Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Автоматизації
електротехнічних та
мехатронних комплексів

Основи веб-розробки

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни	
Рівень вищої освіти	*Перший (бакалаврський)*
Галузь знань	*G Інженерія, виробництво та будівництво*
Спеціальність	*G3 Електрична інженерія*
Освітня програма	*Інжиніринг інтелектуальних електротехнічних та мехатронних комплексів*
Статус дисципліни	*Вибіркова*
Форма навчання	*Очна (денна)*
Рік підготовки, семестр	*2 рік навчання, весняний семестр*
Обсяг дисципліни	*4 кредити, 120 годин (30 год. лекцій, 30 год. практичних, 60 год. СРС)*
Семестровий контроль/ контрольні заходи	*Залік, МКР*
Розклад занять	https://schedule.kpi.ua/
Мова викладання	*Українська*
Інформація про керівника курсу / викладачів	*Лекції, практичні, лабораторні, практичні: PhD, ст.викл. Мугенов Даніїл Джалільович, тел. 068-240-24-43, email: muhenov.daniil@lll.kpi.ua;*
Розміщення курсу	*https://classroom.google.com/ (посилання на курс надається на першому занятті)*

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчання та результати навчання

Дисципліна «Основи веб-розробки» забезпечує формування у студентів практичних навичок створення сучасних веб-сторінок і веб-застосунків: від верстки статичних сторінок засобами HTML/CSS до розробки інтерактивного клієнтського коду на JavaScript, роботи з відкритими REST API та основ серверної розробки. Особлива увага приділяється застосуванню веб-технологій для побудови інтерфейсів моніторингу та візуалізації даних електротехнічних і мехатронних комплексів (дашборди, вебпанелі відображення показників датчиків, інтеграція з відкритими сервісами даних).

Мета вивчення дисципліни – надати студентам комплексні знання та практичні навички розробки веб-сторінок і найпростіших веб-застосунків повного циклу: від макетування та верстки до інтерактивності на стороні клієнта та базової взаємодії із сервером.

Предметом вивчення дисципліни є мова розмітки HTML, каскадні таблиці стилів CSS, мова програмування JavaScript, принципи побудови REST API, а також базові інструменти командної розробки (система контролю версій Git).

Програмні результати навчання: розробляти статичні та інтерактивні веб-сторінки з використанням семантичної розмітки та адаптивної верстки; застосовувати мову JavaScript для обробки подій та динамічної зміни вмісту сторінки; здійснювати інтеграцію веб-застосунку із зовнішніми джерелами даних через REST API; використовувати систему контролю версій Git у процесі командної розробки програмного забезпечення.

Компетентності: здатність розробляти та впроваджувати програмні засоби візуалізації й моніторингу роботи електротехнічних, електромеханічних та мехатронних систем; здатність застосовувати сучасні інформаційні технології та інструменти командної розробки програмного забезпечення для вирішення прикладних інженерних задач.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Навчальна дисципліна «Основи веб-розробки» викладається на основі знань та умінь, одержаних студентами під час вивчення кредитних модулів дисциплін «Інформатика» та «Об'єктно-орієнтоване програмування». Отримані знання дозволять студентам самостійно розробляти веб-інтерфейси для систем моніторингу та диспетчеризації електротехнічних комплексів, брати участь у розробці програмного забезпечення для інтелектуальних систем прийняття рішень, а також використовуватимуться під час виконання випускних кваліфікаційних робіт, що передбачають розробку програмних засобів візуалізації даних.

3. Зміст навчальної дисципліни

Навчальна дисципліна «Основи веб-розробки» складається з 4 розділів:

– Розділ 1. Основи розмітки та стилізації веб-сторінок (HTML, CSS)

– Розділ 2. Адаптивна верстка та сучасні технології компонування (Flexbox, Grid, Media Queries)

– Розділ 3. Основи програмування на JavaScript та взаємодія з DOM

– Розділ 4. Асинхронність, REST API, контроль версій та основи серверної розробки

4. Навчальні матеріали та ресурси

Базова література:

1. Дакетт, Дж. (2019). HTML і CSS. Розробка і дизайн вебсайтів. Київ: КМ-Букс.

2. Дакетт, Дж. (2020). JavaScript і jQuery. Інтерактивна веброзробка. Київ: КМ-Букс.

3. MDN Web Docs – офіційна документація HTML. https://developer.mozilla.org/uk/docs/Web/HTML
4. MDN Web Docs – офіційна документація CSS. https://developer.mozilla.org/uk/docs/Web/CSS
5. MDN Web Docs – CSS Flexbox та Grid. https://developer.mozilla.org/uk/docs/Web/CSS/CSS_Flexible_Box_Layout
6. Marcotte, E. (2011). Responsive Web Design. New York: A Book Apart.
7. MDN Web Docs – офіційна документація JavaScript. https://developer.mozilla.org/uk/docs/Web/JavaScript
8. Flanagan, D. (2020). JavaScript: The Definitive Guide, 7th Edition. O'Reilly Media.
9. Chacon, S., Straub, B. (2014). Pro Git, 2nd Edition. Apress. https://git-scm.com/book/uk/v2
10. Node.js Documentation. https://nodejs.org/docs/

Допоміжна література:

1. Fielding, R. (2000). Architectural Styles and the Design of Network-based Software Architectures (REST). Дисертація.
2. Інтернет-ресурси: freeCodeCamp, developer.mozilla.org, git-scm.com, PVGIS API documentation.

Обов'язковим для прочитання є окремі розділи базової літератури [1]–[10]. Розділи базової літератури, що є обов'язковими для прочитання, та зв'язок цих ресурсів з конкретними темами дисципліни наводиться в методиці опанування навчальної дисципліни. Усі інші джерела є факультативними, з ними рекомендується ознайомитись.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

Застосовуються стратегії активного навчання: лекції-презентації з демонстрацією живого коду, практичні заняття з виконанням завдань у середовищі розробки, виконання індивідуального комплексного проєкту (веб-сторінка з інтерактивними елементами та підключенням зовнішнього API).

№	Назва теми лекції та перелік основних питань	Література
1	Вступ до веб-розробки. Архітектура клієнт-сервер. Принцип роботи HTTP/HTTPS. Огляд сучасного стеку технологій (front-end/back-end). Роль веб-технологій у задачах моніторингу та диспетчеризації електротехнічних комплексів.	[1, 2]
2	Основи HTML. Структура HTML-документа. Семантична розмітка (header, nav, main, section, footer). Основні теги: текст, посилання, списки, зображення.	[1, 3]
3	Таблиці та форми в HTML. Елементи форм (input, select, textarea), атрибути валідації. Вбудовування мультимедіа (audio, video, iframe).	[1, 3]
4	Основи CSS. Синтаксис, селектори (типи, класи, id, атрибутивні, псевдокласи). Каскадність та специфічність. Box model: margin, border, padding, content.	[1, 4]
5	Позиціонування та відображення елементів. Властивість display (block, inline, inline-block, none). Позиціонування (static, relative, absolute, fixed). Z-index.	[1, 4]
6	Технологія Flexbox: властивості контейнера та елементів, побудова однорядних/однорядкових макетів.	[4, 5]

№	Назва теми лекції та перелік основних питань	Література
7	Технологія CSS Grid: побудова дворівневих сіткових макетів, іменовані області (grid-template-areas).	[4, 5]
8	Адаптивний веб-дизайн. Media Queries, підхід Mobile-First. Одиниці вимірювання (px, %, em, rem, vw/vh). Адаптивні зображення.	[4, 6]
9	Основи JavaScript. Змінні (let, const), типи даних, оператори, керуючі конструкції (умови, цикли).	[2, 7]
10	Функції в JavaScript: оголошення, параметри, стрілкові функції. Робота з масивами та об'єктами, методи масивів (map, filter, reduce).	[2, 7]
11	DOM (Document Object Model): вибір та зміна елементів сторінки, створення елементів, обробка подій користувача (click, submit, input).	[2, 8]
12	Асинхронність у JavaScript: Callback, Promise, async/await. Fetch API: надсилання запитів, обробка відповіді у форматі JSON.	[2, 8]
13	Робота з REST API. Принципи побудови REST-сервісів, HTTP-методи (GET, POST, PUT, DELETE). Практика інтеграції зовнішніх відкритих API (напр., дані сонячної інсоляції PVGIS, погодні сервіси) у веб-сторінку.	[8, 9]
14	Контроль версій Git та платформа GitHub: репозиторії, коміти, гілки, злиття, робота в команді. Основи хостингу статичних сайтів (GitHub Pages, Netlify).	[9]
15	Вступ до серверної розробки: середовище Node.js, основи побудови найпростішого серверного застосунку. Основи безпеки веб-застосунків (XSS, HTTPS, валідація вхідних даних). Узагальнення матеріалу курсу.	[9, 10]

Практичні заняття

№	Завдання, які виносяться на практичне заняття
1	Налаштування середовища розробки (VS Code, розширення, браузерні DevTools). Створення першої HTML-сторінки із семантичною розміткою.
2	Верстка текстової сторінки: заголовки, абзаци, списки, зображення, посилання. Перевірка валідності HTML-коду.
3	Розробка HTML-форми зворотного зв'язку з елементами валідації (required, pattern, type=email).
4	Стилізація сторінки засобами CSS: кольори, шрифти, фон, робота із селекторами та каскадом.
5	Верстка макета сторінки за допомогою Flexbox (навігаційне меню, картки контенту).
6	Верстка макета сторінки за допомогою CSS Grid (сітка галереї, дашборд-макет).
7	Адаптація верстаного макета під мобільні пристрої за допомогою Media Queries. Перевірка у режимі responsive design у DevTools.
8	Написання базових скриптів JavaScript: робота зі змінними, умовами, циклами на прикладі обчислювальних задач.
9	Обробка масивів об'єктів (напр., масив показників датчиків) методами map/filter/reduce.

№	Завдання, які виносяться на практичне заняття
10	Динамічна зміна контенту сторінки через DOM: додавання/видалення елементів за подіями користувача.
11	Валідація форми на JavaScript та обробка події submit без перезавантаження сторінки.
12	Отримання даних із відкритого REST API (Fetch, async/await) та їх відображення на сторінці у вигляді таблиці/графіка.
13	Робота з Git та GitHub: створення репозиторію, гілки, коміти, pull request, вирішення конфліктів злиття.
14	Розробка мінімального серверного застосунку на Node.js, що віддає статичні файли та обробляє один API-маршрут.
15	Розгортання (deployment) готового проєкту на GitHub Pages/Netlify. Захист індивідуального навчального проєкту.

6. Самостійна робота студента

Самостійна робота студента (60 год.) передбачає:

– підготовку до аудиторних (лекційних та практичних) занять – 40 год.;

– підготовку до модульної контрольної роботи – 8 год.;

– виконання індивідуального навчального проєкту (веб-сторінка з інтеграцією API) – 8 год.;

– підготовку до заліку – 4 год.

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

На момент проведення кожного заняття, як лекційного, так і практичного, у студента на пристрої, з якого він працює, має бути встановлено додаток Zoom (у випадку дистанційного навчання), а також відкрито курс «Основи веб-розробки» на платформі «Сікорський» (код доступу до курсу надається на першому занятті згідно з розкладом).

Силабус; лекційний матеріал; завдання до кожного практичного заняття; варіанти модульної контрольної роботи; методичні рекомендації до виконання індивідуального проєкту; варіанти залікової контрольної роботи розміщено на платформі «Сікорський» та у системі «Електронний Кампус КПІ».

Під час проходження курсу «Основи веб-розробки» студенти зобов'язані дотримуватись загальних моральних принципів та правил етичної поведінки, зазначених у Кодексі честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

Дедлайни виконання кожного завдання зазначено у курсі «Основи веб-розробки» на платформі «Сікорський».

Усі без винятку студенти зобов'язані дотримуватись вимог Положення про систему запобігання академічному плагіату в Національному технічному університеті України «Київський політехнічний інститут імені Ігоря Сікорського». Індивідуальні проєкти перевіряються на оригінальність коду; запозичення фрагментів коду без посилання на джерело прирівнюється до академічного плагіату.

За участь у Всеукраїнській олімпіаді (конкурсі студентських проєктів з веброзробки) студенту нараховується 5 (І тур) або 10 (ІІ тур) балів. За публікацію індивідуального проєкту у відкритому репозиторії з детальною документацією (README) нараховується додатково 5 балів. Загальна сума заохочувальних балів не може перевищувати 10 балів.

8. Види контролю та рейтингова система оцінювання результатів навчання (РСО)

Поточний контроль: завдання в рамках практичного заняття (15 практичних занять × 4 бали = 60 балів); модульна контрольна робота (МКР, 40 балів).

МКР проводиться на останньому практичному занятті у присутності викладача у вигляді тесту та короткого практичного завдання (написання/виправлення фрагмента HTML/CSS/JS-коду). Тест містить 20 запитань з кількома варіантами відповіді, серед яких лише один правильний; кожна правильна відповідь оцінюється в 1 бал (20 балів); практичне завдання оцінюється в 20 балів.

Завдання в рамках практичного заняття оцінюються в 4 бали за такими критеріями:

– «відмінно» – повністю виконане завдання (не менше 90% необхідного функціоналу), код відповідає стандартам якості, надано пояснення рішення – 4 бали;

– «добре» – завдання виконане достатньо повно (не менше 75% необхідного функціоналу), містить незначні недоліки – 3 бали;

– «задовільно» – завдання виконане неповно (не менше 60% необхідного функціоналу), містить помилки – 2,5 бали;

– «незадовільно» – завдання не виконане або виконане менше ніж на 60% – 0 балів.

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу. Умовою позитивного першого та другого календарного контролів є отримання не менше 50% максимально можливого на момент відповідного календарного контролю рейтингу.

RC(min) = 7 × 4 = 28 балів (на момент першого календарного контролю); RC(min) = 15 × 4 = 60 балів (на момент другого календарного контролю, до виконання МКР).

Семестровий контроль: залік. Умови допуску до семестрового контролю: виконані та зараховані практичні заняття, виконана МКР.

Студенти, які виконали всі умови допуску до заліку та мають рейтингову оцінку 60 і більше балів, отримують відповідну до набраного рейтингу оцінку без додаткових випробувань. Якщо сума балів менша за 60, але виконані практичні та МКР, студент виконує залікову роботу. Студент, який у семестрі отримав більше 60 балів, але бажає підвищити свій результат, може також взяти участь у заліковій роботі. На заліку студенти виконують письмову контрольну роботу, яка оцінюється в 100 балів (тестова частина – 40 балів, практичне завдання з написання коду – 60 балів).

Вид роботи	Кількість	Балів
Практичні заняття	15 × 4 бали	60
Модульна контрольна робота (МКР)	1 × 40 балів	40
Разом	—	**100**

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

Перелік питань, які виносяться на семестровий контроль, наведено у додатку до силабусу.

Здобувач вищої освіти має можливість пройти онлайн-курс(и) за однією або декількома темами, передбаченими робочою програмою навчальної дисципліни (напр., на платформах Coursera, freeCodeCamp, Prometheus). Онлайн-курс здобувач може обрати самостійно або за рекомендацією викладача. 1 год прослуханого курсу оцінюється у 0,83 бали. Максимальна кількість годин, яка може бути зарахована за результатами неформальної освіти, становить 12 год, відповідно максимальна кількість балів за такі результати становить 10 балів.

Робочу програму навчальної дисципліни (силабус):

Складено: старший викладач кафедри автоматизації електротехнічних та мехатронних комплексів, PhD, Мугенов Даніїл Джалільович

Ухвалено кафедрою автоматизації електротехнічних та мехатронних комплексів. Протокол №27 від 23.06.26 р.

Ухвалено Навчально - методичною комісією НН ІЕЕ. Протокол №43 від 29.06.26 р.

Додаток до силабусу освітнього компонента

курсу «Основи веб-розробки»

Перелік завдань, що виносяться на семестровий контроль

1. Опишіть базову структуру HTML-документа (doctype, html, head, body). Яке призначення тегу <meta charset="UTF-8">?

2. У чому різниця між семантичними тегами (header, main, article) та звичайним ? Навіщо використовувати семантичну розмітку?

3. Поясніть різницю між класом (class) та ідентифікатором (id) у CSS. Який селектор має вищу специфічність?

4. Що таке box model у CSS? Поясніть різницю між box-sizing: content-box та box-sizing: border-box.

5. У чому різниця між позиціонуванням relative, absolute та fixed?

6. Поясніть принцип роботи Flexbox: призначення властивостей justify-content та align-items.

7. Чим CSS Grid відрізняється від Flexbox? У яких випадках доцільніше використовувати кожну технологію?

8. Що таке Mobile-First підхід до адаптивної верстки? Наведіть приклад Media Query.

9. Чим відрізняються оператори порівняння == та === у JavaScript?

10. У чому різниця між var, let та const у JavaScript?

11. Поясніть різницю між функцією-виразом та стрілковою функцією (arrow function) у JavaScript.

12. Що таке DOM? Наведіть приклад методу для вибору елемента сторінки та зміни його вмісту.

13. Опишіть механізм спливання подій (event bubbling) у JavaScript.

14. Що таке Promise у JavaScript? Які три стани може мати Promise?

15. Поясніть різницю між синтаксисом .then()/.catch() та async/await під час роботи з асинхронним кодом.

16. Що таке REST API? Назвіть основні HTTP-методи та їх призначення.

17. Яке призначення коду відповіді HTTP 404? Наведіть ще 2 приклади поширених кодів відповіді та їх значення.

18. Що таке репозиторій у Git? Чим відрізняються команди git commit та git push?

19. Що таке злиття гілок (merge) у Git? Що таке конфлікт злиття (merge conflict) і як його вирішити?

20. Яке призначення файлу package.json у проєкті на Node.js?

Приклади практичних завдань

Завдання 1. Виправлення помилки в CSS-селекторі

Умова: розробник написав правило .card:hover { color: blue } , але воно застосовується до всіх елементів .card, а не лише до дочірнього заголовка .card h3.

Завдання: запишіть правильний селектор, щоб колір змінювався лише для заголовка h3 всередині картки під час наведення курсора на картку.

Розв'язок: .card:hover h3 { color: blue; }

Відповідь: властивість застосовується до нащадка h3 при наведенні на батьківський елемент .card завдяки комбінатору «пробіл» (нащадок).

Завдання 2. Верстка Flexbox

Умова: є контейнер із трьома картками однакової ширини, які мають розташовуватись в один рядок з рівномірними проміжками та бути вирівняними по вертикалі по центру.

Завдання: напишіть CSS-правила для контейнера, використовуючи Flexbox.

Розв'язок: .container { display: flex; justify-content: space-between; align-items: center; gap: 16px; }

Завдання 3. Прогнозування результату виконання коду JavaScript

Умова:

```
let arr = [10, 20, 30];
let result = arr.map(x => x / 10).filter(x => x > 1);
console.log(result);
```

Завдання: визначте, що буде виведено в консоль.

Розв'язок: метод map ділить кожен елемент на 10, отримуючи [1, 2, 3]; метод filter залишає лише елементи, більші за 1, отримуючи [2, 3].

Відповідь: [2, 3]

Завдання 4. Робота з Fetch API

Умова: потрібно отримати дані з відкритого API за адресою https://api.example.com/sensors та вивести значення поля temperature у консоль, коректно обробивши можливу помилку мережі.

Завдання: напишіть відповідний код з використанням async/await та конструкції try/catch.

Розв'язок:

```
async function getTemperature() {
  try {
    const response = await fetch('https://api.example.com/sensors');
    const data = await response.json();
    console.log(data.temperature);
  } catch (error) {
    console.error('Помилка отримання даних:', error);
  }
}
```