МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

МЕТОДИЧНІ ВКАЗІВКИ

до виконання практичних робіт

«Основи веб-технологій»

з дисципліни «Основи веб-технологій»
для студентів 122 спеціальності «Комп’ютерні науки»

Затверджено
редакційно-видавничою
радою університету,
протокол № __ від __.__.2023 р.

Харків
НТУ «ХПІ»
2023

Методичні вказівки до виконання практичних робіт з дисципліни «Основи веб-технологій» для студентів 122 спеціальності «Комп’ютерні науки» / уклад. А.О. Лисенко, І.В. Шуба. – Харків : НТУ «ХПІ». 2023 – 33 с.

Укладачі: А.О. ЛИСЕНКО, І.В. ШУБА

Рецензент М.А. ГРИНЧЕНКО

Кафедра стратегічного управління

ЗМІСТ

ВСТУП

Метою викладання курсу є надання студентам знань про сучасні підходи до Web-програмування, засвоєння можливостей використання різноманітних технологій та фреймфорків як на стороні кліентської частини веб-додатку (HTML, CSS, JavaScript) так і на стороні серверу (Python, SQL) для програмування динамічних Web-сайтів і Web-інтерфейсів доступу до баз даних.

Вивчення курсу «Основи веб-технологій» вимагає цілеспрямованої роботи над вивченням спеціальної літератури, активної роботи на лекціях та практичних заняттях, самостійної роботи та виконання індивідуальних завдань.

Завдання вивчення дисципліни

Завданням курсу " Основи веб-технологій" є:

- ознайомлення студентів з основами Python;
- вивчення основ взаэмодіі з сервером баз даних;
- технології створення веб-сторінок, таких як HTML, CSS, шаблонізація;
- використання JavaScript, починаючи з функцій та обробки подій і закінчуючи доступом до об’єктної моделі документів.
- особливості використання AJAX викликів при реалізації динамічного інформаційного середовища;
- особливості завантаження на веб-сайт файлів та зображень та роботу з ними;
- технології захисту веб-додатків.

Загальні компетентності:

- ЗК2. Здатність застосовувати знання у практичних ситуаціях;
- ЗК3. Здатність до розуміння предметної області та професійної діяльності;
- ЗК6. Здатність вчитися і оволодівати сучасними знаннями;
- ЗК7. Здатність до пошуку, оброблення та узагальнення інформації з різних джерел;
- ЗК8. Здатність генерувати нові ідеї (креативність).
- ЗК9. Здатність працювати в команді..

Спеціальні (фахові, предметні) компетентності:

- СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

Результати вивчення дисципліни

У результаті вивчення навчальної дисципліни студент повинен набути наступні результати навчання:

1. *Знання*: мову гіпертекстової розмітки HTML, CSS стилі, JavaScript, технологію розміщення веб-сайту в мережевому середовищі, основи обслуговування WEB-серверів; принципи написання та використання CGI-сценаріїв; синтаксис, семантику операторів мови Python; правила етики при розміщенні інформації в Інтернет; базові принципи веб-дизайну.
2. *Уміння*: використовувати можливості мови HTML для створення Web-сторінок; використовувати можливості технології CSS для створення Web сторінок; розробляти інформаційні ресурси в середовищі Web за допомогою мов програмування JavaScript та Python; розробляти інтерактивні Web-сторінки для Internet та Intranet мереж; використовувати сучасні засоби графічного моделювання та дизайну для проектування WEB-сторінок; модифікувати та розробляти модулі та компоненти для популярних CMS.
3. *Комунікація*: презентувати, обговорювати та захищати власні погляди в усній і письмовій формах та за допомогою інформаційно-комунікаційних технологій.
4. *Автономність та відповідальність*: усвідомлювати соціальну значущість майбутньої професії, необхідність подальшого навчання, вивчення, аналізу, узагальнення та поширення передового педагогічного досвіду, систематично підвищувати свою професійну кваліфікацію

В ході вивчання матеріалів курсу та виконання практичних (лабораторних) завдань студенти отримають теоретичні та практичні навички розробки веб-орієнтованих програмних систем. Результатом навчання буде набуття наступних компетентностей:

РН9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

(розподіл навчального часу за семестрами та видами навчальних занять)

Семестр	Загальний обсяг (годин) / кредитів ECTS	З них		За видами аудиторних занять (годин)			Індивідуальні завдання студентів (КП, КР, РГ, Р, РЕ)	Поточний контроль	Семестровий контроль	
		Аудиторні заняття (годин)	Самостійна робота (годин)	Лекції	Лабораторні заняття	Практичні заняття, семінари		Контрольні роботи (кількість робіт)	Залік	Іспит
1	2	3	4	5	6	7	8	9	10	11
3	**120 /4**	**80**	**40**	**32**	**32**	**16**		**2**		**V**

Співвідношення кількості годин аудиторних занять до загального обсягу складає 30 %.

***Критерії та система оцінювання знань та вмінь студентів*.**

Згідно основних положень ЄКТС, під **системою оцінювання** слід розуміти сукупність методів (письмові, усні і практичні тести, екзамени, проекти, тощо), що використовуються при оцінюванні досягнень особами, що навчаються, очікуваних результатів навчання.

Успішне оцінювання результатів навчання є передумовою присвоєння кредитів особі, що навчається. Тому твердження про результати вивчення компонентів програм завжди повинні супроводжуватися зрозумілими та відповідними **критеріями оцінювання** для присвоєння кредитів. Це дає можливість стверджувати, чи отримала особа, що навчається, необхідні знання, розуміння, компетенції.

Критерії оцінювання – це описи того, що як очікується, має зробити особа, яка навчається, щоб продемонструвати досягнення результату навчання.

Основними концептуальними положеннями системи оцінювання знань та вмінь студентів є:

1. Підвищення якості підготовки і конкурентоспроможності фахівців за рахунок стимулювання самостійної та систематичної роботи студентів протягом навчального семестру, встановлення постійного зворотного зв’язку викладачів з

кожним студентом та своєчасного коригування його навчальної діяльності.

2. Підвищення об’єктивності оцінювання знань студентів відбувається за рахунок контролю протягом семестру із використанням 100 бальної шкали (табл. 2). Оцінки обов’язково переводять у національну шкалу (з виставленням державної семестрової оцінки «відмінно», «добре», «задовільно» чи «незадовільно») та у шкалу ECTS (A, B, C, D, E, FX, F).

Таблиця - Шкала оцінювання знань та умінь: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
90-100	A	відмінно
82-89	B	добре
75-81	C	
64-74	D	задовільно
60-63	E	
35-59	FX	незадовільно з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни

ПРАКТИЧНА РОБОТА №1 (2 години)

«Структура Веб сайту та авторсько-правові аспекти при створені сайту»

Мета: Познайомитись із структурою та специфікою розробки та функціонування веб. Розглянути типи сайтів їх структуру та реалізацію. Розглянути основні питання авторського права, які виникають при розробці сайту.

Теоретичні відомості

Веб (веб-сайти та веб-додатки) мають свою структуру та принципи функціонування, які допомагають забезпечити ефективну та зручну інтернет-присутність. Основні складові та принципи веб-систем:

1) Клієнт-серверна модель.

Один з основних принципів веб-архітектури - це клієнт-серверна модель. За цим принципом веб-додатки поділяються на дві основні компоненти: клієнт та сервер. Клієнт - це програмне забезпечення, яке запускається на боку користувача (наприклад, браузер), і він забезпечує інтерфейс для взаємодії з користувачем. Сервер - це програмне забезпечення, яке запускається на боку сервера та обробляє запити від клієнта, здійснює доступ до баз даних, логіку додатку та повертає результати до клієнта.

2) HTTP протокол.

Для обміну даними між клієнтом та сервером використовується протокол HTTP (Hypertext Transfer Protocol). HTTP забезпечує стандартизований механізм для взаємодії з веб-серверами, включаючи відправку запитів (GET, POST, PUT, DELETE тощо) та отримання відповідей (HTML сторінки, JSON дані тощо).

3) HTML, CSS та JavaScript.

HTML (Hypertext Markup Language), CSS (Cascading Style Sheets) та JavaScript - це основні технології для створення клієнтської сторони веб-додатків. HTML використовується для створення структури веб-сторінок, CSS - для стилізації та оформлення сторінок, а JavaScript - для динамічного управління поведінкою сторінок, інтерактивністю та взаємодією з сервером.

4) Бази даних.

Веб-додатки часто використовують бази даних для зберігання, організації та отримання даних. Серверна частина веб-додатків взаємодіє з базою даних для збереження та зчитування інформації.

Клієнтську частину часто називають frontend частиною, а серверну частину веб-додатку – backend. Однак необхідно зазначити, що серверна частина теж може поділятися на складові: наприклад, веб-сервер, який обслуговує статичний

контент та сервер додатків, який обслуговує бізнес-логіку. В такому разі веб-сервер може також називатися frontend-сервером, а сервер додатків – backend-сервером або application-сервером.

Інтерактивність та взаємодія:

Одні з основних переваг веб-додатків полягають у їхній здатності взаємодіяти з користувачами. Веб-додатки можуть реагувати на взаємодію користувача, виконувати дії на стороні клієнта та сервера та оновлювати інформацію без необхідності перезавантаження сторінки.

Загалом, веб-продукти мають складну структуру, що базується на взаємодії між клієнтами та серверами, використанні стандартних протоколів та технологій та можливості взаємодії та динамічного оновлення інформації.

Види сайтів:

1. Сайт-візитівка.
2. Комерційні сайти:
   а) сайт-вітрина
   б) промо-сайти
   в) інтернет-магазин
3. Інформаційні сайти:
   а) тематичні інформаційні сайти
   б) новинні сайти
   в) блоги
4. Освітні ресурси:
   а) каталоги освітніх ресурсів
   б) платформи для он-лайн курсів
   в) електронні підручники
   г) он-лайн словники, енциклопедії
   д) сервіси для створення презентацій, інфографіки тощо.
5. Соціальні проєкти (блоги, соціальні мережі).
6. Веб-сервіси:
   а) каталоги
   б) пошукові системи
   в) поштові сервіси
   г) сайт хостинг
   д) дошки оголошень

Розробка веб-сайту може бути складним процесом з використанням різних інструментів, який включає кілька етапів. Нижче наведені загальні етапи розробки веб-сайту:

1) Перший етап - це збір і аналіз вимог до веб-сайту. Це означає розуміння цілей сайту, цільової аудиторії, функціональних вимог, дизайну, технічних вимог, бюджету та графіку розробки.
2) Планування. На цьому етапі створюється детальний план розробки, включаючи визначення технологій, вибір платформи, структуру сторінок, складність функціоналу, терміни виконання і розподіл обов'язків між командою розробників.
3) Дизайн веб-сайту охоплює створення макетів, вибір кольорів, шрифтів, графіки та інтерфейсу. Важливо, щоб дизайн був орієнтованим на користувача і відповідав бренду або цілі компанії.
4) Розробка. На цьому етапі розробляються функціональність та динамічні елементи веб-сайту. Програмісти працюють зі збереженням даних, створенням серверної та клієнтської частин, інтеграцією з базами даних та іншими сторонніми системами.
5) Тестування. Після завершення розробки веб-сайту відбувається тестування. Веб-сайт перевіряється на різних пристроях та браузерах, щоб впевнитися, що він працює належним чином і є безпечним. Також виконуються тестування з використанням різних сценаріїв, щоб переконатися, що функціонал працює належним чином.
6) Випробування та оптимізація продуктивності. Забезпечення швидкої роботи веб-сайту і мінімізація часу завантаження сторінок є важливим аспектом. Перевіряються продуктивність, оптимізуються файли, кешування, використовуються CDN (мережі доставки контенту) для покращення швидкості сайту.
7) Реліз та розгортання. Після успішного завершення тестування сайт готовий до релізу. Розгортання означає розміщення сайту на живому сервері і доступність його для відвідувачів.
8) Підтримка та оновлення. Після запуску сайту команда займається підтримкою, виправленням помилок та оновленнями для забезпечення безперебійної роботи та покращення функціональності.

Ці етапи розробки можуть варіюватися в залежності від розміру проекту, специфікацій та методологій роботи команди.

Макет сайту - це візуальне представлення структури, розташування елементів, графіки та інших компонентів веб-сайту перед його фактичним

створенням. Це відображення того, як буде виглядати кожна сторінка сайту, які елементи будуть присутні на ній та як вони будуть взаємодіяти один з одним.

Рис. 1 – Приклад макетів сайту

Авторсько-правові аспекти при створені веб-сайту.

Згідно чинного законодавства України кожен громадянин має право на вільну розробку та захист результатів своєї інтелектуальної діяльності, які не суперечать нормам моралі, принципам гуманності тощо.

Авторське право є частиною інтелектуальної власності і змісту права та захисту авторів або творців оригінальних творів. Це правовий механізм, який надає авторам контроль за використанням їх творів і дозволяє їм отримувати вигоди від комерційного, публічного чи іншого використання цих творів.

Авторське право захищає особисті немайнові та майнові права власників авторського права. Будь-яке використання чужого об’єкту авторського права дозволяється лише з дозволу правовласника, якщо термін майнових прав не сплив (згідно законодавства України цей термін для більшості об’єктів авторського права складає життя автора та 70 років після його смерті) так реалізуються майнові права. Та у разі використання обов’язкова вказівка імені (псевдоніму, ніку тощо) автора – реалізація особистих немайнових прав – бути вказаним як автор твору.

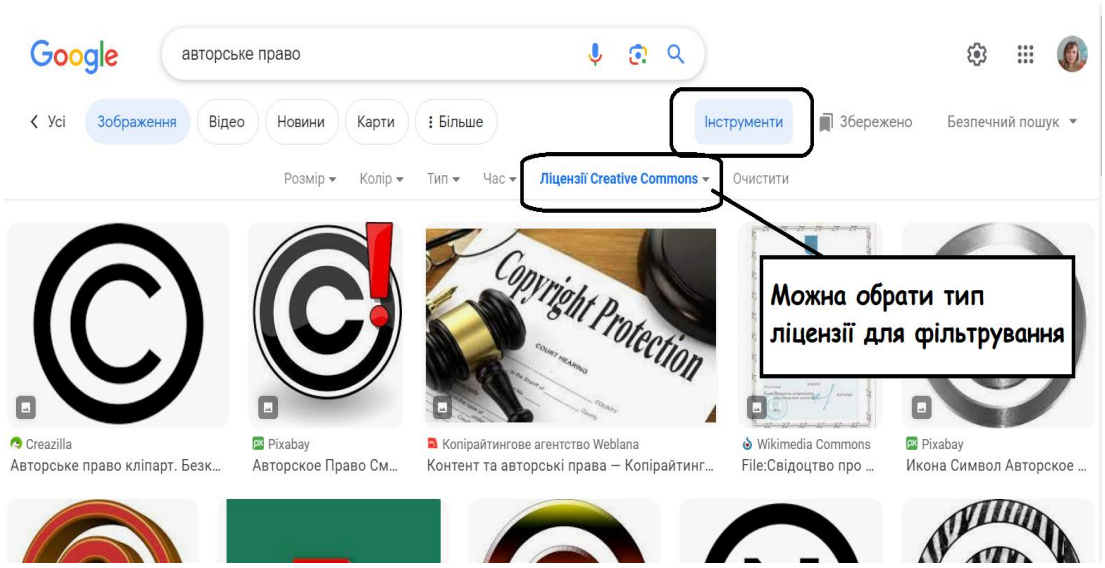
Найчастіше при створені та підтримані веб-сайту розробники стикаються з наступними об’єктами авторського права:

1) комп’ютерні програми;
2) графіка, рисунки (растрові, векторні);
3) фотографічні твори;
4) шрифти
5) музичний контент;
6) текстовий контент.

Загалом всі об’єкти інтелектуальної власності, які можуть бути у веб-сайті представлені на рисунку 1.

Рис. 2 – Об’єкти інтелектуальної власності у веб-сайті[1]

Для того щоб використовувати той чи інший об’єкт авторського права потрібно отримати дозвіл – ліцензію. Суть та обсяг дозволених дій можу бути різними, наприклад є ліцензії поширені в цифровому середовищі Creative Common[2], деякі можуть дозволяти некомерційне використання. У такому випадку використовувати фото, музику, текст тощо, можна лише для власних цілей, і якщо продавати продукт, який містить такий об’єкт авторського права то дії буде визнано порушенням, а значить і відповідним покаранням.



[1] Переклад з матеріалів сайт ВОІВ https://www.wipo.int/portal/en/index.html
[2] На законодавчому рівні в Україні вони не закріплені, проте вже багато років виконують регулятивну функцію в мережі Інтернет.

Рис.3 – Пошук зображень у пошуковій системі Google з урахуванням ліцензій

Рис.4 – Види ліцензій Creative Common

Шрифти захищаються як об'єкти авторського права. Є ресурси де можна взяти безкоштовні шрифти. Наприклад, myfonts (з пошуком "free"), uk.allfont.net, ux.pub[3] тощо.

Рис. 5 – Приклад українських безкоштовних шрифтів

Завдання

1. Згідно обраної тематики розробки веб-сайту розробити його структуру у вигляді макету. Обрати стиль.
2. Знайти у мережі Інтернет контент для подальшого наповнення сайту:
   - ✓ текстові блоки з відповідними посиланнями;

[3] https://ux.pub/varhal/dobirka-biezkoshtovnikh-ukrayinskikh-shriftiv-vid-ukrayinskikh-dizainieriv-3g7j

- ✓ не менше 10 рисунків та/або фотографій з вільною ліцензією;
- ✓ безкоштовну музику та відео;
- ✓ обрати три безкоштовні шрифти для використання у подальшій роботі.

ПРАКТИЧНА РОБОТА №2 (2 години)

«Створення таблиць в HTML. Grid та flex верстка»

Мета: Познайомитись із синтаксисом, основними тегами й атрибутами мови html для створення таблиць. Розглянути grid та flex верстку сторінок сайту.

Теоретичні відомості

HTML (англ. HyperText Markup Language) – стандартна мова розмітки гіпертекстових документів (веб-сторінок) в Інтернеті. Мова HTML інтерпретується браузерами. Отриманий в результаті інтерпретації відформатований текст відображається на екрані монітора комп'ютера або мобільного пристрою.

Таблиця — це представлення даних у вигляді стовпчиків та рядків. Таблиці в HTML використовують для організації та відображення числової інформації, статистики, розкладів, списків товарів, цін і багато іншого. Їх структурований вигляд виглядає легше сприймати дані. Також, на початкових етапах розвитку веб-технологій макети сторінок робилися за допомогою таблиць.

Теги для таблиць:

<*table*> — визначає таблицю
<*tr*> — визначає рядок таблиці
<*td*> — визначає комірку таблиці
<tr> — представляє рядки
<td> — використовують для створення комірок
<th> — використовують для додавання заголовків
<caption> — використовують для додавання підпису
<thead> — додає окремий заголовок до таблиці
<tbody> — показує головне тіло таблиці
<tfoot> — створює окремий колонтитул таблиці

Таблиці можуть мати різний вигляд і складність. В програмах MS Office або Excel можна поєднувати комірки, для цього їх виділити та натисути на команду «об’єднати». Також функції можна застосувати до HTML таблиці, використавши два атрибути: <colspan> для горизонтального об’єднання та <rowspan> для вертикального об’єднання. Атрибутам присвоюються числа, більші за нуль, що позначають кількість комірок, які потрібно об’єднати.

Об'єднання по горизонталі (***colspan***)

```
<table>
  <tr>
    <th>Заголовок 1</th>
    <th colspan="2">Заголовок 2 і 3</th>
  </tr>
  <tr>
    <td>Рядок 1, Колонка 1</td>
    <td>Рядок 1, Колонка 2</td>
    <td>Рядок 1, Колонка 3</td>
  </tr>
</table>
```

Об'єднання по вертикалі **(rowspan)**

```
  <table>
    <tr>
      <th>Заголовок 1</th>
      <th>Заголовок 2</th>
      <th>Заголовок 3</th>
    </tr>
    <tr>
      <td rowspan="2">Рядок 1, Колонка 1</td>
      <td>Рядок 1, Колонка 2</td>
      <td>Рядок 1, Колонка 3</td>
    </tr>
    <tr>
      <td>Рядок 2, Колонка 2</td>
      <td>Рядок 2, Колонка 3</td>
    </tr>
  </table>
```

Вибір між сіткою CSS і flexbox може бути трохи складним (іноді), особливо якщо ви новачок в CSS. Ось кілька початкових питань, які я ставлю собі, обираючи між ними:

- ✓ як відображаються дочірні елементи компонента? Як вбудовані або як стовпці та рядки?
- ✓ як компонент має працювати на екранах різних розмірів?

У більшості випадків, якщо у компонента, що переглядається, всі його дочірні елементи відображаються вбудованими, то, швидше за все, тут найбільш підходящим рішенням є flexbox

Flexbox — це одномерні макети, а Grid — двомірні (рис.5)

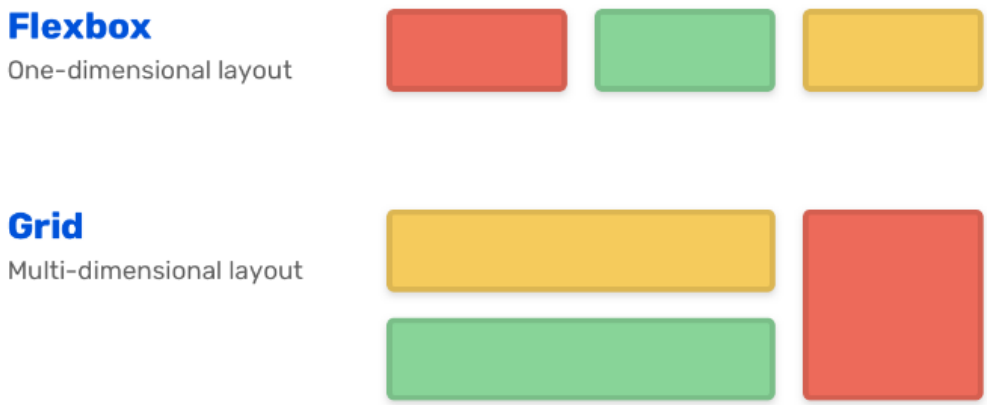


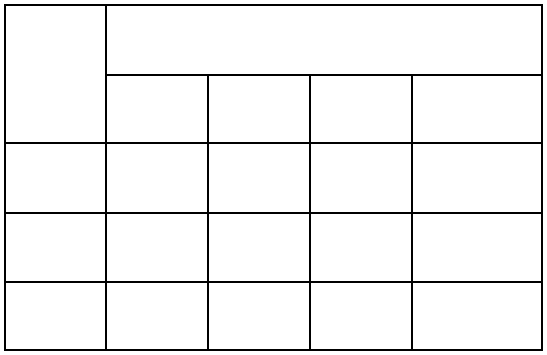
Рис.5

Існують варіанти коли варто поєднати CSS Grid та Flexbox. Кожен модуль макета не тільки має свої варіанти використання, але можна використовувати обидва. Наприклад, список карток. Сітка використовується для розміщення карток, а flexbox – для самого компонента картки.

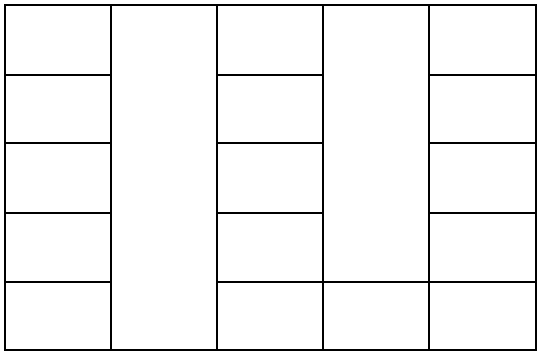
Рис.6 – Макет списку карток на сайті

Завдання

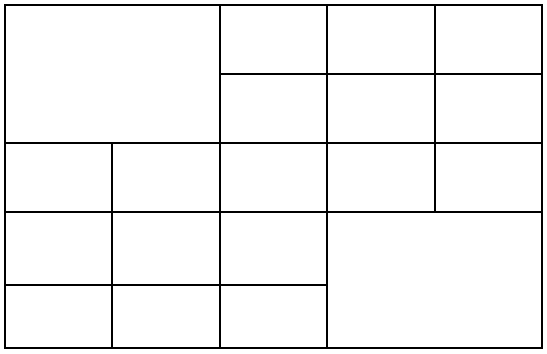
1) Написати код для побудови таблиць різної складності:



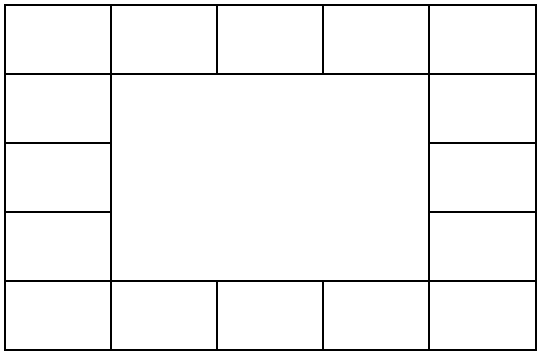
а)



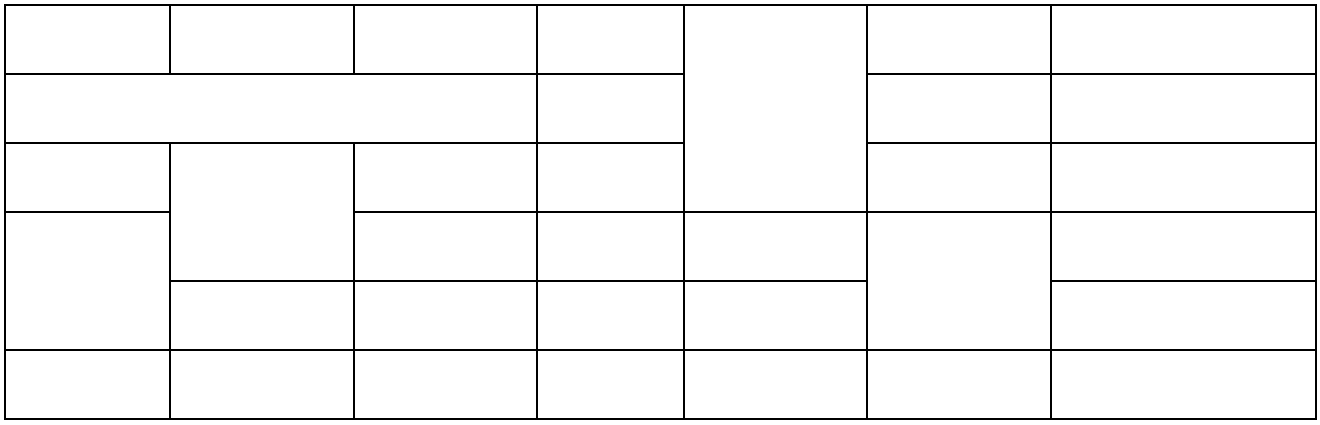
б)



в)



г)



д)

2. Реалізувати у сайті, який розробляєте на лабораторних роботах (на одній із сторінок) список карток як показано на рис.7, використовуючи Grid та Flexbox для розмітки.

Рис.7 – Макет розміщення карток на сторінці

ПРАКТИЧНА РОБОТА №3 (2 години)

«Використання тегів <area> та <map> для створення он-лайн квесту»

Мета: Познайомитись з особливостями використання та варіантами використання тегів <area> та <map>.

Теоретична частина

Тег <map> служить контейнером для елементів <area>, які визначають активні області карт-зображень. Такі області встановлюють **невидимі зони** зображення (рис.8) , які є посиланнями на HTML-документи. Мета використання тегу <map> - у зв'язуванні тега <img> з клієнтською картою-зображенням.

Рис.8 – Приклад форм невидимих зон

Приклад ігрового поля[4] та його елементів «Поліцейська дільниця» наведено на рис.9а де потрібно, виконавши всі завдання викрити вбивцю. На рисунку 9б показано жовтим кольором обрані невидимі поля, натискаючи на які користувач переходе на сторінку з завданням[5] рис.9г. На сторінці завдання обов'язково має бути кнопка «повернутися» або автоматичне повернення на вихідну сторінку після виконання завдання.

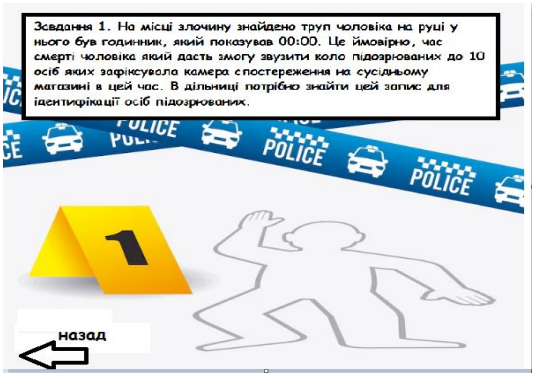
а)

б)

[4] Атрибуція фонового зображення Изображение от macrovector на Freepik

[5] Атрибуція фонового зображення Изображение от studiogstock на Freepik

в)



г)

Рис.9 – Приклад сторінок он-лайн квесту

Для визначення активних областей (координати, форма, розміри тощо) карти-зображення використовується тег <area>

Нижче наведено синтаксис використання тегу <map>

```
<img src="image.jpg" alt="Карта" usemap="#mapid">
<map name="mapid">
  <area shape="rect" coords="0,0,100,100" href="link1.html" alt="Зона 1">
  <area shape="circle" coords="150,150,50" href="link2.html" alt="Зона 2">
  <area shape="poly" coords="250,250,300,200,350,250" href="link3.html"
alt="Зона 3">
</map>
```

Завдання

Створити он-лайн квест, гру, екскурсію тощо з використанням тегів <area> та <map> при чому області <map> повині мати різні форми: коло, багатокутник, квадрат. Додати гру до вашого сайту, який ви розробляєте на лабораторних роботах.

ПРАКТИЧНА РОБОТА №4 (2 години)

«Робота з масивами в Java Script»

Мета: Познайомитися з особливостями роботи з масивами в Java Script.

Теоретична частина

Масив – це спеціальний вид об’єктів, створений для зберігання та обробки впорядкованих елементів. Квадратні дужки використовують для доступу до властивості arr[0], що в свою чергу прийшло з синтаксису об’єктів. Це теж саме, що доступ до властивості об’єкта obj[key], де arr це об’єкт в якому числа використовуються як ключі.

Масиви розширюють функціональність об’єкта тим, що надають можливість працювати з упорядкованими колекціями даних, а також надають доступ до властивості length. Але в основі це досі об’єкт.

Оголошення массиву:

```
// квадратні дужки (як правило)
let arr = [item1, item2...];
// new Array (набагато рідше)
let arr = new Array(item1, item2...);
```

Виклик new Array(number) створює масив з заданою довжиною, але без елементів.

Властивість length демонструє довжину масиву або, якщо точніше, останній цифровий індекс масиву плюс один. Це виконується автоматично методами масиву.

Якщо вручну скорочувати length, масив зменшується причому, ця операція незворотня.

Отримання елементів:

1) можна отримати елемент за його індексом, arr[0]
2) використати метод at(i), який допускає негативні індекси. Для негативних значень і він відступає від кінця масиву. Якщо і >= 0, він працює так само, як arr[i].

Ми можемо використовувати масив як двосторонню чергу за допомогою наступних операцій:

push(...items) додає items в кінець масиву.

pop() видаляє елемент з кінця масиву та повертає його.

shift() видаляє елемент з початку масиву та повертає його.

unshift(...items) додає items в початок масиву.

Для того, щоб пройтись циклом по елементах масиву:

for (let i=0; i<arr.length; i++) — працює швидше, сумісний зі старими браузерами.

for (let item of arr) — новий синтаксис, лише для значень елементів (відсутній доступ до індексів масиву).

Для порівняння масивів використовується цикл for..of, щоб порівнювати елемент за елементом.

Приклад коду простого рандомайзеру для вибору числа від 1 до 100 наведено нижче.

```
function getRandomNumber(min, max) {
  return Math.floor(Math.random() * (max - min + 1)) + min;
}

const minNumber = 1; // Мінімальне число в діапазоні
const maxNumber = 100; // Максимальне число в діапазоні

const randomValue = getRandomNumber(minNumber, maxNumber);
console.log(`Випадкове число: ${randomValue}`);
```

Завдання

Реалізувати скрипт рандомайзеру та додати його до вашого веб-сайту, який ви розробляєте протягом семестру. На сторінці має випадково з заданим інтервалом виводитися на екран інформація, наприклад, вислів відомої особистості, передбачення, анекдот тощо.

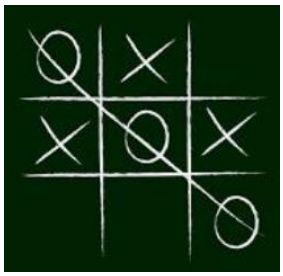
ПРАКТИЧНА РОБОТА №5 (4 години)

«Написання гри мовою JS»

Мета: Застосувати отримані знання та написати код гри мовою Java Script.

Теоретична частина

Мова програмування Java Script може використовуватися для написання інтерактивних елементів, наприклад ігор.



Гра «Хрестики-нулики» - це гра на папері (в докомп'ютерні часи) для двох гравців. На кожному ході гравці мають ставити **О** чи **Х** на ґратці розміром 3 на 3. Гравець, який першим поставив три однакових знаки в горизонтальному, вертикальному чи діагональному ряду, виграє партію.

Алгоритм написання коду гри:

1) створити дошку (майбутнє ігрове поле) за допомогою 2-вимірного масиву та ініціалізувати кожен масиву елемент як порожній;
2) зобразити порожнє за допомогою будь-якого символу. Наприклад, використовувати дефіс;
3) написати функцію для перевірки заповненості хрестиками та нуликами ігрового поля ;
4) подальша ітерація по дошці з умовами: якщо місце порожнє повертається false в іншому випадку повертається true;
5) написати функцію для перевірки, виграв гравець чи ні. Тобто, перевірити всі можливості перевірте всі рядки, стовпці та дві діагоналі.
6) Написати функції для:
   - ✓ показу дошки;
   - ✓ запуску гри (перший хід гравця випадковим .
7) Написати нескінченний цикл, який переривається, коли гра закінчується (або перемога, або нічия).
   - ✓ показати дошку користувачеві, для вибору місця для наступного ходу;
   - ✓ гравець вводить номер рядка та стовпця (обране місце для ходу);
   - ✓ оновити інформацію на дошці, відобразивши відповідний знак (хрестик або нулик) гравця, який робив хід;
   - ✓ перевірити, чи виграв гру поточний гравець чи ні. Якщо поточний гравець виграв гру, надрукувати повідомлення про виграш і розірвати нескінченний цикл.

8) Перевірити, заповнена дошка чи ні. Якщо поле заповнене, роздрукувати повідомлення про «нічию» і розірвати нескінченний цикл.
9) Показати користувачеві остаточний вигляд дошки.

Приклад коду гри «Хрестики-нулики» представлено нижче.

```
<!DOCTYPE html>
<html>
<head>
 <style>
  .board {
   display: grid;
   grid-template-columns: repeat(3, 100px);
   grid-gap: 2px;
  }
  .cell {
   width: 100px;
   height: 100px;
   border: 1px solid #000;
   text-align: center;
   font-size: 2em;
   cursor: pointer;
  }
 </style>
</head>
<body>
 <div class="board" id="board">
  <div class="cell" onclick="makeMove(this)"></div>
  <div class="cell" onclick="makeMove(this)"></div>
  <div class="cell" onclick="makeMove(this)"></div>
  <div class="cell" onclick="makeMove(this)"></div>
  <div class="cell" onclick="makeMove(this)"></div>
  <div class="cell" onclick="makeMove(this)"></div>
  <div class="cell" onclick="makeMove(this)"></div>
  <div class="cell" onclick="makeMove(this)"></div>
  <div class="cell" onclick="makeMove(this)"></div>
 </div>
 <script>
  let currentPlayer = 'X';
  const cells = document.querySelectorAll('.cell');

  function makeMove(cell) {
   if (!cell.textContent) {
    cell.textContent = currentPlayer;
    if (checkWin()) {
```

```
            alert(`Гравець ${currentPlayer} виграв!`);
            resetBoard();
          } else if ([...cells].every(cell => cell.textContent !== '')) {
            alert("Нічия!");
            resetBoard();
          } else {
            currentPlayer = currentPlayer === 'X' ? 'O' : 'X';
          }
        }
      }

      function checkWin() {
        const lines = [
          [0, 1, 2], [3, 4, 5], [6, 7, 8], // рядки
          [0, 3, 6], [1, 4, 7], [2, 5, 8], // стовпці
          [0, 4, 8], [2, 4, 6]             // діагоналі
        ];

        for (const line of lines) {
          const [a, b, c] = line;
          if (cells[a].textContent && cells[a].textContent === cells[b].textContent && cells[a].textContent === cells[c].textContent) {
            return true;
          }
        }
        return false;
      }

      function resetBoard() {
        cells.forEach(cell => cell.textContent = '');
        currentPlayer = 'X';
      }
    </script>
  </body>
</html>
```

Завдання

Напишіть скрипт простої гри використовуючи мову JavaScript, стилізуйте гру та додайте до сторінки вашого веб-сайту.

ПРАКТИЧНА РОБОТА №6 (2 години)

«Дата і час. Застосування у веб-технологіях»

Мета: Познайомитися із особливостями відображення та реалізації дати та часу (календар та годинник) на веб-сайті мовою JavaScript.

Теоретична частина

Відображення на сайті дати та часу застосовується з різних причин. Наприклад, щоб користувачу було зручно орієнтуватися в часі коли він робить замовлення. Звісно не всі веб-сайти використовують клас Date при розробці.

Існують різні формати відображення дати і відвідувачів веб-сайту з різних місць вони можуть заплутати. Вибір конкретного формату залежить від замовника або від потреб майбутніх користувачів. Формат ***MM/DD/YY*** (місяць/день/рік) є унікальним для Сполучених Штатів Америки. Майже вся Європа використовує формат ***DD/MM/YY*** (день/місяць/рік). Японія використовує формат ***YY/MM/DD*** (рік/місяць/день). В Україні частіше за все ***DD/MM/YYYY*** . Розділювачами можуть бути: слеш, тире або періоди. В деяких місцях нулі, що ставляться попереду друкують, а в деяких їх упускають.

Клас `Date` в JavaScript використовується для роботи з даними та часом. Він дозволяє створювати об'єкти, які містять інформацію про дату та час, а також виконувати різноманітні операції з ними. Прилад основних методів та властивостей класу `Date`

Створення об'єкту (без аргументу видає поточну дату):
```
const currentDate = new Date();
```

Збереження різних компонентів дати
```
const year = currentDate.getFullYear(); // Рік (наприклад, 2023)
const month = currentDate.getMonth(); // Місяць (0 - січень, 11 - грудень)
const day = currentDate.getDate(); // День місяця (1 - 31)
const hours = currentDate.getHours(); // Години (0 - 23)
const minutes = currentDate.getMinutes(); // Хвиліни (0 - 59)
const seconds = currentDate.getSeconds(); // Секунди (0 - 59)
```

Форматування дати в рядок:
```
const formattedDate = currentDate.toISOString(); // "2023-08-22T14:30:00.000Z" ```
```

При роботі з ***об'єктом Date*** важливо пам'ятати, що обчислення виконуються за допомогою єдиного всесвітнього часу (UTC), незважаючи на те, що ваш комп'ютер може відображати час відповідно до вашої часової зони.Методи Метод Опис

getDate() Повертає день місяця (від 1 до 31) для зазначеної дати за місцевим часом.

getDay() Повертає день тижня (від 0 до 6; 0 = неділя, 1 = понеділок і т.д.) для зазначеної дати за місцевим часом.

getFullYear() Повертає рік (чотири цифри).

getHours() Повертає годину (від 0 до 23).

getMilliseconds() Повертає мілісекунди (від 0 до 999).

getMinutes() Повертає хвилини (від 0 до 59).

getMonth() Повертає місяць (від 0 до 11; 0 = січень, 1 = лютий тощо).

getSeconds() Повертає секунди (від 0 до 59).

getTime() Повертає кількість мілісекунд, що минули з півночі 01.01.1970.

getTimezoneOffset() Повертає різницю у часі між часом UTC та місцевим часом у хвилинах.

getUTCDate() Повертає день місяця за всесвітнім часом (від 1 до 31).

getUTCDay() Повертає день тижня за всесвітнім часом (від 0 до 6).

getUTCFullYear() Повертає рік за всесвітнім часом (чотири цифри).

getUTCHours() Повертає годину за всесвітнім часом (від 0 до 23).

getUTCMilliseconds() Повертає мілісекунди за всесвітнім часом (від 0 до 999).

getUTCMinutes() Повертає хвилини за всесвітнім часом (від 0 до 59).

getUTCMonth() Повертає місяць за всесвітнім часом (від 0 до 11).

getUTCSeconds() Повертає секунди за всесвітнім часом (від 0 до 59).

parse() Аналізує рядок дати (наприклад, "21 травня 1992 року") і повертає рядок зі значенням дати, що містить число в мілісекундах з 1 січня 1970 року 00:00:00.

setDate() Встановлює день місяця для зазначеної дати за місцевим часом (від 1 до 31).

setFullYear() Встановлює рік (чотири цифри).

setHours() Встановлює годинник для зазначеної дати за місцевим часом (від 0 до 23).

setMilliseconds() Встановлює мілісекунди для вказаної дати за місцевим часом.

setMinutes() Встановлює хвилини (від 0 до 59).

setMonth() Встановлює місяць (від 0 до 11).

setSeconds() Встановлює секунди (від 0 до 59).

setTime() Встановлює дату у мілісекундах після (або до) 01.01.1970.

setUTCDate() Задає день місяця.

setUTCFullYear() Встановлює рік за всесвітнім часом (чотири цифри).

setUTCHours() Встановлює годину для зазначеної дати за всесвітнім часом.

setUTCMilliseconds() Задає мілісекунди для вказаної дати за всесвітнім часом.

setUTCMinutes() Встановлює хвилини для зазначеної дати за всесвітнім часом.

setUTCMonth() Задає місяць для зазначеної дати за всесвітнім часом.

setUTCSeconds() Встановлює секунди для вказаної дати за всесвітнім часом.

toDateString()

toISOString() Перетворює дату на рядок, використовуючи стандарт ISO 8601.

toJSON() Повертає дату у вигляді рядка, відформатовану як дата JSON.

toLocaleDateString()

toLocaleTimeString() Повертає частину дати у вигляді рядка з поданням дати на основі параметрів системи.

toLocaleString() Повертає дату у вигляді рядка з поданням дати на основі параметрів системи.

toString() Повертає рядок, який представляє вказаний об'єкт Date.

toTimeString() Повертає частину дати у вигляді рядка.

toUTCString() Перетворює дату на рядок, використовуючи часовий пояс UTC.

UTC() Приймає ті ж параметри, що й довга форма конструктора (тобто 2-7) і повертає кількість мілісекунд з 1 січня 1970 р., 00:00:00 UTC.

valueOf() Повертає примітивне значення об'єкта Date.

Завдання

Написати скрипт, який буде відображати реальний час та календар на сторінці вашого веб-сайту.

КОНТРОЛЬНІ ЗАПИТАННЯ

1) Основи HTML (синтаксис, структура документа).
2) Основні теги HTML, робота з текстом, списки.
3) Створення посилань.
4) Зображення та їх властивості в HTML.
5) Створення таблиць, відступи, вирівнювання.
6) Використання форм в HTML-документі.
7) Поняття каскадних таблиць стилів CSS, їх призначення.
8) Типи стилів. Переваги стилів.
9) Способи додавання стилів на сторінку.
10) Базовий синтаксис CSS.
11) Класи у CSS.
12) Ідентифікатори у CSS.
13) Селектори і їх види.
14) Можливості JavaScript, де застосовується.
15) Робота JavaScript з DOM сторінки.
16) JavaScript команди і коментарі.
17) Змінні в JavaScript і операції над ними.
18) Арифметичні оператори в JavaScript.
19) Логічні оператори та умовні конструкції в JavaScript.
20) Вікна оповіщення і підтвердження в JavaScript.
21) JavaScript функції.
22) Локальні і глобальні змінні.
23) Цикли JavaScript.
24) Події та їх обробка.
25) Перевірка форм в JavaScript.
26) Спеціальні символи в JavaScript.
27) Методи об'єктів в JavaScript.
28) Масиви в JavaScript.

РЕКОМЕНДОВАНІ ДЖЕРЕЛА

1. HTML Living Standard [Електронний ресурс] – Режим доступу: https://html.spec.whatwg.org/multipage.

2. Resources for developers, by developers. MDN Web Docs. Basic sections of a document [Електронний ресурс] – Режим доступу: https://developer.mozilla.org/ru/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure.

3 Ресурс https://html5css.ru/html/html_images.php

4. Web-програмування. Частина 1 (frontend) : навч. посіб. / В. В. Босько, Л. В. Константинова, К. М. Марченко, О. С. Улічев ; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. - Кропивницький : ЦНТУ, 2022. - 208 с.

5 Client-side form validation [Електронний ресурс] – Режим доступу: https://developer.mozilla.org/en-US/docs/Learn/Forms/Form_validation.

6. Robin Nixon. Learning PHP, MySQL & JavaScript. 6th Ed. 2021, 825p.

7. Dave Crane, Eric Pascarello, Darren James. Ajax in Action. Manning; 1st edition (November 3, 2005) 680 pages.

8. Український веб-довідник [Електронний ресурс] – Режим доступу: https://css.in.ua/html/events.

9. Бородкіна І.Л., Бородкін Р. О. Web-технології та Web-дизайн : застосування мови HTML для створення електронних ресурсів. Видавництво: Ліра До, 2020, 212с.

10. PHP Documentation Group [Електронний ресурс] – Режим доступу: https://www.php.net/manual/en/intro-whatis.php.

11. Перевірка (валідація) полів в формі перед відправкою jQuery validator [Електронний ресурс] – Режим доступу: https://созданиесайта.net/news-new/proverka-validatsiya-poley-v-forme-pered-otpravkoyjquery.html .

12. jQuery Validation Plugin [Електронний ресурс] – Режим доступу: https://jqueryvalidation.org/documentation/ .

13. Web-програмування. Лабораторний практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 125 «Кібербезпека» та 113 «Прикладна математика» / А. Ю. Шелестов, Н. М. Куссуль; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 1047 Кбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 61 с.

ДОДАТОК

Приклад оформлення звіту з практичної роботи

Міністерство освіти і науки України
Національний технічний університет
«Харківський політехнічний інститут»
Кафедра стратегічного управління

ЗВІТ
ПРО ВИКОНАННЯ ПРАКТИЧНОЇ РОБОТИ №__
з дисципліни «Основи Веб-технологій»

Тема: «_____»

Виконав(ла): студент гр. КН -___
_____ПІБ студента(тки)_____
Перевірив(ла): викладач
_____ПІБ викладача_____

Харків 202_

Навчальне видання

Методичні вказівки
для практичних робіт
«Основи Веб-технологій»
з дисципліни «Основи Веб-технологій»

для студентів
122 спеціальності – комп’ютерні науки

Укладачі:
ЛИСЕНКО Антон Олександрович
ШУБА Ірина Володимирівна

Відповідальний за випуск (завідувач кафедри) Марина ГРИНЧЕНКО
Роботу рекомендував до друку (експерт РВР) Ігор ГАМАЮН
Комп’ютерна верстка _____
В авторській редакції

План 2023 р., поз. 268

Підп. до друку (дата підпису проректора)_____.
Гарнітура Times New Roman.

Видавничий центр НТУ «ХПІ».
Свідоцтво про державну реєстрацію ДК № 5478 від 21.08.2017 р.
61002, Харків, вул. Кирпичова, 2