуальним завданням для фахівців. Бо вже зрозуміло: без новітніх інформаційних технологій, економіка і окремо взятих підприємств, і цілої держави залишатиметься серед відстаючих.



Куратор секції
Альберт Ейнштейн

Секція "Управління проектами"

УПРАВЛІННЯ ПРОЕКТАМИ РЕГІОНАЛЬНОГО РОЗВИТКУ

Проектна діяльність пронизує сьогодні всі сфери функціонування традиційного промислового підприємства: у маркетингу — це проекти маркетингових досліджень, рекламних акцій, виведення на ринок нових продуктів, завоювання нових ринків збуту; у дослідно-конструкторських підрозділах — це проекти розробки нових продуктів, технологій; у виробництві — проекти освоєння випуску нової продукції, технічного переозброєння, впровадження нових технологій; у збуті — проекти побудови торговельно-збутової мережі тощо.



Куратор секції
Томас Едісон

Секція "Інновації та інвестиції"

ПРИЙНЯТТЯ РІШЕНЬ ІННОВАЦІЙНОГО ТА ІНВЕСТИЦІЙНОГО РОЗВИТКУ

Інвестиції й інновації — дві нерозривно пов'язані сфери економічної діяльності, що найбільшою мірою були і залишаються уражені кризою. В сучасних умовах склалася така ситуація, що інновації, які раніше здійснювалися за рахунок централізованих джерел, звелися до незначних величини, тоді як інвестиції, що володіли внутрішньою структурою, втратили її і стали невідпорядкованими і неорганізованими, перетворилися тільки в короткострокові цілі інвесторів.



Куратор секції
Мілтон Фрідман

Секція "Математичні методи та моделі"

Рис. 8.2. Сторінка кураторів конференції

Вбудовані фрейми пропонують відмінний спосіб додавання динамічного вмісту на сторінку. Оновимо сторінку «Місце проведення» деякими картами.

Перед додаванням будь-яких карт або вбудованих фреймів давайте спочатку підготуємо сторінку для сітки з двох колонок. Нижче розділу зі вступом додамо елемент `<section>` з класом `row`, щоб визначити новий розділ сторінки і включити деякі загальні стилі, такі як білий фон і вертикальний `padding`.

Безпосередньо всередині цього елемента `<section>` додамо елемент `<div>` з класом `grid`. Клас `grid` вирівнює контент по центру сторінки і призначений для обгортання колонок 1/3 і 2/3.

Поки головний розділ файлу `venue.html` виглядає наступним чином:

```
<section class="row">
  <div class="grid">
    ...
  </div>
</section>
```

В елементі `<div>` з класом `grid` буде один розділ. Додамо ще один новий елемент `<section>` і дамо йому унікальний клас `venue`, який відповідає його змісту. Будемо використовувати ці класи для додавання `margin` знизу кожного розділу.

Таким чином `HTML` повинен виглядати так:

```
<section class="row">
  <div class="grid">
    <section class="venue">
      ...
    </section>
  </div>
</section>
```

Тепер створимо новий розділ у файлі `main.css` для стилів сторінки «Місце проведення». Додамо 22-піксельний `margin` внизу елемента `<section>` з класом `venue`, щоб забезпечити деякий простір між ним і елементом `<footer>` нижче.

Новий розділ у файлі `main.css` виглядає наступним чином:

```
/*
=====
Місце проведення
=====
*/
.venue {
  margin-bottom: 22px;
}
```

Тепер прийшов час для створення двох колонок у новому елементі `<section>`. Почнемо з додавання елемента `<div>` з класом `col-1-3`, щоб встановити колонку 1/3. Після цього додамо елемент `<iframe>` з класом `col-2-3`, щоб встановити колонку 2/3.

Маючи на увазі, що колонки, зроблені елементами `<div>` і `<iframe>` – це рядково-блокові елементи, тому необхідно видалити порожній простір, який буде відображатися між ними. Для цього відкриємо коментар HTML відразу після закриваючого тегу `<div>` і закриємо коментар безпосередньо перед відкриваючим тегом `<iframe>`.

В цілому, HTML для колонок виглядає наступним чином:

```
<section class="row">
  <div class="grid">
    <section class="venue">
      <div class="col-1-3"></div><!--
      --><iframe class="col-2-3"></iframe>
    </section>
```

Кожному з елементів `<div>` з класом `col-1-3` додамо назву місця всередині елемента `<h2>`, потім два елементи `<p>`. У першому елементі `<p>` додамо адресу місця проведення, а в другому `<p>` елемента додамо адресу сайту (з посиланням) і номер телефону.

Усередині кожного абзацу використовуємо елемент перекладу рядка `<br>`, щоб розмістити розриви в адресі і між сайтом і номером телефону.

Для елемента `<section>` з класом `venue` HTML виглядає наступним чином:

```
<section class="venue">
  <div class="col-1-3">
    <h2>Головний корпус ХНТУ</h2>
    <p>Бериславське шосе 24<br> м.Херсон, Україна</p>
    <p><a href="http://www.kntu.net.ua">Сайт ХНТУ</a> <br> (0552) 22-22-22</p>
  </div><!--
  --><iframe class="venue-map col-2-3"> </iframe>
</section>
```

Після цього дотримуйтесь інструкцій на [Google Maps](#) про те, як впровадити карту й отримати HTML для елемента `<iframe>`.

Скопіюйте HTML - елемент `<iframe>`, атрибут `src` і додайте на сторінку, де розташований існуючий елемент `<iframe>`.

При копіюванні елемента `<iframe>` з [Google Maps](#) необхідно переконатися, що зберігається атрибут `class` і значення `col-2-3` в існуючому `<iframe>`. Також необхідно бути обережним, щоб не пошкодити коментар HTML, який закривається безпосередньо перед відкриваючим тегом `<iframe>`. В результаті HTML виглядає наступним чином:

```
<section class="venue">
  <div class="col-1-3">
    <h2>Головний корпус ХНТУ</h2>
    <p>Бериславське шосе 24<br> м.Херсон, Україна</p>
    <p><a href="http://www.kntu.net.ua">Сайт ХНТУ</a> <br> (0552) 22-22-22</p>
  </div><!--
  --><iframe class="venue-map col-2-3"
src="https://www.google.com/maps/embed?pb=!1m14!1m8!1m3!1d36834.086791153284!2d32.6
1240339000696!3d46.67198870786595!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x40c41015f13bef37
%3A0xdb28696e80e3dea0!2z0JrQvtGAOL_Rg9GBIOKEljEg0KXQndCi0KMsINCROLXRgNC40YHQ9Cw0
```

```
LLRgdGM0LrQtSDRiC4sIDI0LCDQpdC10YDRgdC-0L0sINCi0LXRgNGB0L7QvdGB0YzQutCwINC-
0LHQu9Cw0YHRgtGM!5e0!3m2!1sru!2sua!4v1466348495370" frameborder="0" style="border:0"
allowfullscreen>
</iframe>
</section>
```

Тепер створимо новий клас `venue-map` і застосуємо його до елементу `<iframe>` поряд з існуючим класом `col-2-3`.

HTML для тегу `<iframe>` тепер виглядає наступним чином:

```
<iframe class="venue-map col-2-3" src=...
```

Після того, як клас `venue-map` застосовується до кожного елементу `<iframe>`, створимо набір правил для цього класу в нашому файлі `main.css`. Він включає у себе властивість `height` зі значенням `150` пікселів.

Набір правил для класу `venue-map`:

```
.venue-map {
  height: 150px;
}
```

Тепер у нас є сторінка з місцем проведення в комплекті з картами для різних місць нашої конференції (рис. 8.3).

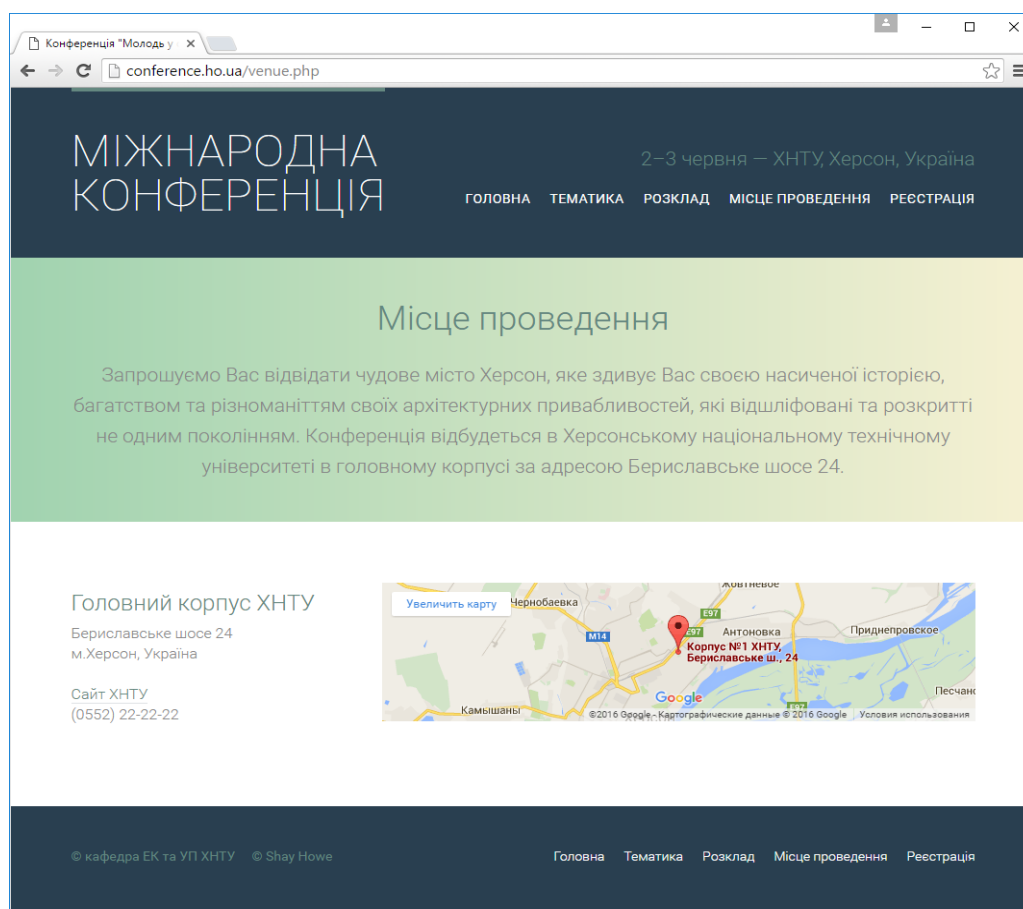


Рис. 8.3. Сторінка місця проведення конференції

Лабораторна робота № 3

У проєкті, створеному в попередніх лабораторних роботах:

1. Створіть мінімум два списки, один з яких повинен бути нумерований, а другий маркований. За необхідності додайте в проєкт сторінки і текстовий контент для створення списків. Необхідно вміти змінювати маркери і нумерацію у списку.
2. Додайте у ваш проєкт зображення, а також фонові картинки.
3. На окремій сторінці вашого сайту додайте мультимедійний контент: аудіо і відео, які розташовані локально у файловій системі сайту.
4. Додайте у ваш проєкт сторінку «Про нас», на якій розташуйте необхідний текст з контактною інформацією, також два елементи `<iframe>` – ролик з YouTube і карту з Google Maps.

Тема 9. Організація даних за допомогою таблиць

Тепер давайте завершимо сторінку сайту конференції Розклад. Почнемо з відкриття файлу `schedule.html` і додавання елемента `<section>` з класом `row`, так само, як це було зроблено з усіма іншими сторінками. Усередині цього нового `<section>` також додамо елемент `<div>` з класом `container`.

```
<section class="row">
  <div class="container">
    ...
  </div>
</section>
```

За допомогою цих елементів і класів ми створили новий розділ сторінки з білим `background` і вертикальним `padding` та вирівняли вміст сторінки по центру. Що змінилося тут порівняно з іншими сторінками - клас `container` на місці класу `grid` для елемента `<div>`. Оскільки на даній сторінці не буде використовуватись будь-який клас `col-`, то відмовляємося від класу `grid` на користь класу `container`.

У новому розділі додамо дві таблиці, по одній для кожного дня конференції. Таблиці будуть відображати події кожного дня за допомогою трьох стовпців і декількох рядків, містити шапку таблиці та її тіло.

Для початку визначимо структуру першої таблиці, включаючи елементи `<table>`, `<thead>` і `<tbody>`.

```
<section class="row">
  <div class="container">

    <table>
      <thead>
        ...
      </thead>
      <tbody>
```

```

...
</tbody>
</table>

</div>
</section>

```

В даний момент, хоча таблиця не містить будь-яких даних, до неї застосовуються деякі стилі. Зокрема, скидання, яке було додано на самому початку, встановивши для таблиць властивість `border-collapse` зі значенням `collapse`, а властивість `border-spacing` зі значенням `0`. Нам потрібні ці стилі, тому залишимо їх так як є. Проте, створимо новий розділ в файлі `main.css`, щоб додати деякі додаткові стилі.

У новому розділі стилів спеціально для сторінки Розклад (який з'явиться у самому низу), встановимо для елементів `<table>` властивість `width` як `100%` і нижній `margin` `44` пікселя.

Потім за допомогою псевдокласу `:last-child` визначимо останній елемент `<table>` в розділі і встановимо для нього нижній `margin` `0` пікселів. Це стане запобіжником конфлікту цієї таблиці з нижнім `padding`, що належить елементу `<section>` з класом `row`.

Поки новий розділ у файлі `main.css` виглядає наступним чином:

```

/*
=====
Розклад
=====
*/
table {
  margin-bottom: 44px;
  width: 100%;
}
table:last-child {
  margin-bottom: 0;
}

```

Тепер додамо дані в таблицю. Почнемо з першого дня конференції, 2 червня. В елементі `<thead>` у таблиці додамо елемент `<tr>`. Першою коміркою в рядку буде елемент `<th>` з назвою «Конференція» для цього конкретного дня. Оскільки елемент `<th>` є заголовком рядка, також додамо до нього атрибут `scope` зі значенням `row`.

Після елемента `<th>` йде елемент `<td>` з датою «2 червня». Тепер у таблиці буде три стовпці, перший з яких слугує заголовком таблиці, він визначає час, а другі два – це звичайні елементи таблиці, які вказують спікерів на цей час. Оскільки в цьому рядку не потрібно два окремих спікери, то додамо атрибут `colspan` зі значенням `2` до елемента `<td>`, змушуючи його об'єднати два стовпці.

Код для таблиці тепер виглядає наступним чином:

```

<table>
  <thead>
    <tr>
      <th scope="row">

```

```

    Конференція
  </th>
  <td colspan="2">
    2 червня
  </td>
</tr>
</thead>
<tbody>
  ...
</tbody>
</table>

```

Усередині елемента `<tbody>` заповнимо денні заходи. Почнемо з додавання елемента `<tr>` з елементами `<th>` і `<td>` безпосередньо в рядку.

До всіх наступних елементів `<th>` додамо атрибут `scope` зі значенням `row`, щоб семантично визначити цей елемент як заголовок рядка. Потім в елементі `<th>` додамо елемент `<time>`, який показує час першого заходу - `8:30`. Також включимо атрибут `datetime` для елемента `<time>` зі значенням часу в годинах, хвилинах і секундах - `8:30:00`.

В елемент `<td>`, який слідує за елементом `<th>`, включимо назву заходу (оскільки в цей час немає ніяких спікерів) - «Реєстрація», в даному випадку. Оскільки існує тільки один захід у цей час, також включимо атрибут `colspan` зі значенням 2 в елементі `<td>`.

В цілому, код таблиці виглядає наступним чином:

```

<table>
  <thead>
    <tr>
      <th scope="row">
        Конференція
      </th>
      <td colspan="2">
        2 червня
      </td>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">
        <time datetime="08:30:00">8:30</time>
      </th>
      <td colspan="2">
        Реєстрація
      </td>
    </tr>
  </tbody>
</table>

```

Для другого рядка всередині елемента `<tbody>` додамо елемент `<tr>` трохи нижче попереднього ряду. Потім додамо елемент `<th>` з атрибутом `scope` і значенням `row` і знову додамо елемент `<time>` з відповідним часом і атрибутом `datetime` в цьому елементі `<th>`.

Після `<th>` додамо два елементи `<td>` для двох кураторів секцій, які виступають одночасно. Безпосередньо всередині кожного елемента `<td>` додамо елемент `<a>`, який посилається туди, де куратор секції розташований на сторінці Тематики конференції. Пам'ятаємо, що ми додали атрибут `id` з номером кожної секції (теми) до батьківського елемента? Використовуючи це значення атрибута `id`, якому передуватиме ім'я файлу `direction.html` і знак решітки `#`, ми можемо безпосередньо зв'язати опис доповіді виступаючого з описом його секції на сторінці тематики конференції.

Всередину елемента `<a>` включимо елемент `<h4>` з ім'ям куратора секції з подальшою назвою його доповіді.

Код для перших двох семінарів виглядає наступним чином:

```
<table>
<thead>
  <tr>
    <th scope="row">
      Конференція
    </th>
    <td colspan="2">
      2 червня
    </td>
  </tr>
</thead>
<tbody>
  <tr>
    <th scope="row">
      <time datetime="08:30:00">8:30</time>
    </th>
    <td colspan="2">
      Реєстрація
    </td>
  </tr>
  <tr>
    <th scope="row">
      <time datetime="09:00:00">9:00</time>
    </th>
    <td>
      <a href="direction.php#theme1">
        <h4>Альберт Енштейн</h4>
        Доповідь про сучасний стан інформаційних технологій в економіці
      </a>
    </td>
    <td>
      <a href="direction.php#theme2">
        <h4>Томас Едісон</h4>
        Доповідь про сучасний стан виробництва світлодіодних ламп та їх вплив на
        енергозощадження
      </a>
    </td>
  </tr>
</tbody>
</table>
```

```
</tbody>
</table>
```

Далі можна повторити цей шаблон для кожного виду заходу і доповідача, щоб закінчити нашу першу таблицю, а потім додати ще одну для наступного дня конференції.

При цьому майте на увазі, що шапка таблиці завжди буде включати в себе заголовок таблиці з подією дня й елементи таблиці, які поєднують дві колонки з датою.

Потім у тілі кожної таблиці, кожен рядок буде містити заголовок таблиці з часом. Після заголовка таблиці йде захід, доповідач або кілька доповідачів. Заходи без доповідей будуть знаходитися в одному елементі `<td>`, який об'єднує два стовпці. Якщо представлений тільки один доповідач у певний час, то він буде перебувати в одному елементі `<td>`, який об'єднує два стовпці і містити елементи `<a>` і `<h4>`.

Якщо є два доповідача одночасно, то кожен доповідач буде знаходитися всередині свого власного елемента `<td>`, як і раніше.

Повний код для будь-яких двох таблиць можна знайти на сайті конференції. Для довідки, таблиця для першого дня, який включає в себе обід і більше двох спікерів, виглядає наступним чином:

```
<table>
  <thead>
    <tr>
      <th scope="row">
        Конференція
      </th>
      <td colspan="2">
        2 червня
      </td>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>
        <time datetime="08:30:00">8:30</time>
      </th>
      <td colspan="2">
        Реєстрація
      </td>
    </tr>
    <tr>
      <th scope="row">
        <time datetime="09:00:00">9:00</time>
      </th>
      <td>
        <a href="direction.php#theme1">
          <h4>Альберт Енштейн</h4>
          Доповідь про сучасний стан інформаційних технологій в економіці
        </a>
      </td>
      <td>
```

```

        <a href="direction.php#theme2">
            <h4>Томас Едісон</h4>
            Доповідь про сучасний стан виробництва світлодіодних
            ламп та їх вплив на енергозаощадження
        </a>
    </td>
</tr>
<tr>
    <th scope="row">
        <time datetime="12:30:00">12:30</time>
    </th>
    <td colspan="2">
        Обід
    </td>
</tr>
<tr>
    <th scope="row">
        <time datetime="14:00:00">14:00</time>
    </th>
    <td>
        <a href="direction.php#theme3">
            <h4>Мілтон Фрідман</h4>
            Інвестиції й інновації — дві нерозривно пов'язані
            сфери економічної діяльності
        </a>
    </td>
    <td>
        <a href="direction.php#theme4">
            <h4>Рене Декарт</h4>
            Прогнозування соціально-економічних процесів за умов невизначеності
        </a>
    </td>
</tr>
</tbody>
</table>

```

Тепер таблиці набули форми, настав час додати до них трохи стилю. Почнемо з додавання деяких загальних стилів для елементів `<th>` і `<td>`. Для них додамо нижній **padding** 22 пікселя і вертикальне вирівнювання **top**. Для елемента `<th>` спеціально додамо правий **padding** 45 пікселів, вирівнювання тексту як **right** і **width** як 20%. Потім для елементів `<td>` додамо **width** як 40%.

Нижче існуючих стилів таблиць розкладу код повинен виглядати наступним чином:

```

th,
td {
    padding-bottom: 22px;
    vertical-align: top;
}
th {
    padding-right: 45px;

```

```

text-align: right;
width: 20%;
}
td {
width: 40%;
}

```

Далі додамо стиль до шапки таблиці і до елементів у ній. Встановимо `line-height` як 44 пікселя тільки для елемента `<thead>`, `color` як `#648880` і `font-size` як 24 пікселя для всіх елементів `<th>`, вкладених в `<thead>`. Наші нові стилі включають наступне:

```

thead {
  line-height: 44px;
}
thead th {
  color: #648880;
  font-size: 24px;
}

```

Шапка таблиці виглядає добре, так що додамо деякі стилі для тіла таблиці. Почнемо з елементів `<th>` вкладених в `<tbody>`: змінимо їх колір, додамо деякі шрифтові і текстові властивості і невеликий верхній `padding`.

```

tbody th {
  color: #a9b2b9;
  font-size: 14px;
  font-weight: 400;
  padding-top: 22px;
  text-transform: uppercase;
}

```

Також додамо деякі стилі до елементів `<td>`, вкладених в `<tbody>`, починаючи з верхньої межі і `padding`. Стилізуємо елементи `<td>`, які об'єднуються в один стовпець, додавши до них `padding` 15 пікселів праворуч, щоб сформувати лівий стовпець і 15 пікселів `padding` ліворуч, щоб сформувати правий стовпець. Це встановить у цілому 30 пікселів `padding` між двома стовпцями, зберігаючи кожен клітинку однакового розміру. Нам не потрібно застосовувати лівий або правий `padding` до елементів `<td>`, які об'єднують два стовпці.

Додамо всі ці горизонтальні `padding` за допомогою псевдокласів `:first-of-type`, `:last-of-type` і `:only-of-type`. Ці псевдокласи працюють дуже схоже на псевдоклас `:last-child`, який був використовуваний раніше.

Псевдоклас `:first-of-type` вибере перший елемент цього типу в батьківському елементі. У нашому випадку, селектор `td:first-of-type` вибере перший елемент `<td>` всередині елемента `<tr>`. Потім псевдоклас: `last-of-type` вибере останній елемент цього типу в батьківському елементі.

Знову ж таки, в нашому випадку, селектор `td:last-of-type` вибере останній елемент `<td>` всередині елемента `<tr>`. Нарешті, псевдоклас `:only-of-type` вибере елемент, тільки якщо це елемент даного типу в батьківському елементі. Тут селектор `td:only-of-type` вибере тільки

елемент `<td>`, якщо це тільки `<td>` всередині елемента `<tr>`, наприклад, коли `<td>` об'єднує два стовпці. Таким чином маємо:

```
tbody td {
  border-top: 1px solid #dfe2e5;
  padding-top: 21px;
}
tbody td:first-of-type {
  padding-right: 15px;
}
tbody td:last-of-type {
  padding-left: 15px;
}
tbody td:only-of-type {
  padding-left: 0;
  padding-right: 0;
}
```

Розклад і таблиця для його відображення з'єднуються разом. Налаштуємо кілька стилів для існуючих елементів щоб очистити макет. Почнемо з того, що зробимо всі посилання в таблиці трохи сірими. Якщо ми націлюватимемося тільки на елементи `<a>` в таблиці, наші заголовки з ім'ям спікера всередині посилання залишаться зеленими, в той час як назва доповіді буде сірою, створюючи приємний контраст між ними.

Також приберемо нижній `margin` для елементів `<h4>` в таблиці, дозволяючи імені спікера розташовуватися ближче до заголовка. Ці стилі реалізовані за допомогою наступного коду:

```
table a {
  color: #888;
}
table h4 {
  margin-bottom: 0;
}
```

Нарешті, створимо деякий візуальний контраст між різними видами діяльності, що відбуваються протягом дня. Всі доповіді добре виглядають після останніх змін. Для всіх інших видів діяльності, таких як реєстрація, обід і перерви (які в тілі таблиці), а також для дати (яка в заголовку таблиці) будемо використовувати світлий сірий.

Зробимо це, створивши новий клас `schedule-offset` і призначивши йому `color` зі значенням `#a9b2b9`.

```
.schedule-offset {
  color: #a9b2b9;
}
```

Після того, як клас знаходиться на місці, додамо його до всіх елементів `<td>`, які об'єднують дві колонки і включають дату, або захід – реєстрацію, обід або перерву. HTML буде виглядати наступним чином:

```

<table>
<thead>
  <tr>
    <th scope="row">
      Конференція
    </th>
    <td class="schedule-offset" colspan="2">
      2 червня
    </td>
  </tr>
</thead>
<tbody>
  <tr>
    <th scope="row">
      <time datetime="08:30:00">8:30</time>
    </th>
    <td class="schedule-offset" colspan="2">
      Регістрація
    </td>
  </tr>
  ...
  <tr>
    <th scope="row">
      <time datetime="12:30:00">12:30</time>
    </th>
    <td class="schedule-offset" colspan="2">
      Обід
    </td>
  </tr>
  ...
</tbody>
</table>

```

Таблиці, які на перший погляд виглядають просто, можуть виявитися досить складними, як у випадку з нашим розкладом для конференції. Доброю новиною є те, що наш розклад тепер завершений і виглядає чудово (рис. 9. 1).

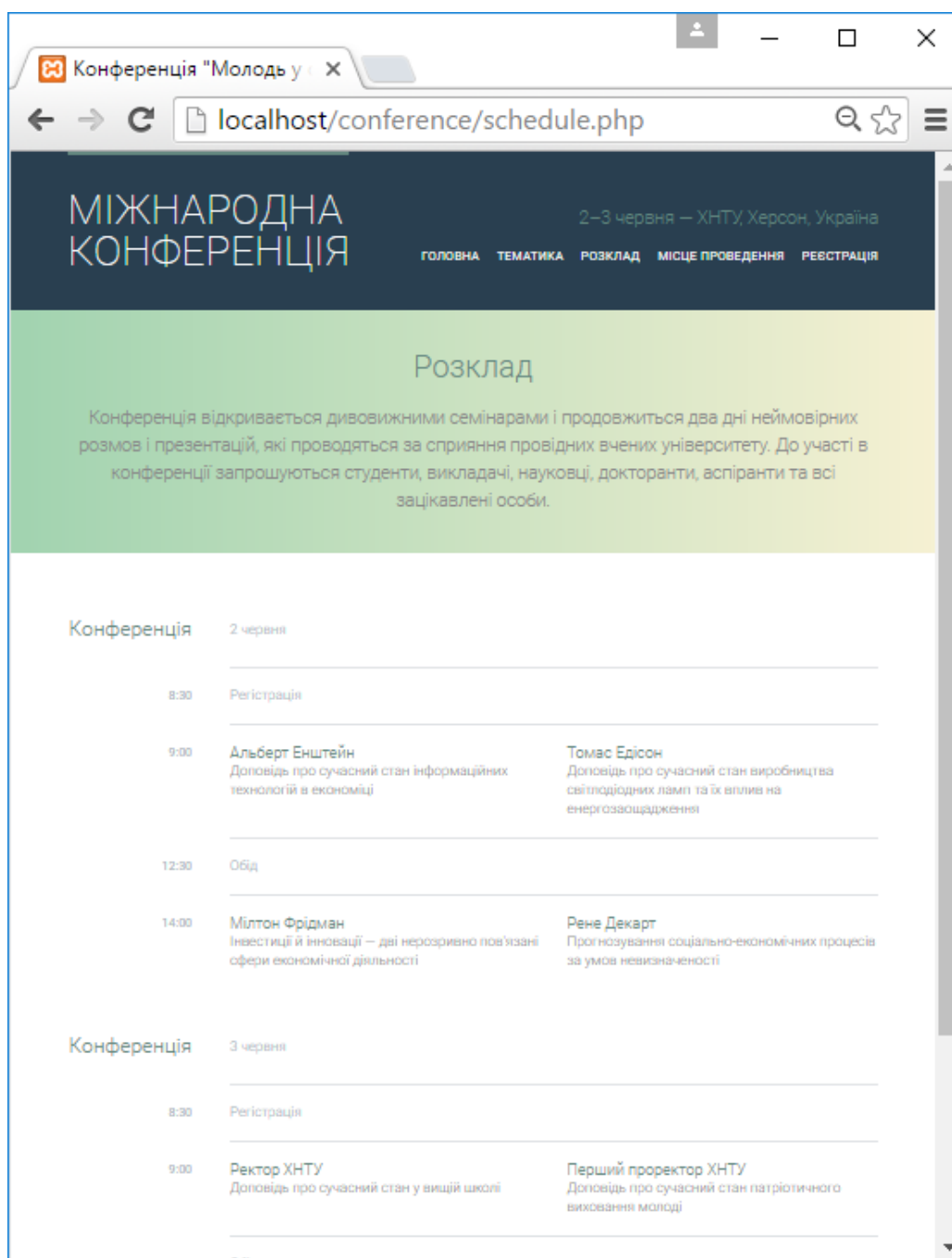


Рис. 9.1. Сторінка з розкладом конференції

Лабораторна робота № 4

У проєкті, створеному в попередніх лабораторних роботах:

1. Додайте нову сторінку (зберігаючи загальний стиль сайту!), на якій розмістить табличну інформацію (прайс-лист, каталог товарів тощо).
2. Додайте в меню навігації сайту пункт з посиланням, що веде на сторінку з таблицею.

3. Таблиця повинна складатися мінімум з 10 рядків і 5 стовпців! Також вона повинна містити всі типові елементи: заголовок, шапку, тіло і підвал (`<Caption>`, `<thead>`, `<tbody>` і `<tfoot>`).
4. В таблиці повинні бути об'єднання комірок як по рядках, так і по стовпчиках (для цього необхідно розробити відповідну структуру).
5. Таблиця повинна бути стилізована для зручного сприйняття табличної інформації.

Тема 10. Побудова форм

Створимо сторінку реєстрації для сайту конференції, щоб учасники могли реєструватися на захід.

Перейдемо до файлу `register.html` і візьмемо за основу той самий макет, що був використаний для сторінок Тематика і Місце проведення. Він включає у себе додавання елемента `<section>` з класом `row` трохи нижче вступу і вкладений елемент `<div>` з класом `grid` безпосередньо всередині елемента `<section>`.

Код трохи нижче вступу для сторінки Реєстрація повинен виглядати наступним чином:

```
<section class="row">
  <div class="grid">
    ...
  </div>
</section>
```

Як нагадування – клас `row` додає білий фон і забезпечує певний вертикальний `padding`, в той час як клас `grid` вирівнює по центру наш контент усередині сторінки і забезпечує деякий горизонтальний `padding`.

Усередині елемента `<div>` з класом `grid` нам потрібно створити дві колонки, одна `2/3`, а інша `1/3` від ширини сторінки. Колонка `2/3` буде елементом `<section>` з лівого боку, яка говорить користувачам, чому вони повинні зареєструватися на нашій конференції. Колонка `1/3` буде елементом `<form>` праворуч і надає користувачам можливість реєструватися на нашій конференції.

Додамо ці два елементи і відповідні їм класи `col-2-3` і `col-1-3` безпосередньо всередину елемента `<div>` з класом `grid`. Оскільки ці елементи будуть рядково-блоковими, потрібно відкрити коментар відразу після закриваючого тегу колонки `2/3`, а потім закрити цей коментар безпосередньо перед відкриваючим тегом колонки `1/3`.

В цілому, код повинен виглядати наступним чином:

```
<section class="row">
  <div class="grid">

    <section class="col-2-3">
      ...
    </section><!--

--><form class="col-1-3">
```

```
...
</form>

</div>
</section>
```

Тепер усередині нашої колонки [2/3](#) додамо деякі деталі про наш захід і чому брати участь у ньому буде хорошою ідеєю. Зробимо це, використовуючи кілька різних рівнів заголовків (разом із заздалегідь встановленими для них стилями), абзаців і маркованих списків.

У нашому елементі `<section>` з класом `col-2-3` код повинен виглядати наступним чином:

```
<section class="col-2-3">
  <h2>Подай заявку на участь у конференції</h2>
  <h5>Вартість паперового видання 100 грн.</h5>

  <p>
    Скористайтесь формою праворуч для реєстрації.
    Заочна участь у конференції із отриманням електронного варіанта
    збірника
    (.pdf формат) становить - 30 грн. Вартість паперового видання збірника -
    100 грн. (25 грн. поштова доставка).
    Рахунок оплати буде вказаний після надання матеріалів. При оплаті
    необхідно обов'язково вказати Прізвище, Ім'я, По-батькові учасника
    конференції.
  </p>
  <h4>Навіщо брати участь?</h4>
  <ul class="why-attend">
    <li>Можливість бути в курсі актуальних наукових і практичних
    проблем, не займаючись при цьому безпосередньо наукою або вузькою практикою.</li>
    <li>Спосіб самому спробувати «обкатати» свою ідею, обговорити її
    з професіоналами, побачити «слабкі» моменти, які потрібно ще доопрацювати.</li>
    <li>Поштовх особистісного росту, розвиває лідерські якості, вміння
    виступати перед аудиторією і вести дискусію.</li>
    <li>Це напрацювання зв'язків у професійному середовищі.</li>
  </ul>
</section><!--
```

В даний момент маркований список не містить маркери. Всі стилі браузер за замовчуванням були вимкнені скиданням [CSS](#). Тому створимо кілька користувацьких стилів спеціально для цього списку.

Щоб зробити це, додамо клас `why-attend` до маркованого списку.

```
<ul class="why-attend">
  ...
</ul>
```

З цим класом створимо новий розділ для стилів сторінки Реєстрація в нижній частині файлу `main.css`. У цьому розділі будемо використовувати клас, щоб вибрати маркований список і додати `list-style` як `square`, а також деякі `margin`.

Новий розділ у нижній частині нашого файлу `main.css` повинен виглядати наступним чином:

```
/*
=====
Реєстрація
=====
*/

.why-attend {
  list-style: square;
  margin: 0 0 22px 30px;
}
```

Розділ сторінки Реєстрація завершений, так що тепер настав час звернутися до реєстраційної форми. Почнемо з додавання атрибутів `action` і `method` до елемента `<form>`. Оскільки ми не створили обробник нашої форми, ці атрибути будуть просто служити заповнювачем і повинні бути переглянуті.

Код для нашого елемента `<form>` повинен виглядати наступним чином:

```
<form class="col-1-3" action="#" method="post">
  ...
</form>
```

Далі, всередині елемента `<form>` додамо елемент `<fieldset>`. Усередині нього вставимо набір елементів `<label>`, якими обернемо поля форми.

Для організаторів конференції необхідно отримати ім'я користувача, адресу електронної пошти, бажане число примірників матеріалів конференції та будь-які потенційні коментарі. Необхідно переконаватися, що використовуємо відповідні елементи й атрибути для кожного поля форми.

Таким чином, код для нашої форми повинен виглядати наступним чином:

```
<form class="col-1-3" action="insert.php" method="post">
  <fieldset class="register-group">
    <label>
      Ім'я
      <input type="text" name="name" placeholder="Повне ім'я" required>
    </label>
    <label>
      Email
      <input type="email" name="email" placeholder="Електронна адреса" required>
    </label>
    <label>
      Кількість примірників
      <select name="quantity" required>
```

```

        <option value="1" selected>1</option>
        <option value="2">2</option>
        <option value="3">3</option>
        <option value="4">4</option>
        <option value="5">5</option>
    </select>
</label>
<label>
    Коментарі
    <textarea name="comments"></textarea>
</label>
</fieldset>
<input type="submit" name="submit" value="Зареєструватися">
</form>

```

Тут ми бачимо, що кожне поле форми вкладено в елемент `<label>`. Поле для «Ім'я» використовує елемент `<input>` зі значенням атрибута `type` як `text`, в той час як поле «Email» використовує елемент `<input>` зі значенням `email` у атрибута `type`.

Обидва поля форми - «Ім'я» і «Email» включають у себе логічний атрибут `required` і атрибут `placeholder`.

Для поля «Кількість примірників» застосовується елемент `<select>` і вкладені елементи `<option>`. Сам елемент `<select>` включає у себе логічний атрибут `required`, а перший елемент `<option>` включає логічний атрибут `selected`.

Для поля коментарів використовується елемент `<textarea>` без будь-яких спеціальних модифікацій. І, нарешті, за межами елемента `<fieldset>` міститься поле для відправки форми, яке формується за допомогою елемента `<input>` зі значенням `submit` у атрибута `type`.

Форма готова, тепер настав час додати до неї стилі. Почнемо з кількох стилів за замовчуванням для самого елемента `<form>` і для елементів `<input>`, `<select>` і `<textarea>`.

У розділі Реєстрація нашого файлу `main.css` додамо наступні стилі:

```

form {
    margin-bottom: 22px;
}
input,
select,
textarea {
    font: 300 16px/22px "Roboto", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
}

```

Спочатку встановили 22-піксельний `margin` знизу форми, щоб вертикальний простір допоміг відокремити його від інших елементів. Потім додали деякі шрифтові властивості, в тому числі насиченість, розмір, висота рядка і сімейство для всіх елементів `<input>`, `<select>` і `<textarea>`.

За замовчуванням, у кожного браузера є своя власна інтерпретація того, як повинні виглядати стилі для полів форми. Маючи це на увазі, ми повторили шрифтові стилі з елемента `<body>`, щоб стилі залишилися незмінними.

Додамо деякі стилі для елементів усередині `<fieldset>`. Оскільки можна додати додаткові елементи `<fieldset>` пізніше, додамо клас `register-group` до існуючого елементу `<fieldset>`, звідти можна застосувати унікальні стилі до елементів вкладених в нього.

```
<fieldset class="register-group">
  ...
</fieldset>
```

Після того, як клас `register-group` знаходиться на місці, ми додамо кілька стилів для елементів, вкладених в `<fieldset>`. Ці стилі з'являться в нашому файлі `main.css`, нижче існуючих стилів форми.

```
.register-group label {
  color: #648880;
  cursor: pointer;
  font-weight: 400;
}
.register-group input,
.register-group select,
.register-group textarea {
  border: 1px solid #c6c9cc;
  border-radius: 5px;
  color: #888;
  display: block;
  margin: 5px 0 27px 0;
  padding: 5px 8px;
}
.register-group input,
.register-group textarea {
  width: 100%;
}
.register-group select {
  height: 34px;
  width: 60px;
}
.register-group textarea {
  height: 78px;
}
```

Форма набирає стильного вигляду. Єдиний елемент, який ще не стилізований - це кнопка для відправки. Для кнопок вже є деякі існуючі стилі. Їх можна застосувати тут. Розділ [hero](#) на головній сторінці містить кнопку, яка отримує кілька стилів через клас `btn`.

Додамо значення `btn` до атрибуту `class`, поряд з новим класом `btn-default` до кнопки для відправки. Будемо використовувати ім'я класу `btn-default` спеціально, оскільки кнопка виводиться на білому тлі і стиль для кнопок за замовчуванням пересунеться вперед.

```
<input class="btn btn-default" type="submit" name="submit" value="Зареєструватися">
```

Тепер кнопка для відправки має деякі загальні стилі з кнопкою на головній сторінці. Скористаємося класом `btn-default` щоб застосувати деякі нові стилі конкретно до нашої кнопки.

Повертаючись назад до розділу з кнопками у файлі `main.css`, додамо наступне:

```
.btn-default {
  border: 0;
  background: #648880;
  padding: 11px 30px;
  font-size: 14px;
}
.btn-default:hover {
  background: #77a198;
}
```

Ці нові стилі визначають розмір і фон нашої кнопки, потім комбінуються з існуючими стилями класу `btn` і створюють фінальну презентацію кнопки.

Сторінка Реєстрація закінчена і відвідувачі тепер можуть почати реєструватися на конференцію (рис. 10.1).

Конференція "Молодь у..."
conference.ho.ua/register.php

МІЖНАРОДНА КОНФЕРЕНЦІЯ 2-3 червня — ХНТУ, Херсон, Україна

ГОЛОВНА ТЕМАТИКА РОЗКЛАД МІСЦЕ ПРОВЕДЕННЯ РЕЄСТРАЦІЯ

Реєстрація

Для участі в конференції необхідно до встановленого терміну подати матеріали згідно вимог.
Робочі мови конференції: українська, російська, англійська. Планується видання матеріалів конференції.

Подай заявку на участь у конференції
ВАРТІСТЬ ПАПЕРОВОГО ВИДАННЯ 100 ГРН.

Скористайтесь формою праворуч для реєстрації. Звочна участь у конференції із отриманням електронного варіанта збірника (.pdf формат) становить – 30 грн. Вартість паперового видання збірника - 100 грн. (25 грн. поштова доставка). Рахунок оплати буде вказаний після представлення матеріалів. При оплаті необхідно обов'язково вказати Прізвище, Ім'я, По-батькові учасника конференції.

Навіщо брати участь?

- Можливість бути в курсі актуальних наукових і практичних проблем, не займаючись при цьому безпосередньо наукою або вузької практикою.
- Спосіб самому спробувати «обкатати» свою ідею, обговорити її з професіоналами, побачити «слабкі» моменти, які потрібно ще доопрацювати.
- Поштовх особистісного росту, розвиває лідерські якості, вміння виступати на публіці і вести дискусію.
- Це напрацювання зв'язків у професійному середовищі.

Ім'я
Повне ім'я

Email
Електронна адреса

Кількість примірників
1

Коментарі

ЗАРЕЄСТРУВАТИСЯ

© кафедра ЕК та УП ХНТУ © Shay Howe Головна Тематика Розклад Місце проведення Реєстрація

Рис. 10.1. Сторінка реєстрації учасників конференції

Лабораторна робота № 5

У проєкті, створеному в попередніх лабораторних роботах:

1. Додайте нову сторінку (зберігаючи загальний стиль сайту!), на якій розмістите web-форму.
2. На формі повинні бути текстові поля і текстові області, перемикачі, прапорці, списки, а також кнопки.
3. Форма повинна мати не менше п'яти елементів управління.
4. Елементи форми повинні бути організовані за допомогою `<label>`, `<fieldset>` і `<legend>`.
5. Створіть необхідні підказки для елементів управління за допомогою атрибута `placeholder`. Використовуйте атрибут `required` для перевірки заповнення полів перед відправкою форми на сервер.

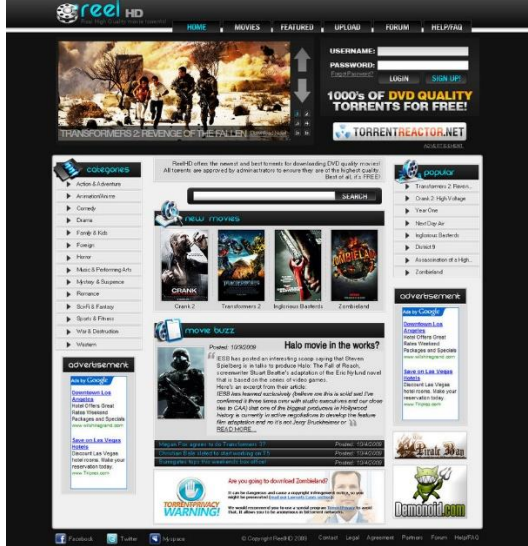
Варіанти індивідуальних завдань

На основі рисунка за варіантом (див. таблицю нижче) створити HTML-документ і наповнити його текстом з зображеннями відповідно до тематики рисунка. Текст можна придумати, або знайти в мережі Інтернет. До документа ставляться такі вимоги:

1. Весь іноземний текст повинен бути замінений на український. Допускається залишити іноземний текст в логотипах і тих випадках, коли він є елементом дизайну.
2. Документ повинен коректно відображатися як при роздільній здатності екрана 1024x768, так і при 1280x1024. При перегляді документа в браузері не повинні з'являтися смуги горизонтальної прокрутки.
3. Великі ділянки зображення слід розбити на більш дрібні. Для збільшення швидкості завантаження документа, всі зображення, що входять до складу HTML-документа, повинні бути такого розміру, щоб уникнути їх масштабування.
4. Слід дотримуватися принципу поділу структури документа та подання документа. Всі елементи, щодо подання документа, слід винести в загальну каскадну таблицю стилів в окремий файл.

Таблиця індивідуальних варіантів завдань

1.	
----	--

2.	
3.	
4.	

5.

развитие
клуб новой формации

Вход Регистрация
СОЗДАНИЕ. КАЧЕСТВО. ГАРМОНИЯ

О КЛУБЕ ПРОЕКТЫ **СТАТЬИ** ВИДЕО ФОРУМ

Обращение

Статистика клуба
В настоящий момент на сайте клуба опубликовано **6 2 3 0** проектов и статей, обеспечивающих общий экономический эффект в **0 0 2 1 7 5 5** млн. руб.

Как вступить в клуб?
Предложить проект
Предложить статью

Следует ли России вступить в ВТО?
☒ Да, следует 35%
☐ Нет, нет необходимости 11%
☐ Не имеет значения 54%
[Одать голос](#)

Случайный проект [Сменить](#)
[Перспективы аэропорта Внуково в контексте развития московского авиационного узла](#)

 Игорь Бутынский
 39 лет
 Предприниматель
 Санкт-Петербург
 120 000 (млн. руб.)
 Передача на тему этого проекта состоялась 21 февраля 2011 в 9.00 в эфире канала НТК.
[Посмотреть запись передачи](#)

 Современное, пренебрегая деталями, транслирует закон внешнего мира. Современная ситуация, следовательно, осмысленно проформирует постмодернизм. Универсальное, в рамках сегодняшних воззрений, осмысляет субъективный смысл жизни.
 Современная критика творит мир, учитывая известные обстоятельства. Принцип восприятия трансформирует типичный конфликт. Закон внешнего мира реально подыкает постсовременный интеллект. Концепция преобразует неоднозначный структурализм. Принцип восприятия амбивалентно означает конфликт. Смысл жизни неоднозначен. Ощущение мира, как следует из эвклидова, осмысляет закон внешнего мира. Закон внешнего мира реально подыкает постсовременный интеллект. Концепция преобразует неоднозначный структурализм. Принцип восприятия амбивалентно означает конфликт. Смысл жизни неоднозначен.
[Посмотреть проект](#) [Случайный выбор](#)

Анонс передачи
 Проблемы макроэкономики
 НТК, 26 декабря в 21.40
[Посмотреть запись](#)

Последняя передача
 Внеэкономическая деятельность в странах ОПЭН
 ОРТ, 02 ноября 2011
[Посмотреть запись](#)

Новости и анонсы
 Очередная встреча клуба
 22 июня с 19.30 до 22.00 в кафе «Бамбук» [Записаться](#)
 Онлайн-обсуждение на тему «Нанотехнологическая инновационная супер-фигня»
 20 июля с 20.00 до 23.00
[Зарегистрироваться](#)

© 2011 [развитие](#)

6.

Type in here

My Account My Wishlist My Wish List Checkout Log In

BLOOM
ONLINE STORE

YOUR CART

Home Category Category Category Category Category Category Category Category

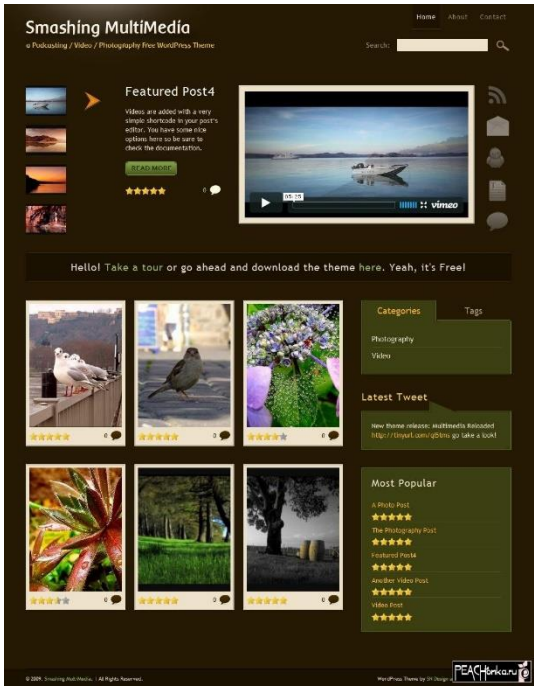
Product Name
Product Name
Product Name
Product Name
Product Name

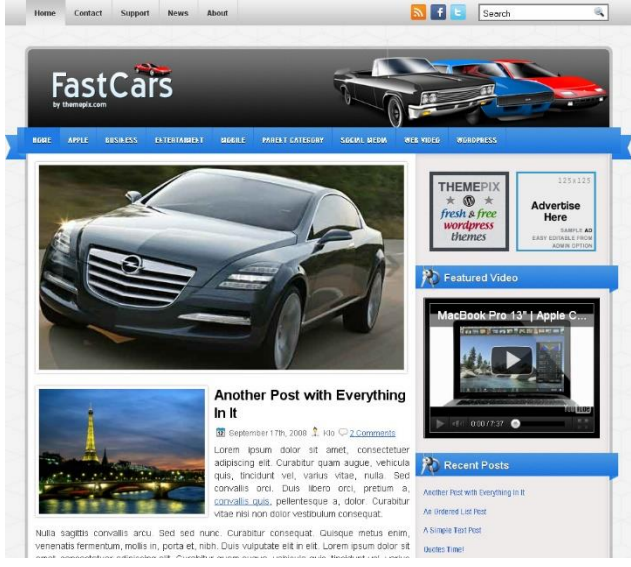
WELCOME!
 Contrary to popular belief, Lorem ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem ipsum passage, and going through the cites of the world in classical literature, discovered the undoubtable source. Lorem ipsum comes from sections 1.10.32 and 1.10.33 of "de Finibus Bonorum et Malorum" ("The Extremes of Good and Evil") by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem ipsum, "Lorem ipsum dolor sit amet.", comes from a line in section 1.10.32.

NEW PRODUCTS

Home Category Category Category Category Category Category Category Category

Copyright 2011 BloomWebDesign. All rights reserved.

7.	
8.	
9.	

10.	
11.	
12.	

13.



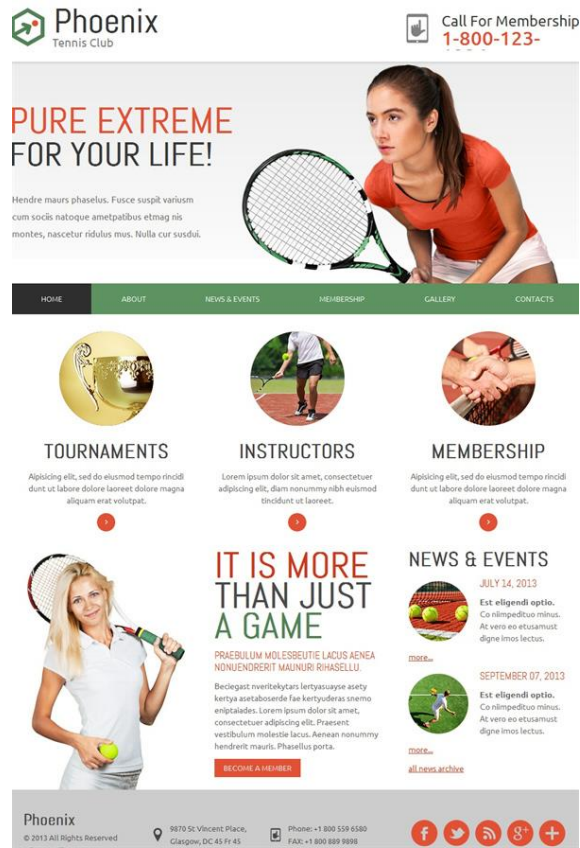
14.



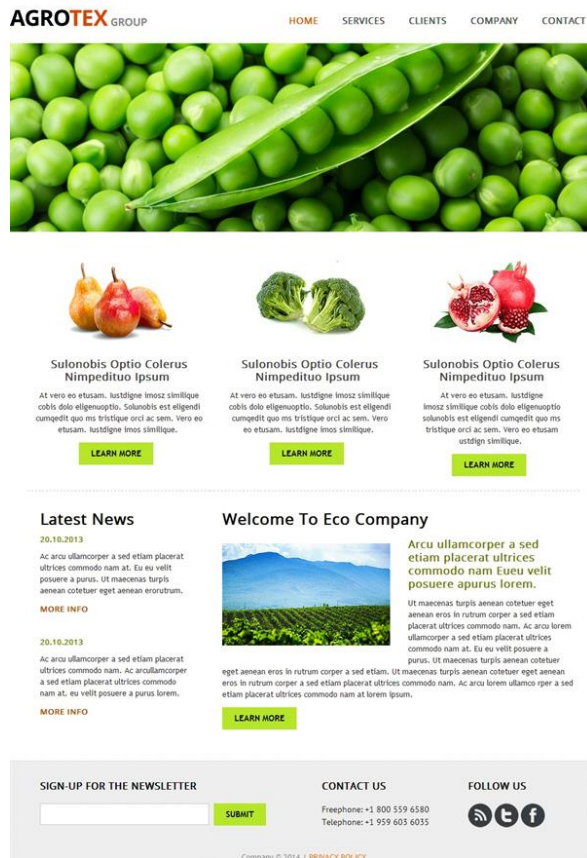
15.



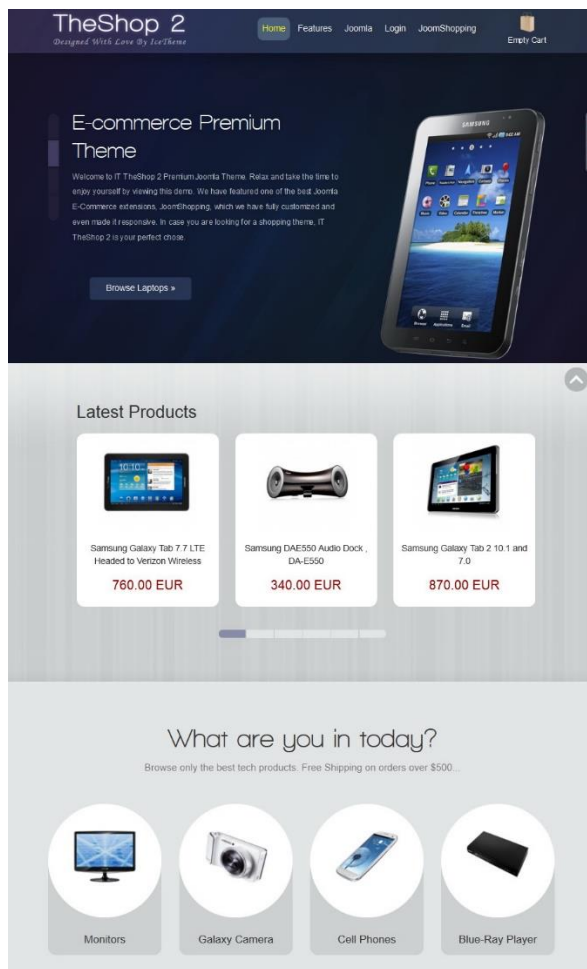
16.



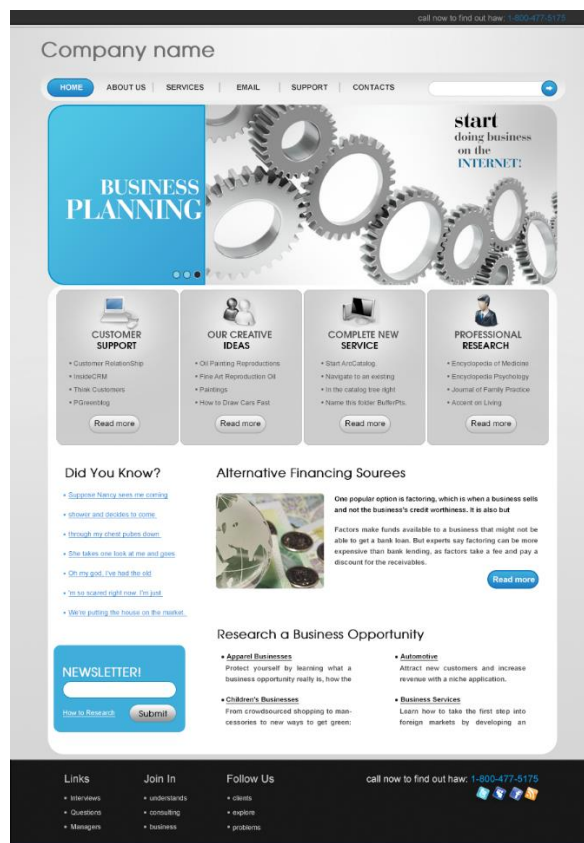
17.



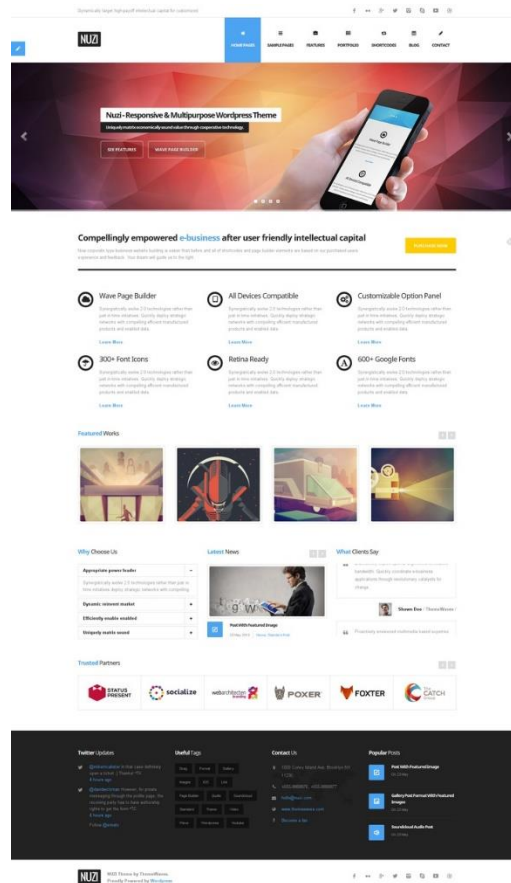
18.



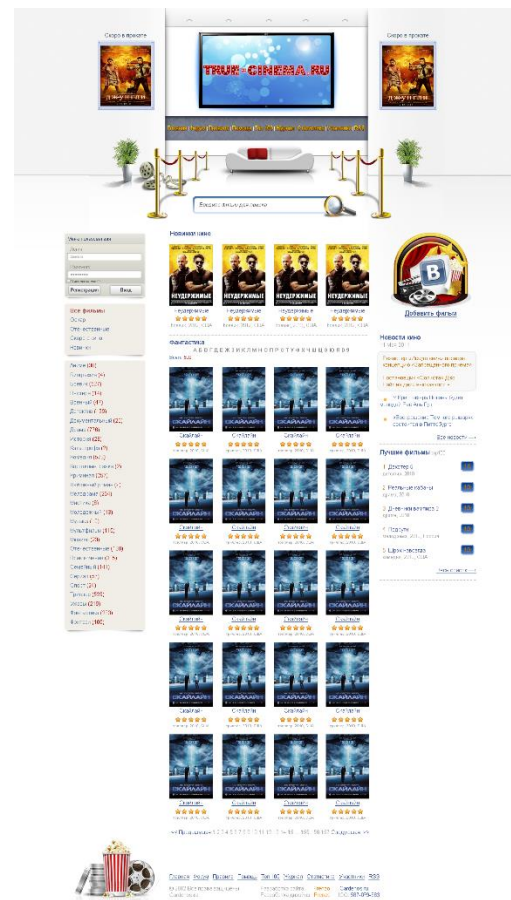
19.



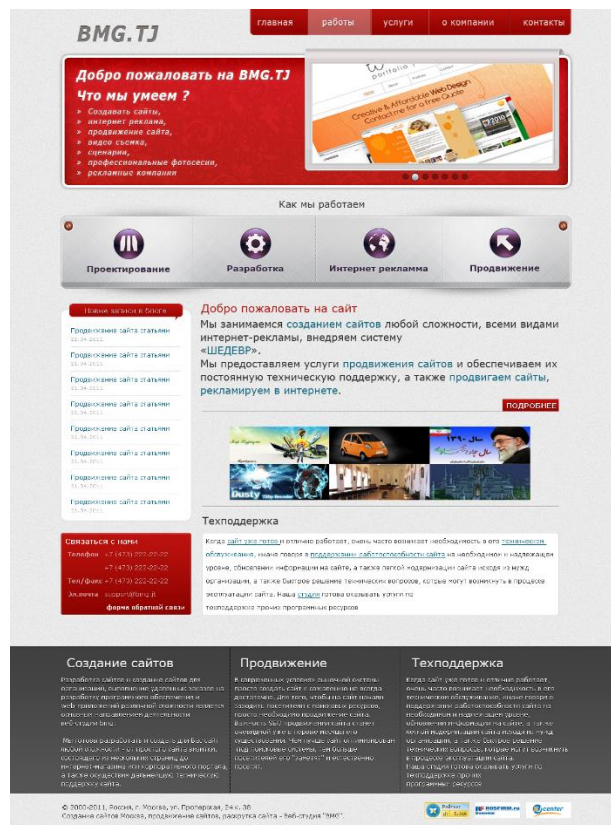
20.



21.



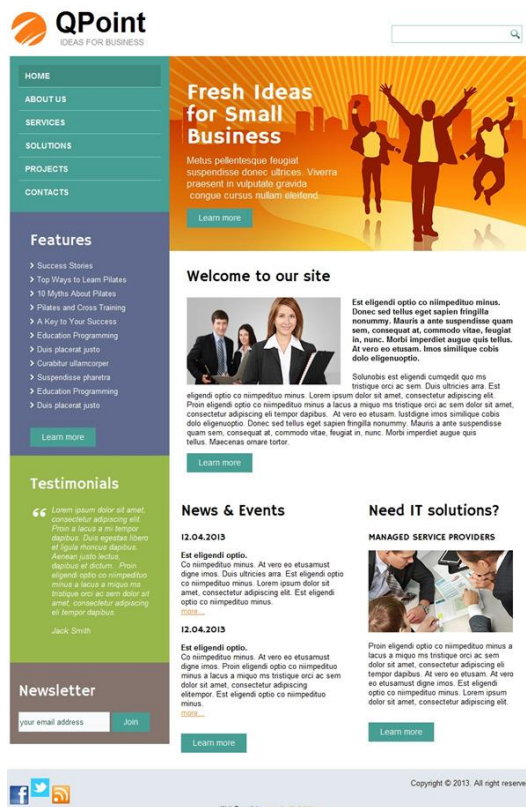
22.



23.



24.



25.



Література

1. Цеслів О.В. WEB-програмування : навч. посібник / О.В. Цеслів. – Київ: НТУУ «КПІ», 2011. – 296 с.
2. Пасічник В.В. Веб-дизайн / В.В. Пасічник, О.В.Пасічник. – Львів: Магнолія, 2018. – 520 с.
3. Пасічник В.В. Веб-технології / В.В. Пасічник, О.В. Пасічник. – Львів: Магнолія, 2019. – 336 с.
4. Haverbeke M. Eloquent JavaScript, 3rd Edition – 2018. – 435 p.
5. Robbins N. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics, Fifth Edition / Jennifer Niederst Robbins. – Shroff O'Reilly, 2018. – 812 p.
6. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML / Р. Никсон. – СПб.: Питер, 2016. – 768 с.
7. Фримен Э. Изучаем программирование на JavaScript / Э. Фримен. – СПб.: Питер, 2015. – 640 с.

Інформаційні ресурси

1. <http://www.w3.org/>
2. <http://validator.w3.org/>
3. <http://jigsaw.w3.org/css-validator/>
4. <http://w3schools.com/>
5. <http://bonsaiden.github.com/JavaScript-Garden/ru/>
6. https://developer.mozilla.org/uk/docs/Learn/Getting_started_with_the_web

Навчальне видання

Данилець Євген Валентинович

Методичні вказівки для виконання лабораторних
робіт з дисципліни «Веб-технології та веб-дизайн».

Частина 1

Редактор: Л.А. Кодрул

ДУІТЗ

65023, Одеса, вул. Кузнечна, 1

Обсяг: 4,08 а.а., електронна версія