



ВЕБТЕХНОЛОГІЇ ВІЗУАЛІЗАЦІЇ ДАНИХ

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 (2) та лабораторні (7) 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, pavlo.borysenko@gmail.com</i> Практичні: <i>Борисенко Павло Борисович, pavlo.borysenko@gmail.com;</i> <i>Громова Вікторія Вікторівна, vikvikgrom@gmail.com</i>
Розміщення курсу	Канали #web-course та #web-labs у кафедральному Slack Матеріали: https://drive.google.com/drive/folders/1Ohhz_8yNXob5UMRT2OYKzPDN2dKNKDb0

Програма навчальної дисципліни

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

Веброзробка — важливий розділ у сучасній розробці програмного забезпечення. Розуміння принципів front-end та back-end розробки дозволяє проектувати стабільні та підтримувані вебзастосунки, які становлять важливу частину сучасної користувацької екосистеми. З точки зору науки про дані та математичного моделювання, вебпрограмування загалом важливе в першу чергу як джерело даних та як засіб візуалізації та представлення цих даних користувачу.

В рамках цього курсу ми коротко розглянемо основні концепції веброзробки, що дозволять вам легше орієнтуватися у сучасних веб-технологіях та краще зрозуміти зв'язок між ними та рештою програмування, а також поговоримо про системи та засоби візуалізації даних, та їх використання в задачах науки про дані.

А на лабораторних роботах ви зможете з нуля опанувати мову програмування JavaScript — класичну мову програмування на клієнті, — та в комбінації із Python використати її для візуалізації різноманітних датасетів з використанням бібліотек Seaborn, Plotly та D3.

Курс «Вебтехнології візуалізації даних» є вибірковою підготовкою 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;
- принципів візуалізації даних та роботи із бібліотеками Seaborn, Plotly та D3.

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

Після засвоєння дисципліни студенти матимуть наступні:

- загальні компетентності:
 - ЗК1. Здатність учитися і оволодівати сучасними знаннями.
 - ЗК3. Здатність застосовувати знання у практичних ситуаціях.
 - ЗК6. Здатність до абстрактного мислення, аналізу та синтезу.
 - ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
 - ЗК8. Знання та розуміння предметної області та розуміння професійної діяльності.
- фахові компетентності:
 - ФК3. Здатність обирати та застосовувати математичні моделі для розв'язання прикладних задач, моделювання, аналізу, проектування, керування, прогнозування, прийняття рішень.
 - ФК4. Здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію.
 - ФК5. Здатність проектувати бази даних, інформаційні системи та ресурси.
 - ФК8. Здатність використовувати сучасні технології програмування та тестування програмного забезпечення.
 - ФК13. Здатність зрозуміти постановку завдання, сформульовану мовою певної предметної галузі, здійснювати пошук та збір необхідних вихідних даних.
 - ФК14. Здатність сформулювати математичну постановку задачі, спираючись на постановку мовою предметної галузі, та обирати метод її розв'язання, що забезпечує потрібні точність і надійність результату.
- уміння та навички:
 - РН4. Використовувати математичний опис, аналіз та синтез дискретних об'єктів та систем, використовуючи поняття й методи дискретної математики та теорії алгоритмів.
 - РН11. Уміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символьних алгоритмів.

Це досягається серед іншого завдяки опануванню:

- знання:
 - синтаксису та принципів роботи HTML та CSS;
 - синтаксису мови JavaScript;
 - про принципи клієнт-серверної взаємодії;
 - базових принципів роботи фреймворку Flask;
 - принципів побудови та ефективної обробки даних із різних джерел;
 - патернів веб-розробки;
 - принципів роботи з REST API;
 - бібліотек Seaborn, Plotly та D3;
- уміння:
 - конструювати і деконструювати веб-сторінки;
 - організовувати роботу із зовнішними джерелами даних;

- використовувати патерни веб-розробки та підходи разом з іншими парадигмами;
- систематизувати, підготувати та візуалізувати дані;
- навички:
 - створення та стилювання веб-сторінок;
 - роботи з Flask;
 - роботи із бібліотеками візуалізації Seaborn, Plotly та D3;
 - підготовки та передачі датасетів у вебзастосунках;
- досвід:
 - візуалізації даних мовами Python та JavaScript;
 - використання Flask;
 - використання бібліотек Seaborn, Plotly та D3;
 - 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:

- 3019. Програмування мовою Python.
- 3020. Програмування.
- 3022. Алгоритми і структури даних.

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. Принципи візуалізації даних.

Розділ 2. Front-end

Тема 2.1. HTML і CSS. Оформлення вебсторінок.

Тема 2.2. Вступ до JavaScript. Типи, функції та об'єкти. JSON.

Тема 2.3. D3. Використання різноманітних графічних компонентів.

Тема 2.4. Plotly. Динамічні графіки.

Розділ 3. Back-end

Тема 3.1. Flask. Базові конструкції та типи. Генерація сторінок і шаблонізація.

Тема 3.2. Seaborn та matplotlib. Локальна візуалізація даних. Jupyter-записники.

Тема 3.3. Передача даних від сервера до клієнта за допомогою Flask.

Тема 3.4. Розгортання застосунку онлайн.

4. Навчальні матеріали та ресурси

Підручники

Ви можете обрати один з наступних підручників:

1. Edward R. Tufte, *The Visual Display of Quantitative Information*, 2 ed., 2001.
Англійська версія: https://www.edwardtufte.com/tufte/books_vdqi
Це класичний підручник з візуалізації даних.
Усі розділи обов'язкові — це не дуже велика книга.
2. Alexandru Telea, *Data Visualization: Principles and Practice*, 2 ed., 2014.
Англійська версія:

https://www.researchgate.net/publication/220691300_Data_Visualization_Principles_and_Practice

Це підручник більш сучасний та більше орієнтований на веб і комп'ютерну репрезентацію даних.

Розділи 1-8 — обов'язкові. Розділи 9-12 — рекомендовані.

Крім того, ми рекомендуємо в якості додаткового читання:

3. Miguel Grinberg, *Flask Web Development*, 2 ed., 2018.

Англійська версія:

<https://www.oreilly.com/library/view/flask-web-development/9781491991725/>

Це доступний підручник з Flask, що на доступних прикладах розбирає основні стратегії і патерни веб-розробки.

Розділи 1-4 — обов'язкові. Розділ 5, 7-8, 14, 17 — рекомендовані.

4. Jack Chan et al., *Python API Development Fundamentals*, 2019.

Англійська версія:

<https://www.packtpub.com/product/python-api-development-fundamentals/9781838983994>

Це хороший практичний підручник із побудови REST API на Python.

Розділи 1-2, 4-5 — обов'язкові. Розділи 3 і 7-10 — рекомендовані.

Додаткові матеріали

1. Документація Seaborn.

Посилання: <https://seaborn.pydata.org/tutorial.html>

2. Документація D3.

Посилання: <https://github.com/d3/d3/wiki>

3. Документація Plotly.

Посилання: <https://plotly.com/javascript/getting-started/>

4. Документація Flask.

Посилання: <https://flask.palletsprojects.com/en/2.1.x/user-s-guide>

5. 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.

6. W3School.

Посилання: <https://www.w3schools.com/>

Це набір туторіалів та гайдів з багатьох базових речей.

Рекомендовані онлайн-курси

1. Codecademy. *Visualize Data with Python*.

<https://www.codecademy.com/learn/paths/visualize-data-with-python>

2. Coursera. *Data Visualization with Plotly Express*.

<https://coursera.org/projects/data-visualization-plotly-express>

3. Coursera. *Python for Data Visualization: Matplotlib & Seaborn*.

<https://www.coursera.org/projects/python-for-data-visualization-seaborn>

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. Принципи візуалізації даних. Подання інформації за Тафтом. Доступність і густина інформації.

Розділ 2. Front-end

Лекція 4. HTML розмітка та каскадні таблиці стилів (CSS). Принцип відділення семантики і представлення. Існуючі шаблони базової візуалізації (Tufte-CSS).

Лекція 5. Вступ до JavaScript. Типи, функції і об'єкти. JSON.

Лекція 6. D3. Використання графічних компонентів. Візуалізація JSON-даних.

Лекція 7. Plotly. Динамічні графіки. Отримання користувацького вводу. Додавання інтерактивних елементів.

Розділ 3. Back-end

Лекція 8. Flask. Базові типи даних. Генерація сторінок і шаблонізація. Мова шаблонів Jinja.

Лекція 9. Seaborn та matplotlib. Локальна візуалізація даних.

Лекція 10. Jupyter-записники. Використання записників для препроцесингу даних. Хостинг записників онлайн. Google Collab.

Лекція 11. Передача даних від сервера до клієнта за допомогою Flask. REST-архітектура.

Лекція 12. Розгортання застосунку онлайн.

Лабораторні роботи

Розділ 1. Базові принципи веб-розробки

Лабораторна 0. Розробка структури застосунку.

Завдання: вибрати та узгодити тематику та набір даних для візуалізації.

Розділ 2. Front-end

Лабораторна 1. Принципи семантичного дизайну та розмітки. Каскадні таблиці стилів. D3.

Завдання: реалізувати вебсторінку та підключити до неї бібліотеку D3. Візуалізувати обраний датасет (його елементи) використовуючи 3 базові типи графіків.

Лабораторна 2. Динамічні графіки та інтерактивні елементи з використанням Plotly.

Завдання: підготувати програму мовою Python, що візуалізує обраний датасет (подібно попередній лабораторній) за допомогою Plotly. Побудувати вебсторінку, що використовує Plotly для аналогічної візуалізації. Реалізувати можливість динамічно взаємодіяти чи змінювати графіки (вводити межі, варіювати тип візуалізації, переглядати деталі окремих точок даних тощо).

Розділ 3. Back-end

Лабораторна 3. Flask.

Завдання: створити Flask-застосунок, що реалізує попередньо створені веб-сторінки за допомогою шаблонів. Реалізувати систему роутингу.

Лабораторна 4. Seaborn.

Завдання: реалізувати Jupyter-записник із використанням бібліотеки Seaborn. Візуалізувати обраний датасет подібно до попередніх робіт. Порівняти засоби динамічної взаємодії із веб-версією.

Лабораторна 5. Використання даних з серверу.

Завдання: побудувати Flask-застосунок, що повертає вебсторінку та дані за запитами до різних URL. Реалізувати доступ до цих даних з «клієнтської» сторінки, та візуалізувати їх використовуючи довільну бібліотеку.

Лабораторна 6. Розгортання застосунку.

Завдання: розгорнути отримані застосунки онлайн. Розгорнути Jupyter-записник із використанням Google Collab.

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.*
 - *Розширені можливості конфігурування Flask.*
 - *Додаткові види графіків.*

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 прездати за погодженням із викладачем.

Оцінювання

Кожне завдання у практичних, домашніх практичних та контрольних роботах оцінюється у фіксовану кількість балів, що явно вказані у формулюванні чи назві завдання.

Максимальну кількість балів студент отримує у випадку, якщо він навів повний та правильний розв'язок або припустився несуттєвих помилок, які не вплинули на нього.

Меншу ніж максимальну кількість балів студент отримує у випадку, коли наведений ним розв'язок є правильним, проте неповним (задача не розв'язана до кінця), або хід розв'язку є правильним та повним, проте студент припустився помилок, які суттєво вплинули на відповідь.

Нуль балів студент отримує у випадку, коли задача взагалі не розв'язана, або наведений хід розв'язку містить грубі помилки, або наведено тільки відповідь на задачу (окрім випадків, коли завдання чітко вимагає лише відповіді).

Окрім правильності розв'язку у окремих випадках (вони також будуть визначені явно) буде оцінюватися чистота коду, адекватність алгоритму та уміння користуватися інструментами розробки (наприклад, уміння завантажувати роботи на віддалений репозиторій git).

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 Unicheck 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 (10%):**

5 балів x 2 роботи = **10** 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 (70%):**

10 балів x 7 робіт = **70** 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 (20%):**

20 балів x 1 заняття = **20** балів

Семестровий проект є сумарним результатом виконання всіх 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, Plotly, Flask та Seaborn), що оцінюються у 25 балів кожне.

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

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

<i>Кількість балів</i>	<i>Оцінка</i>
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 — теоретичні та практичні матеріали для лабораторних робіт, 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 № 10 від 02.01.2024)

Погоджено Методичною комісією факультету прикладної математики (протокол № 6 від 26.01.2024)