



# Силабус дисципліни «Серверні фреймворки веб-застосувань»

**Викладач:** Бурлаченко Іван Сергійович  
старший викладач кафедри комп'ютерної інженерії  
факультету комп'ютерних наук ЧНУ імені Петра Могили.

## Очікувані результати навчання

В результаті вивчення дисципліни студент

### *має знати:*

- об'єктно-орієнтовану модель мов JavaScript та PHP і особливості проектування програмних компонентів;
- основні архітектурні особливості програмного забезпечення, розробленого на Express та Laravel;
- як організувати декомпозицію коду проекту на складові компоненти для зручності подальшої підтримки та повторного використання;
- як вирішувати системні завдання під час web-розробки на рівні, достатньому для промислового застосування;

### *має вміти:*

- впроваджувати програмні залежності у конфігурацію Express та Laravel проектів;
- проектувати модель сервісів даних у Express та Laravel застосунках і налаштовувати web socket з'єднання;
- кешувати шаблони, відображати динамічні дані, створювати призначені для користувача функції і обробляти помилки під час виконання;
- використовувати і налаштовувати сесии для збереження даних між запитами;
- налаштовувати Express та Laravel застосунки для підвищення продуктивності і безпеки програмних продуктів;
- впровадити аутентифікацію і авторизацію, та запобігати SQL ін'єкціям, CSRF, XSS, перехопленню кліків.

**Обсяг:** буде визначено після здійснення процедури вибору студентами вибіркових дисциплін.

**Мета:** навчитися використовувати системний підхід до проектування архітектури великих програмних продуктів, працювати в командах back-end розробників.

## Оригінальність навчальної дисципліни

Матеріали дисципліни представляють сукупний досвід інженерії програмного забезпечення багатьох tech-лідерів проектів та solution-архітекторів міжнародних ІТ компаній.

## Зміст дисципліни

- Тема 1. Node.js. Основи розроблення Node.js застосунків.
- Тема 2. Node.js. Робота з модулями. Об'єкт global та глобальні змінні.
- Тема 3. Node.js. Передача параметрів застосунком. NPM. Package.json. Установка модулів. Визначення команд.
- Тема 4. Node.js. Nodemon. Асинхронність в Node.js.Робота з файлами.
- Тема 5. Node.js. Події. Stream. Pipe.
- Тема 6. Node.js. Серверна архітектура застосунків на Node.js. Створення сервера. Порти.
- Тема 7. Node.js. Шаблони проектів. Відправка файлів.
- Тема 8. Express. Початок роботи з Express. Конвеєр обробки запиту і middleware.
- Тема 9. Express. Відправлення відповіді. статичні файли. маршрутизація
- Тема 10. Express. Переадресація. Передача даних з застосунком. Параметри рядка запиту.
- Тема 11. Express. POST-запити і відправка форм. Параметри маршруту. Router
- Тема 12. Express. JSON і AJAX. Представлення і бібліотека формування представлення Handlebars. Часткові представлення в Handlebars.
- Тема 13. Express. Layout в Handlebars. Хелпери в Handlebars. Шаблонізатор EJS.

Тема 15. Express. Робота з MySQL в Express. Sequelize.

Тема 16. Express. Підключення до MySQL. Виконання запитів до MySQL.

Тема 17. Express. Promise API. Основні операції з даними. Пули підключень.

Тема 18. Express. означення моделей даних. Запити до бази даних. Sequelize і Express.

Тема 19. Express. в'язок один-до-багатьох. Зв'язок один-до-одного. Зв'язок багато-до-багатьох.

Тема 20. Express. Робота з MongoDB у Node.js. Функції Mongoose.

Тема 21. Express. Додавання і отримання даних в MongoDB. Видалення документів в MongoDB. Оновлення документів в MongoDB.

Тема 22. Express. Mongoose. Визначення схеми в Mongoose. CRUD в Mongoose. Express і Mongoose.

Тема 23. Express. MEAN архітектура для Node.js застосунків.

Тема 24. Express. Написання REST API: організація взаємодії з MongoDB.

Тема 25. Express. Впровадження REST API: використання API з внутрішньої програми Express.

Тема 26. Express. Angular. Розширення браузерних застосунків на основі шаблону (MVC)

Тема 27. Express. MERN архітектура для Node.js застосунків

Тема 28. Express. Використання утиліт: bodyParser. dotenv. passport.js. validator. nodemon

Тема 29. Express. Впровадження REST API: використання API з внутрішньої програми Express.

Тема 30. Express. Розширення браузерних застосунків на основі архітектурних шаблонів React.

Тема 31. PHP 7. Змінні, Константи, Типи даних. Цикли і управління циклами.

Тема 32. PHP 7. Функції. Декларація, виклик і повернення значень.

Тема 33. PHP 7. ООП. Парадигми ООП. Реалізація SOLID

Тема 14. Express. Шаблонізатор Pug. Часткові представлення і layout в Pug.

Тема 34. PHP 7. Абстрактні класи. Інтерфейси об'єктів. Трейти.

Тема 35. PHP 7. Анонімні класи. Перевантаження. Ітератори об'єктів. Магічні методи.

Тема 36. PHP 7. Клонування об'єктів. Порівняння об'єктів. Пізні статичне зв'язування.

Тема 37. PHP 7. Об'єкти і посилання. Сериалізація об'єктів. Коваріантність і контраваріантність.

Тема 38. PHP 7. Composer. Основні команди. Робота з сесіями (сеансами). Суперглобальні змінні.

Тема 39. Laravel. Встановлення та налаштування фреймворку. Homestead. Valet.

Тема 40. Laravel. Маршрутизація. Middleware. Контролери. Запити і введення

Тема 41. Laravel. Http відповіді. Переадресація запитів. CSRF-захист. Структура відображення даних у Views.

Тема 42. Laravel. Постачальники послуг (Service providers).Сервіс-контейнери/

Тема 43. Laravel. Контракти. Фасади. Етапи виконання запитів.

Тема 44. Laravel. Архітектура модульних проєктів. Патерни Laravel проєктів.

Тема 45. Laravel. сервіси. Аутентифікація. Авторизація. Оплата (білінг)

Тема 46. Laravel. сервіси. Кешування. Колекції. Командна шина.

Тема 47. Laravel. сервіси. Розширення фреймворку. Elixir. Шифрування.

Тема 48. Laravel. сервіси. Envoy. Scout. Журналювання помилок.

Тема 50. Laravel. сервіси. Події. Файлова система і хмари.

Тема 51. Laravel. сервіси. API Authentication (Passport). Хешування.

Тема 52. Laravel. Функції. Локалізація. Робота з e-mail.

Тема 53. Laravel. Розроблення пакетів. Пагінація. Черги.

Тема 54. Laravel. Сесії. Шаблони Blade. Планувальник. Модульні тести. Перевірка введення.

Тема 55. Laravel. Інтеграція JavaScript і CSS у Laravel проєкт.

Тема 56. Laravel. Бази даних. Основи використання. Конструктор запитів.Конструктор таблиць.

Тема 57. Laravel. Міграції. Seeding. Redis. Eloquent ORM. Відношення. Колекції та перетворювачі.

Тема 58. Laravel. Сериалізація. Artisan. Основи командного інтерфейсу та створення команд.

принципів на РНР. Огляд патернів проектування.

### **Пререквізити**

Програмування, об'єктно-орієнтоване програмування, алгоритми і структури даних

### **Постреквізити**

інженерія програмного забезпечення, крос-платформне програмування, web-технології, технології розробки мобільних застосунків, паралельне програмування, методи машинного навчання, формальні граматики та компілятори, розпізнавання образів, системи та методи прийняття рішень.

**Семестровий контроль:** залік/іспит

### **Оцінювання:**

За семестр: 70/60 балів

За залік/іспит: 30/40 балів

### **Види робіт:**

Лабораторні роботи – 60/50 балів (12/10 x 5 б.)

Індивідуальне проєктне завдання – 10 балів.

### **Технічне забезпечення**

Modern browsers

VS Code IDE

Moodle, Google Meet, Zoom, Webex, профілі соціальних мереж

### **Політика щодо дедлайнів**

Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку.

### **Політика щодо академічної доброчесності**

Передбачає самостійне виконання лабораторних робіт та індивідуального проєктного завдання. Списування під час заліку (в т. ч. із використанням мобільних пристроїв) заборонено. У разі виявлення плагіату або списування роботи не зараховуються.

### **Критерії оцінювання лабораторних робіт**

- вміння самостійно вирішувати поставлені задачі та представляти рішення у вигляді оптимального програмного коду.
- вміння користуватися специфічними інструментальними засобами для створення програмного забезпечення, що задовільняє SOLID принципам.
- вміння лаконічно та логічно формувати відповіді на запитання, пов'язані з виконаними лабораторними роботами.

### **Критерії оцінювання індивідуального проєктного завдання**

- формулювання актуальності, проблеми, мети і завдань, практичного значення;
- вміння самостійно створювати оптимальний програмний код для вирішення поставленої задачі з використанням паттернів програмування.;
- вміння користуватися специфічними інструментальними засобами для розробки ПЗ з ефективною архітектурою;
- якість доповіді студента (форма доповіді, зміст, доказова база, висновки) ; повнота та логічність відповідей на поставлені питання.