**Висновки по Electron**

**Плюси:**

* Дозволяє нам відкривати файли наступних форматів: .exe, .batch.
* Дозволяє відкривати google chrome, або інший браузер з параметрами (наприклад url).
* За допомогою js бібліотек дозволяє формувати jwt токени.
* За допомогою нього можна запускати як веб додатки так і exe додатки для Windows.

**Мінуси:**

* Для підключення до ORACLE database треба встановлювати додатково Oracle Instant Client вище 12 версії.
* На даний момент є проблеми з встановленням і обновленням самого фреймворку із-за налаштувань proxy.
* Не змогли поки спробувати зібрати тестовий додаток на Electron в файл .exe, оскільки для цього треба мати можливість встановлювати цей фреймворк, так як він є в залежностях пакету, який проводить збірку .exe файлу.

**Пропозиції**

Весь процес роботи користувача з порталом можна розділити на декілька кроків:

1. Запуск .exe файлу. Відкривається вікно з формою автентифікації користувача.
2. Після того, як користувач ввів коректний логін і пароль він потрапляє на персоналізовану сторінку порталу, де бачить набір додатків.
3. При натисненні на лінк, якщо це web додаток (формується url з токеном) і відбувається перехід на нове вікно браузера Chrome з вікном додатку, якщо це exe файл, то він просто запускається в ОС.

На даний момент є три варіанти побудови архітектури побудови порталу (головної сторінки). Така кількість варіантів пов’язана з специфікою роботи nodeJS додатків з базою даних ORACLE і підбором варіантів для того, щоб не треба було додатково встановлювати і налаштовувати Instant Client на стороні кожного користувача, а зробити це один раз на сервері.

**Варіант №1.**

* Electron + REST API service (nodeJS + express). Розробка модуля порталу безпосередньо на Electron а інформацію з бази даних отримувати через звернення до REST API сервера. В такому випадку ми налаштуємо локальний сервер nodeJS + express + Instant Client і зможемо через нього спробувати працювати з Electron.  
  З мінусів: потребує тестування підключення багатьох користувачів. Це пов’язано з тим, що nodeJS при підключенні наступного користувача змінює connect по якому заходив попередній користувач. Як варіант вирішити за допомогою використання двох підключень (перше для перевірки валідності користувача), а інше вже для роботи з даними.

**Варіант №2.**

* Electron + Web додаток на сервері. Ідея в тому, щоб написати веб додаток порталу, розмістити його на сервері, і відкривати його при завантаженні Electron. Тобто весь функціонал буде на сервері, а Electron тільки як інструмент для завантаження і відкриття exe файлів. Із плюсів – це можливість використовувати різні технології на сервері (JS, php) і проводити дрібні оновлення безпосередньо без обновлення exe файлу.  
  Із мінусів – це проблема з генерацією url (токенів) в Electron для переходу на інші додатки, оскільки в Electron не буде в такому випадку підключення до бази даних і він не зможе отримати значення secret key з бази даних.

**Варіант №3.**

* Electron + Web додаток + REST API service. Це комбінація двох попередніх варіантів для того, щоб дати можливість Electron працювати з базою даних (через REST API) і при цьому винести весь основний функціонал порталу на сервер як веб додаток, а Electron буде як посередник (з можливістю запускати exe файли).

Такі різні варіанти архітектури додатку були розглянуті з урахуванням масштабування порталу в майбутньому і зручності підтримки роботи цього додатку.