

## Вибір рішення для адмін-панелі

### AdminBro

Плюси:

- Інтуїтивний інтерфейс та широкий функціонал для створення адмін-панелі.
- Підтримка різних типів полів, включаючи текст, числа, зображення тощо.
- Можливість налаштування прав доступу.

Мінуси:

- Обмежена кастомізація інтерфейсу в порівнянні з іншими рішеннями.

### Strapi

Плюси:

- Повноцінна CMS з гнучким API, що відповідає REST або GraphQL.
- Зручна адмін-панель з можливістю розширення функціоналу.
- Підтримка ролями і правами доступу.

Мінуси:

- Складність налаштування та кастомізації може бути високою для складних проектів.

### Keystone.js

Плюси:

- Гнучка архітектура для створення власних моделей та розширень.
- Підтримка аутентифікації, авторизації та валідації даних.
- Можливість додавати власні React-компоненти до адмін-панелі.

Мінуси:

- Може бути складнішим для новачків у порівнянні з іншими рішеннями.

Мій вибір для адмін-панелі: Strapi. Він надає гнучку CMS та адмін-панель разом з можливістю налаштування прав доступу. Крім того, Strapi підтримує різні бази даних і має активну спільноту, що полегшує розробку.

## Вибір технології обробки завантаження файлів на веб-сервер:

### express-fileupload

Плюси:

- Простий у використанні і не вимагає додаткових налаштувань.
- Легко інтегрується з Express.
- Підтримує завантаження файлів через форми та API запити.

Мінуси:

- Обмежена функціональність порівняно з деякими іншими бібліотеками, такими як Multer.

## **formidable**

Плюси:

- Може обробляти завантаження файлів, а також розпарсити дані форми.
- Добре справляється з великими завантаженнями через стрімінгову обробку.

Мінуси:

- Деякою мірою менше інтуїтивний у порівнянні з іншими бібліотеками.
- Потребує додаткових зусиль для обробки файлів і даних форми окремо.

## **Multer**

Плюси:

- Має багато налаштувань і опцій для роботи з завантаженням файлів, включаючи обмеження типів файлів, розмірів і т.д.
- Підтримує зберігання файлів у різних місцях, включаючи локальний диск і обліковий запис Amazon S3.
- Легко інтегрується з Express і надає зручний спосіб обробки завантаження файлів.

Мінуси:

- Може здаватися складним для початківців або для використання в простих випадках, де потрібно лише базове завантаження файлів

## **Вибір технології для обробки зображень на сервері:**

### **Sharp**

Плюси :

- Швидкість: Sharp славиться своєю високою швидкістю обробки зображень завдяки використанню оптимізованих алгоритмів.
- Багатофункціональність: Sharp надає широкий спектр функцій для обробки зображень, включаючи зміну розміру, обрізку, редакцію метаданих, роботу з кольорами і багато інших операцій.
- Підтримка багатьох форматів: Sharp підтримує багато різних форматів зображень, що дозволяє працювати з різними типами файлів.
- Легка інтеграція з Express.js: Ви можете легко інтегрувати Sharp в свій проект Express.js, що робить його дуже зручним для обробки зображень на сервері.

Мінуси:

- Не підтримує інші типи медіа: Sharp спеціалізується на зображеннях і не підтримує обробку відео або інших типів медіа-файлів.
- Складний синтаксис для деяких операцій: Деякі розширені операції можуть вимагати складного синтаксису, що може бути складно читати та розуміти.

## Jimpr

Плюси:

- Простота використання: Jimpr має простий і зрозумілий API, що робить його досить легким для початківців та швидким для розробників.
- Легка інтеграція: Ця бібліотека також легко інтегрується з Express.js і іншими популярними веб-фреймворками.

Мінуси:

- Швидкість: У порівнянні з Sharp, Jimpr може бути повільнішим, особливо при обробці великих зображень.
- Менша підтримка форматів: Jimpr підтримує менше форматів зображень порівняно з Sharp. ImageMagick та GraphicsMagick:

**ImageMagick і GraphicsMagick** (дві різні бібліотеки, але вони схожі за функціональністю та плюсами/мінусами.)

Плюси:

- Повна функціональність: Вони надають широкий спектр можливостей для обробки зображень та підтримують багато форматів.
- Підтримка відео і інших медіа: ImageMagick і GraphicsMagick можуть обробляти не лише зображення, але і відео та інші типи медіа-файлів.

Мінуси:

- Великий обсяг пам'яті і ресурсів: Вони можуть споживати багато оперативної пам'яті та CPU-часу при обробці великих файлів або великої кількості файлів одночасно.
- Складний синтаксис: Обидві бібліотеки можуть бути складними для використання через свій складний синтаксис, особливо для початківців.
- Вимога до інсталяції на сервері: Вам потрібно буде інсталювати ці бібліотеки на сервері, що може бути складним завданням для деяких хостинг-провайдерів.

Обираючи бібліотеку для обробки зображень, важливо враховувати особливості проекту. Якщо завдання системи не передбачає активної та гнучкої обробки багатьох форматів, то у більшості випадків буде достатньо скористатися Sharp.

## Вибір рішень для керування медіа-файлами

### Amazon S3 / AWS Amplify Storage

Плюси:

- Висока доступність і масштабованість.
- Підтримка CDN для швидкого розповсюдження медіа-файлів.
- Широкий функціонал для роботи з файлами.

Мінуси:

- Вимагає інтеграції з AWS, що може бути складним для новачків.

### Cloudinary

Плюси:

- Легка інтеграція з готовими бібліотеками для багатьох платформ.
- Зручні інструменти для оптимізації та обробки зображень.
- Підтримка CDN та автоматичної масштабування.

Мінуси:

- Вартість використання після певного обсягу завантажень.

### Filestack

Плюси:

- Простий API для завантаження та обробки файлів.
- Підтримка багатьох джерел завантаження (URL, Dropbox, Google Drive тощо).
- Гнучкість в конфігурації та вартість користування.

Мінуси:

- Може бути обмеженим для великих обсягів файлів.

Важко обрати конкретну технологію, вибір дуже сильно залежить від виду та складності проекту. Однак, якщо на проєкті вже використовуються сервіси Amazon AWS, то краще зупинити свій вибір на рішеннях які пропонує компанія Amazon. В іншому випадку, краще обрати Cloudinary.

## Дод. завдання

У кожному проєкті, я б використовував би такі пакети:

- Eslint, prettier - для контролю якості коду та відповідності встановленим правилам форматування.
- Swagger - для ведення документації API
- Morgan, Winston – для логування.
- class-validator, express-validator для валідації полів запиту.

