

## Лабораторна робота №6

**Тема: Знайомство з Laravel. Встановлення, конфігурація, робота з БД та HTTP. Eloquent ORM, Auth, Login, Controllers, Schemas&Models, CRUD operations. Створення першого базового додатку на Laravel.**

**Завдання 1:** Встановити Laravel:

- Вивчити теоретичний матеріал: [Інсталяція Laravel](#).
- Перш ніж встановлювати Laravel, переконайтесь в тому, що на вашій локальній машині вже є [PHP](#), [Composer](#), та [the Laravel installer](#).
- Користуючись документацією, встановіть та запустіть Laravel на своїй локальній машині.

**Завдання 2:** Знайомство з конфігурацією та структурою додатка Laravel:

- Вивчити теоретичний матеріал: [Конфігурація Laravel](#).
- Користуючись документацією, сконфігуруйте ваш додаток, щоб він коректно працював на вашій машині. Додайте .env файл, та залиште в ньому декілька змінних (наприклад: порт додатку, DEBUG\_MODE, підключення до БД, логін root користувача, пароль root користувача і тд.)

**Завдання 3.1:** Знайомство з роботою з БД у Laravel:

- Вивчити теоретичний матеріал: [БД Laravel](#).

**Завдання 3.2:** Знайомство з роботою з БД у Laravel, використовуючи Eloquent ORM:

- Вивчити теоретичний матеріал: [Laravel & Eloquent ORM](#).
- Створіть БД та підключіть ваш додаток до неї, використовуючи функціонал ORM.

**Завдання 5:** Створення додатку:

Використовуючі набуті навички, створіть ваш перший додаток Laravel, опираючись на наступні вимоги:

1. Базова авторизація в Laravel: можливість входу як адміністратор.

2. Використання сідів бази даних для створення першого користувача з електронною адресою **admin@admin.com** та паролем "password".
3. Функціонал CRUD (Створення / Перегляд / Оновлення / Видалення) для двох елементів меню: **Users** та **Products**.
4. Таблиця **Users** в базі даних містить такі поля: ПІБ(обов'язково), електронна пошта, пароль, вік, роль, телефон.
5. Таблиця **Products** в базі даних містить такі поля: Назва (обов'язково), категорія(List), ціна, опис, зображення(мінімум 100x100).
6. Використайте міграції бази даних для створення зазначених схем.
7. Зберігання зображень продуктів у папці **storage/app/public** та забезпечення доступу до них з публічної директорії.
8. Використання базових [Laravel-ресурс-контролерів](#) з методами за замовчуванням – **index, create, store** тощо.
9. Використовувати [Laravel starter kit для авторизації та базової теми](#), але видалити можливість реєстрації.

### **DB relationships. Validation. Pagination. SMTP. Laravel frontend customization.**

1. Додати таблицю **Invoices** в базу даних, яка містить такі поля: ID(auto-increment), UserID(ключ до **Users**), ProductID(ключ до **Products**), дата, к-ть товару, сума. Використайте міграції бази даних для створення зазначеної схеми.
2. Використання функції валідації Laravel через [Request-класи](#).
3. Використання пагінації Laravel для відображення списків Користувачів/Продуктів/Накладних — по 10 записів на сторінку.
4. Використання бібліотеки [Datatables.net](#) для відображення таблиць – з або без рендерингу на сервері.
5. Email-сповіщення: надсилання листа при створенні нового користувача, використовуючи SMTP (використання Mailgun або Mailtrap).
6. Зробити проєкт багатомовним (використовуючи папку **resources/lang**).
7. Використання більш складної теми інтерфейсу, наприклад, **AdminLTE**.\*