



# FRONT-END РОЗРОБКА

## Робоча програма навчальної дисципліни (Силабус)

### Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>11 Математика та статистика</i>
Спеціальність	<i>113 Прикладна математика</i>
Освітня програма	<i>Наука про дані та математичне моделювання</i>
Статус дисципліни	<i>Вибіркова</i>
Форма навчання	<i>очна(денна)</i>
Рік підготовки, семестр	<i>3 курс, весняний семестр</i>
Обсяг дисципліни	<i>4 кредити</i>
Семестровий контроль/ контрольні заходи	<i>Залік/Контрольні (3) та лабораторні (6) роботи</i>
Розклад занять	<i>Лекції — 1 раз на тиждень (18 лекцій), лабораторні — 1 раз на тиждень (18 занять)</i>
Мова викладання	<i>Українська/Англійська</i>
Інформація про керівників курсу / викладачів	Лектор: <i>Борисенко Павло Борисович, <a href="mailto:pavlo.borysenko@gmail.com">pavlo.borysenko@gmail.com</a></i> Практичні: <i>Борисенко Павло Борисович, <a href="mailto:pavlo.borysenko@gmail.com">pavlo.borysenko@gmail.com</a>; Громова Вікторія Вікторівна, <a href="mailto:vikvikgrom@gmail.com">vikvikgrom@gmail.com</a></i>
Розміщення курсу	Канали #web-course та #web-labs у кафедральному Slack Матеріали: <a href="https://drive.google.com/drive/folders/1Ohhz_8yNXob5UMRT2OYKzPDN2dKNKDb0">https://drive.google.com/drive/folders/1Ohhz_8yNXob5UMRT2OYKzPDN2dKNKDb0</a>

### Програма навчальної дисципліни

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

*Front-end розробка — важливий розділ у сучасній розробці програмного забезпечення. Розуміння принципів front-end розробки дозволяє проектувати стабільні та підтримувані веб-застосунки, які становлять важливу частину сучасної користувацької екосистеми. З точки зору науки про дані та математичного моделювання, веб-програмування загалом і його front-end частина важливі в першу чергу як джерело даних та як засіб візуалізації та представлення цих даних користувачу.*

*В рамках цього курсу ми розглянемо основні концепції веб-розробки, що дозволять вам легше орієнтуватися у сучасних веб-технологіях та краще зрозуміти зв'язок між ними та рештою програмуванням, а також зрозуміти особливості роботи в клієнт-серверних системах. А на лабораторних роботах ви зможете з нуля опанувати мову програмування JavaScript — класичну мову програмування на клієнті, — від простих структур даних до фреймворку React від Facebook.*

*Курс «Front-end розробка» є вибіркоvim курсом підготовки спеціалістів у сфері науки про дані та математичного моделювання, що дозволяє краще осягнути прикладні аспекти програмування у клієнт-серверних системах.*

Метою курсу є вивчення:

- основних понять і концепцій веб-програмування,
- базових конструкцій та синтаксису мови JavaScript,
- принципів та підходів до побудови клієнт-серверних систем з точки зору клієнта;
- принципів роботи з REST API.

Предметом вивчення є програмний код мовою JavaScript та його синтаксичні примітиви; структури даних та примітиви фреймворку React; принципи і підходи до організації клієнт-серверної взаємодії.

Після засвоєння дисципліни студенти матимуть наступні:

- компетентності:
  - здатність до абстрактного мислення, аналізу та синтезу;
  - знання та розуміння предметної області та розуміння професійної діяльності;
  - здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію;
  - здатність використовувати сучасні технології програмування та тестування програмного забезпечення;
  - здатність зрозуміти постановку завдання, сформульовану мовою певної предметної галузі, здійснювати пошук та збір необхідних вихідних даних;
- знання:
  - синтаксису та принципів роботи HTML та CSS;
  - синтаксису мови JavaScript та її розширення JSX;
  - про принципи клієнт-серверної взаємодії;
  - принципів динамічної роботи з веб-застосунками;
  - принципів роботи фреймворку React
  - принципів побудови та ефективної обробки даних із різних джерел;
  - патернів веб-розробки;
  - принципів роботи з REST API та транспіляторами;
- уміння:
  - конструювати і деконструювати веб-сторінки;
  - організовувати роботу із зовнішними джерелами даних, що використовують технологію REST;
  - працювати із динамічними веб-сторінками;
  - використовувати патерни веб-розробки та підходи разом з іншими парадигмами;
- навички:
  - створення та стилювання веб-сторінок;
  - роботи з REST API;
  - роботи з React;
  - роботи зі слабо типізованою мовою програмування;
- досвід:
  - програмування веб-застосунків мовою JavaScript;
  - використання React;
  - написання REST API;
  - роботи із зовнішніми джерелами даних.

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

Дисципліна вивчається у весняному семестрі 3 курсу та базується на результатах навчання з дисциплін:

- Алгоритми і структури даних
- Програмування

- Програмування мовою Python
- Об'єктно-орієнтоване програмування
- Функціональне програмування.

Успішне проходження курсу допоможе студентам у вивченні наступних дисциплін з навчального плану підготовки бакалаврів:

- Аналіз даних
- Інформаційні системи.

### **3. Зміст навчальної дисципліни**

#### *Розділ 1. Базові принципи веб-розробки*

*Тема 1.1. Вступ до веб-розробки. Архітектура клієнт-серверної взаємодії.*

*Тема 1.2. Структура HTTP запитів. URI та адресація. REST API.*

*Тема 1.3. Семантична модель документу. HTML і об'єктна модель документа (DOM).*

*Тема 1.4. Сильове оформлення документа і CSS.*

#### *Розділ 2. Мова програмування JavaScript*

*Тема 2.1. Вступ до JavaScript. Типи, функції та об'єкти. JSON.*

*Тема 2.2. Маніпуляції DOM. Події і таймери.*

*Тема 2.3. AJAX: XMLHttpRequest і fetch. Асинхронна взаємодія із сервером.*

*Тема 2.4. JavaScript-фреймворки: jQuery та React. Virtual DOM.*

#### *Розділ 3. React та спеціальні теми.*

*Тема 3.1. Базові конструкції та типи. Генерація сторінок і шаблонізація. Роутинг.*

*Тема 3.2. Автентифікація користувачів. Хеші та cookies. Валідація даних і обробка помилок.*

*Тема 3.3. Побудова і використання REST API на основі React.*

*Тема 3.4. Використання React компонентів. Серверний рендеринг.*

*Тема 3.5. Розгортання застосунку онлайн.*

### **4. Навчальні матеріали та ресурси**

#### **Підручники**

Ви можете обрати один з наступних підручників:

1. Alex Banks, Eve Porcello, *Learning React*, 2017.  
Англійська версія: <https://www.oreilly.com/library/view/learning-react/9781491954614/>  
Це найбільш доступний за викладом підручник.  
Розділи 2-7, 11-12 — обов'язкові. Розділи 8-10 — рекомендовані.
2. Marijn Haverbeke, *Eloquent JavaScript*, 2018.  
Англійська версія: <https://eloquentjavascript.net/>  
Російська версія: [https://karmazzin.gitbook.io/eloquentjavascript\\_ru/](https://karmazzin.gitbook.io/eloquentjavascript_ru/)  
Цей підручник більше практично орієнтований.  
Розділи 1-6, 8-11, 13-15, 18, 20 — обов'язкові.  
Решта — на ваш вибір, вони покривають великий масив цікавих прикладних задач.
3. *The Modern JavaScript Tutorial*.  
Англійська версія: <https://javascript.info/>  
Російська версія: <https://learn.javascript.ru/>  
Це непоганий довідковий підручник із завданнями.

## Додаткові матеріали

1. Документація MDN.  
Посилання: <https://developer.mozilla.org/en-US/>
2. Документація React.  
Посилання: <https://uk.reactjs.org/>  
Тут же є і хороший курс для початківців.
3. Документація HTML і CSS.  
HTML: <https://html.spec.whatwg.org/multipage/>  
CSS: <https://www.w3.org/Style/CSS/current-work>  
Це доволі суха документація. Радимо її читати лише для загального ознайомлення.
4. W3School.  
Посилання: <https://www.w3schools.com/>  
Це набір тьюторіалів та гайдів з багатьох базових речей.

## Рекомендовані онлайн-курси

1. Coursera. Full-Stack Web Development with React.  
<https://www.coursera.org/specializations/full-stack-react>
2. EdX. HTML5 and CSS Fundamentals.  
<https://www.edx.org/course/html5-and-css-fundamentals>
3. EdX. Programming for the Web with JavaScript.  
<https://www.edx.org/course/programming-for-the-web-with-javascript>

## Навчальний контент

### 5. Методика опанування навчальної дисципліни (освітнього компонента)

#### Лекції

##### Розділ 1. Базові принципи веб-розробки

Лекція 1. Вступ до веб-розробки. Архітектура клієнт-серверної взаємодії. Модель OSI. HTTP.

Лекція 2. Структура HTTP запитів. URI та адресація. REST API. Restful модель. Мікросервісні архітектури.

Лекція 3. Семантична модель документу. HTML розмітка та об'єктна модель документу (DOM). Принципи рендерингу веб-сторінок. Позиціонування і box model.

Лекція 4. Стильове оформлення документа. Каскадні таблиці стилів (CSS). Принцип відділення семантики і представлення. Невізуальні елементи.

##### Розділ 2. Мова програмування JavaScript

Лекція 5. Вступ до JavaScript. Типи, функції і об'єкти. JSON. Особливості роботи зі слабо-типізованими мовами. Вказівник this. Класи. Транспіляція в JavaScript.

Лекція 6. Маніпуляції DOM. Події та таймери. Обробка вводу від користувача. Sandbox і взаємодія із операційною системою.

Лекція 7. AJAX: XMLHttpRequest і fetch. Асинхронна взаємодія із сервером. Веб-сокети.

Лекція 8. JavaScript-фреймворки. jQuery. React. Node.js і серверний JavaScript.

Лекція 9. Встановлення React і Vabel. Компонентна модель. Типові патерни веб-розробки.

##### Розділ 3. React та спеціальні теми.

Лекція 10. Базові типи даних. Два типи побудови компонент. Генерація сторінок і шаблонізація. Роутинг.

Лекція 11. Автентифікація користувачів. Хеші і cookies. Валідація даних і обробка помилок. Принципи роботи з користувацькими даними.

*Лекція 12. Побудова та використання REST API на основі React. Принципи мікросервісних архітектур.*

*Лекція 13. Використання React компонент. Серверний рендеринг.*

*Лекція 14. Розгортання застосунку онлайн.*

## **Лабораторні роботи**

### **Розділ 1. Базові принципи веб-розробки**

*Лабораторна 0. Розробка структури застосунку.*

*Завдання: вибрати та узгодити тематику та вміст веб-застосунку. Розробити структуру навігації та дизайн застосунку (ескізно).*

*Лабораторна 1. Принципи семантичного дизайну та розмітки. Каскадні таблиці стилів. Таблична та сіткова верстка. Box model.*

*Завдання: реалізувати семантичну модель кількох елементів інтерфейсу веб-застосунку (сторінок чи окремих компонентів) за допомогою HTML-розмітки. Реалізувати стильове оформлення принаймні одного елементу інтерфейсу за допомогою каскадної таблиці стилів.*

### **Розділ 2. Мова програмування JavaScript**

*Лабораторна 2. Маніпуляції об'єктною моделлю документа. Події і обробка вводу.*

*Завдання: реалізувати принаймні один елемент інтерфейсу (сторінку, компонент), що містить елементи взаємодії із користувачем; скрипт, що реалізовує обробку даних від користувача, та модифікує сторінку без перезавантаження. Реалізувати обробку кількох подій.*

*Лабораторна 3. React та віртуальна DOM. AJAX та асинхронне програмування.*

*Завдання: реалізувати набір компонент, що реалізують відображення віджету з погодою на кілька днів (або іншого, за погодженням із викладачем), що використовує асинхронне отримання даних за допомогою AJAX, виконує блокування інтерфейсу, де потрібно, та обробку помилок комунікації із зовнішнім сервером.*

### **Розділ 3. React та спеціальні теми**

*Лабораторна 4. REST API та обмін даними між клієнтом та сервером.*

*Завдання: реалізувати компонент, що генерує дані у форматі JSON. Реалізувати API-ендпоінт, що повертає згенеровані дані. Реалізувати систему роутингу. Реалізувати компонент, що отримує дані з цього ендпоінту.*

*Лабораторна 5. Node.js і динамічні компоненти. Верифікація даних. Автентифікація.*

*Завдання: реалізувати компонент для автентифікації користувачів, що взаємодіє із зовнішньою базою даних за допомогою REST API. Реалізувати верифікацію уведених користувачем даних та їх динамічне відображення.*

## **6. Самостійна робота студента**

*До самостійного опрацювання виносяться:*

- підготовка до аудиторних та лабораторних занять — до 2 годин на тиждень;
- підготовка до контрольних робіт та заліку — до 10 годин за семестр;
- виконання лабораторних робіт — до 60 годин;
- вивчення наступних тем:
  - Базові теги HTML та CSS.
  - Синтаксис JavaScript та JSX.
  - Адаптивна верстка.



- Розробка з урахуванням потреб людей із обмеженими можливостями.
- Використання Geolocation API та Swagger.

## Політика та контроль

### 7. Політика навчальної дисципліни

#### Відвідування

Відвідування лекцій необов'язкове, але ми заохочуємо студентів не пропускати лекційні заняття через можливість ставити уточнюючі питання та брати участь у живому обговоренні. Записи лекцій (та оглядової частини лабораторних занять) поточного чи минулих років будуть доступні онлайн.

Виконання і захист лабораторних робіт обов'язкове. Активність на лабораторних заняттях становить 60% семестрового рейтингу, тому не варто нехтувати нею.

Пропущені контрольні роботи можна перездати за погодженням із викладачем.

#### Оцінювання

Оцінювання контрольних та практичних робіт відбувається з точністю до десятих, округлення за звичними правилами.

#### Дедлайни

Контрольні роботи мають бути здані у рамках часу, відведеного на їх проведення.

Лабораторні роботи мають окремі визначені терміни (дедлайни). Роботи, здані після цих термінів, будуть оцінюватися з модифікатором:

- здані після **софт дедлайну** (окремі для кожної роботи, але не раніше ніж за 2 тижні після отримання завдання) — 0,5, тобто отримують половину балів;
- здані після **хард дедлайну** (тиждень до заліку) — не отримують балів.

#### Додаткові бали

Активність на лекціях та лабораторних заняттях — відповіді на запитання викладача, знаходження помилок у лекційних чи лабораторних матеріалах; питання, що свідчать про вдумливу роботу з навчальним матеріалом, надання оригінальних рішень у лабораторних чи домашній практичній роботах тощо — заохочується додатковими балами на розсуд викладача.

Крім того, заохочується додатковими балами підтверджене сертифікатами проходження курсів (онлайн чи офлайн), що стосуються тем дисципліни.

Протягом курсу можна отримати не більше 10 додаткових балів.

#### Академічна доброчесність

Ми підтримуємо принципи академічної доброчесності і рівності всіх студентів. У випадку виявлення випадків списування (у контрольних, лабораторних, домашніх роботах) чи плагіату — бали за відповідні роботи будуть анульовані. Повторні порушення принципів академічної доброчесності можуть призвести до недопуску до складання заліку.

Викладачі можуть перевіряти роботи, виконані у рамках курсу, за допомогою систем виявлення плагіату Unichек та MOSS.

Якщо не зазначено іншого, усі контрольні заходи проводяться у форматі «відкритої книги». Це означає, що ви маєте право користуватися будь-якими ресурсами, окрім допомоги сторонніх осіб. Ми довіряємо нашим студентам і покладаємо надію на те, що вони не порушать цю довіру.

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

Поточний контроль:

- **модульні контрольні роботи (30%):**

10 балів x 3 роботи = **30** балів

Модульні контрольні роботи представляють собою 10 завдань за темою модуля, зазвичай 5 тестових і 5 практичних. На виконання кожної модульної контрольної роботи виділяється 2 години. Ці роботи проводяться замість лекційних занять. Вас буде заздалегідь попереджено про проведення контрольної роботи.

- **захист лабораторних робіт (60%):**

10 балів x 6 робіт = **60** балів

Захист лабораторних робіт складається із демонстрації коду та дієздатності програми, а також короткої співбесіди. На кожну лабораторну роботу виділяється кілька занять.

Лабораторні роботи мають дедлайни як описано вище.

- **презентація семестрового проекту (10%):**

10 балів x 1 заняття = **10** балів

Семестровий проект є сумарним результатом виконання усіх лабораторних робіт та представляє собою програмний продукт із завершеним функціоналом, визначеним студентами та викладачем на першому лабораторному занятті. Презентація відбувається у вигляді короткої (до 5 хв) доповіді та демонстрації роботи програми.

**Календарний контроль:** проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу. Календарний контроль проводиться за результатами лабораторних занять, проведених на момент початку контролю.

**Семестровий контроль:** залік.

Залікова оцінка виставляється на основі семестрового рейтингу, або — за бажанням студента чи при семестровому рейтингу менше 60 балів — за результатами написання залікової контрольної роботи.

Залікова контрольна робота складається із 4 практичних питань (по 1 питанню про HTML/CSS, JavaScript, AJAX-запити та REST API), що оцінюються у 25 балів кожне.

Умови допуску до семестрового контролю: мінімальний рейтинг не нижче 25 балів та виконання усіх лабораторних робіт.

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)

- додатки до силабусу — приклади залікової контрольної роботи, положення про PCO;
- зазвичай залік проходить на останньому занятті з дисципліни, відповідно, хард дедлайн буде за тиждень до цього;
- у випадку проведення курсу дистанційно, результати контролю (контрольні та залікові роботи) мають бути виконані в цифровому вигляді: як текстові файли, чи файли з кодом, або за неможливості — у вигляді розбірливих фото. Усі такі матеріали мають бути завантажені на указаний викладачем ресурс у терміни відведені під відповідний тип контролю. Ми надаватимемо буферні 5 хвилин на випадок форс-мажорних подій. У окремих випадках залікова контрольна може бути проведена у формі співбесіди. Аналогічно, окремі форми захисту можуть проводитися заочно, через листування чи месенджери.

**Робочу програму навчальної дисципліни (силабус):**

**Складено** асистентом Борисенком Павлом Борисовичем

**Ухвалено** кафедрою ПМА (протокол № 7 від 09.02.2022)

**Погоджено** Методичною комісією факультету прикладної математики (протокол № 6 від 25.03.2022)