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

Кафедра прикладної математики

ПОБУДОВА REST-СЕРВІСІВ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни	
Рівень вищої освіти	***Перший (бакалаврський)***
Галузь знань	*11 Математика та статистика*
Спеціальність	*113 Прикладна математика*
Освітня програма	*Наука про дані та математичне моделювання*
Статус дисципліни	*Вибіркова*
Форма навчання	*очна(денна)*
Рік підготовки, семестр	*3 курс, весняний семестр*
Обсяг дисципліни	*4 кредити*
Семестровий контроль/ контрольні заходи	*Залік/Контрольні (2) та лабораторні (6) роботи*
Розклад занять	*Лекції — 1 раз на тиждень (18 лекцій),* *лабораторні — 1 раз на тиждень (18 занять)*
Мова викладання	*Українська/Англійська*
Інформація про керівників курсу / викладачів	Лектор: *Борисенко Павло Борисович, pavlo.borysenko@gmail.com* Практичні: *Борисенко Павло Борисович, pavlo.borysenko@gmail.com;* *Громова Вікторія Вікторівна, vikvikgrom@gmail.com*
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 розглянемо основні концепції веброзробки, 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, — на прикладі фреймворку React від Facebook та використати його для комунікації із REST-сервісом, що надає ваші дані користувачу, який ви побудуєте за допомогою Python та фреймворку Flask.

Курс «Побудова REST-сервісів» є вибірковим 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 та фреймворку React,*
- *принципів та підходів до побудови клієнт-серверних систем з точки зору сервера;*
- *принципів побудови та роботи з REST API за допомогою фреймворку Flask.*

Предметом вивчення є програмний код мовою JavaScript та його синтаксичні примітиви; структури даних та примітиви фреймворку Flask; 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 Flask;*
  - *принципів побудови та ефективної обробки даних із різних джерел;*
  - *патернів веб-розробки;*
  - *принципів роботи з REST API;*
- *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;*
  - *створення та роботи з REST API;*
  - *роботи з Flask та React;*
  - *роботи зі слабко типізованою мовою програмування;*
- *досвід:*
  - *програмування REST-застосунків мовою Python;*
  - *використання Flask;*
  - *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.*

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.

Розділ 2. Front-end: JavaScript та React

Тема 2.1. Семантична модель документу. HTML і об'єктна модель документа (DOM). Стильове оформлення документа і CSS.

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

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

Тема 2.4. React.

Розділ 3. REST-сервіси

Тема 3.1. Flask. 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 Flask.

Тема 3.4. Використання React компонентів для отримання даних від REST-сервісів.

Тема 3.5. Розгортання застосунку онлайн.

4. Навчальні матеріали та ресурси

Підручники

Ви можете обрати один з наступних підручників:

1. *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 веброзробки.
   Розділи 1-4, 7-8, 14, 17 — обов'язкові. Розділ 5 — рекомендований.
2. *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, 7-8, 10 — обов'язкові. Розділи 3 і 10 — рекомендовані.

Крім того, ми рекомендуємо в якості додаткового підручника:

3. *Alex Banks, Eve Porcello, Learning React, 2017.*
   Англійська версія: https://www.oreilly.com/library/view/learning-react/9781491954614/
   Це найбільш доступний за викладом підручник з React.
   Розділи 2-7, 11-12 — обов'язкові. Розділи 8-10 — рекомендовані.

Додаткові матеріали

1. *Документація Flask.*
   Посилання: https://flask.palletsprojects.com/en/2.1.x/ - user-s-guide
2. *Документація MDN.*
   Посилання: https://developer.mozilla.org/en-US/
3. *Документація React.*
   Посилання: https://uk.reactjs.org/
   Тут же є і хороший курс для початківців.
4. *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.
5. *W3School.*
   Посилання: https://www.w3schools.com/
   Це набір туторіалів 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.

Розділ 2. Front-end: JavaScript та React

Лекція 3. Семантична модель документу. HTML розмітка та об'єктна модель документу (DOM). Принципи рендерингу веб-сторінок. Стильове оформлення документа. Каскадні таблиці стилів (CSS). Принцип відділення семантики і представлення. Невізуальні елементи.

Лекція 4. Вступ до JavaScript. Типи, функції і об'єкти. JSON. Особливості роботи зі слабко-типізованими мовами. Вказівник this. Класи. Маніпуляції DOM. Події та таймери. Обробка вводу від користувача.

Лекція 5. AJAX: XMLHttpRequest і fetch. Асинхронна взаємодія із сервером. Веб-сокети. Axios.

Лекція 6. Встановлення React і Babel. Компонентна модель. Типові патерни веб-розробки.

Розділ 3. REST-сервіси.

Лекція 7. Flask. Базові типи даних. Генерація сторінок і шаблонізація. Мова шаблонів Jinja.

Лекція 8. Роутинг. Принципи поділу REST-сервісів. Використання методів HTTP.

Лекція 9. Автентифікація користувачів. Хеші і cookies. Валідація даних і обробка помилок. Принципи роботи з користувацькими даними.

Лекція 10. Санітизація вводу користувача. Небезпеки використання сирого вводу. Вектори SQL-атак. XSS.

Лекція 11. Побудова та використання REST API на основі React. Принципи мікросервісних архітектур.

Лекція 12. Використання React компонентів для отримання даних від REST-сервісів.

Лекція 13. 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).

Розділ 2. Front-end: JavaScript та React

Лабораторна 1. Принципи семантичного дизайну та розмітки. Каскадні таблиці стилів. Таблична та сіткова верстка. Маніпуляції об'єктною моделлю документу. Події і обробка вводу.

Завдання: реалізувати семантичну модель кількох елементів інтерфейсу веб-застосунку (сторінок чи окремих компонентів) за допомогою HTML-розмітки. Реалізувати стильове оформлення за допомогою каскадної таблиці стилів. 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 2. React та віртуальна DOM. AJAX та асинхронне програмування. Споживання інформації зі сторонніх REST-сервісів.

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. REST-сервіси.

Лабораторна 3. Flask.

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

Лабораторна 4. Верифікація даних. Автентифікація.

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

Лабораторна 5. REST API.

Завдання: побудувати REST API як окремий Flask-застосунок. Реалізувати доступ до цього API з «клієнтського» Flask-застосунку, що генерує веб-сторінки. Розгорнути отримані сервіси онлайн.

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 та розробка з урахуванням потреб людей із обмеженими можливостями.*
    - *Розширені можливості конфігурування Flask.*

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 отримають половину балів;*
- *здані після* **хард дедлайну** *(тиждень до заліку) — не отримають 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 (РСО)

Поточний контроль:

- ***модульні контрольні роботи*** *(20%):*
  10 балів х 2 роботи = ***20*** *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 х 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*** *(20%):*
  20 балів х 1 заняття = ***20*** *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 РСО;*
- *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)