# ЗВІТ: Етап 4 - Порівняльний аналіз розробки

**Варіант:** 88  
**Порівнювані версії:** Десктоп (Electron) vs Веб (Browser)

## 1. Порівняльна таблиця

| **Критерій** | **Десктоп (Electron)** | **Веб (Browser)** |
| --- | --- | --- |
| **Розробка** |  |  |
| Складність | Середня (Electron + React) | Проста (тільки React) |
| Розмір коду | +15% (Electron код) | Базовий |
| Час розробки | Більше | Менше |
| **Функціонал** |  |  |
| Основні операції | ✓ Всі | ✓ Всі |
| Синхронізація папок | ✓ Так | ✗ Немає |
| Доступ до локальної ФС | ✓ Повний | ✗ Обмежений |
| Офлайн робота | ✓ Можлива | ✗ Потрібен інтернет |
| **Деплоймент** |  |  |
| Розмір дистрибутива | ~180MB | ~3MB |
| Встановлення | Завантажити + встановити | Відкрити URL |
| Оновлення | Нова версія | Автоматичне (F5) |
| Крос-платформність | Win/Mac/Linux окремо | Будь-який браузер |
| **Продуктивність** |  |  |
| Швидкість запуску | 3-5 секунд | <1 секунда |
| Споживання пам'яті | 200-400MB | 50-150MB |
| **Користувацький досвід** |  |  |
| Доступність | Треба встановлювати | Миттєва |
| Інтеграція з ОС | Повна | Мінімальна |

## 2. Спільні елементи

**Backend (100% спільний):**

* Spring Boot REST API
* Всі контролери, сервіси, репозиторії
* JWT автентифікація
* PostgreSQL база даних
* File storage на диску сервера
* Бізнес-логіка сортування/фільтрації

**Frontend (85% спільний):**

* React компоненти: App, Login, Register, FileList, FileViewer
* HTTP клієнт (api.js)
* Логіка UI
* Стилі та layout

**Архітектура:**

* REST API комунікація
* JWT токени в localStorage
* Однакові HTTP endpoints
* Однакова обробка помилок

## 3. Відмінності

**Доступ до файлової системи:**

Десктоп: Electron preload API дає доступ до Node.js fs модуля. Може автоматично читати/записувати файли у будь-які папки. Використовується для синхронізації - програма сканує локальну папку та автоматично завантажує/вивантажує файли.

Веб: Браузер обмежений sandbox. Може тільки працювати з файлами через file input (користувач явно вибирає) та завантажувати через browser download. Не може автоматично зберігати у локальні папки.

**Packaging:**

Десктоп: electron-builder створює окремі інсталятори для Windows (.exe), macOS (.dmg), Linux (.deb/.rpm). Включає Chromium runtime - тому великий розмір.

Веб: npm run build створює статичні HTML/CSS/JS файли. Деплояться на будь-який веб-сервер або CDN.

**Оновлення:**

Десктоп: Потрібен механізм auto-update або користувач вручну завантажує нову версію. Складніше керувати версіями у різних користувачів.

Веб: При кожному refresh браузер завантажує останню версію з сервера. Всі користувачі миттєво отримують оновлення.

## 4. Переваги та недоліки

**Десктоп:**

Переваги:

* Повний доступ до файлової системи
* Синхронізація локальних папок
* Інтеграція з ОС (file associations, tray icons)
* Може працювати офлайн

Недоліки:

* Великий розмір дистрибутива (~180MB)
* Треба встановлювати
* Складніше оновлювати
* Окремі білди для кожної ОС

**Веб:**

Переваги:

* Миттєвий доступ (просто URL)
* Маленький розмір (~3MB)
* Автоматичні оновлення
* Працює на всіх платформах

Недоліки:

* Немає синхронізації папок
* Обмежений доступ до ФС
* Потрібен інтернет
* Менша інтеграція з ОС

## 5. Обґрунтування оптимальної технології

**Для даного проєкту (File Manager) оптимальне рішення: Веб-версія з REST API**

**Обґрунтування:**

1. **Цільова аудиторія:** Більшість користувачів потребують просто зберігати і ділитися файлами. Це ефективніше робити через веб.
2. **Use cases покриття:** Веб-версія покриває 90% сценаріїв використання. Синхронізація папок потрібна тільки невеликій кількості power users.
3. **Time-to-market:** Веб швидше розробляти та деплоїти. Можна швидко отримати feedback від користувачів.
4. **Вартість:** Веб дешевший у розробці та підтримці. Не треба підтримувати 3 платформи (Win/Mac/Linux).
5. **Масштабованість:** Легко додати мобільні клієнти. Один REST API для всіх платформ.

**Стратегія впровадження:**

* **MVP:** Веб-версія для швидкого виходу на ринок
* **Якщо є попит:** Додати Electron версію для синхронізації папок
* **Переваги підходу:** 85% коду спільний, мінімальні додаткові витрати

**Альтернатива:**

Якщо синхронізація папок - критична вимога, тоді десктоп стає пріоритетом. Але для типового file manager веб оптимальніший.

## Висновок

Обидві версії успішно реалізовані. Веб-версія є оптимальною для більшості сценаріїв через простоту доступу, легкість оновлення та нижчу вартість. Десктоп доцільний як додаткова опція для користувачів які потребують синхронізацію папок.