Voyis Imaging Inc.
120 Randall Drive, Unit E
Waterloo ON, Canada, N2V 1C6

+1 519.489.0005

Project Brief: Electron-based Image Editor tool

Objective:
Design and implement a lightweight windows application written in the Electron framework, capable of uploading, syncing and displaying image data from the server. This project is designed to evaluate your ability to design a full-stack framework under Electron, set up a robust, maintainable data workflow, interface with a mock server environment (Docker, PostgreSQL, API), and integrate third-party libraries(WASM, N-API).

Timeline:

- **Project Duration:** 2 weeks from the start date.

Evaluation Criteria:

1. **Functionality:** Completeness of the implemented features as per the requirements.
2. **UI/UX Design:** Intuitiveness, responsiveness, and overall aesthetic quality of the interface.
3. **Code Quality:** Readability, maintainability, and adherence to best practices.
4. **Performance:** Smooth UI interaction and fast response.
5. **Bonus Points:**
    - Implementation of additional features (WASM or native-addon-api.).
    - Innovative UI/UX improvements that enhance user.

Notes:

- Candidates are encouraged to use publicly available datasets for testing and demonstration.
- A clean, professional UI and seamless interaction will weigh heavily in the evaluation.

Voyis Imaging Inc.
120 Randall Drive, Unit E
Waterloo ON, Canada, N2V 1C6

+1 519.489.0005

- Questions or requests for clarification can be submitted via email during the project period.

Functional Requirements:

1. Server Structure:

PostgreSQL:

- This database needs to run on a separate Docker container to mock a server-side database.

API Server:

- This is the entry to all DB related request, including all CRUDL operation.

Mounted Storage:

This is a mounted storage to mock some existing data on the server-side.

---

2. Application Structure:

- **File Upload:**
  - Users must be able to upload the following file types into the server-side:
    - **Image Data:** JPG, TIF, PNG
    - **Image quality:** 4k

      Folder Config: A JSON file store list of folders and files type to perform batch insert.
  - The application should display feedback on uploaded files, including:
    - Total file count, Total file size, number of corrupted images.

- **Gallery Viewer**
  - Provide a **gallery viewer** for images that have been uploaded through the application. The gallery will show only thumbnails of images.
  - Basic controls for user interaction:

Voyis Imaging Inc.
120 Randall Drive, Unit E
Waterloo ON, Canada, N2V 1C6

+1 519.489.0005

File filter by type, customized selection, batch export/download to folder.

- **Single-Image Viewer**

  If user double clicks on any file, it will jump to this single viewer

  Images: **Pan, Zoom, area selection on the image:** For area selection, user can choose to generate new images from the selected area.

---

3. Control Panel

- Provide a **control panel** to manage the synchronization between application and server-side DB.
- Include the following functionalities:
  - **When the local application is outdated:** Update local application from the remove server. (i.e new file added to the server)
  - **Handle conflict between server and local application synchronization:** You are free to choose a simple, common strategy (e.g., "Last Write Wins," "Server Always Wins," or "Local Always Wins"). In your README.md (or design document), you must do the following:
    - Clearly state which automated strategy you chose and why.
    - Briefly discuss the potential flaws or data-loss risks of your chosen strategy.

4. User Interface

- Design an intuitive and responsive UI following modern design principles.
- **Example Layout:**
  - **Left Panel:** File upload section with metadata display.
  - **Center Panel:** Tabs for switching between Single-File Viewer or thumbnail gallery view.
  - **Bottom Panel:** Log of user actions and system feedback.

Voyis Imaging Inc.
120 Randall Drive, Unit E
Waterloo ON, Canada, N2V 1C6

+1 519.489.0005

*Note: You can come up with your own layout

- Ensure the UI adapts to different screen sizes and maintains usability on desktop and tablet devices.

---

5. Additional Features (Optional Enhancements)

- **Exif data extraction/insertion:**
    - Allow users to review/edit current image meta data and sync with the server's Database.
- **WASM / Native-addon-api:**
    - This is a free-style feature. You can choose any interesting wasm package or node native addon api to the application. Note: this **must** be a useful functionality for the images.

Non-Functional Requirements:
Technology Stack:

- **Front-End Framework:** Use a modern framework like **React.js**, **Vue.js** or similar.
- **Electron Framework:** Modern Development Toolchain/Build system are recommended like **electron-vite, electron-forge**

Deliverables:

- A fully functional application that can be reliably started from source code. * The README.md file must contain clear, step-by-step instructions on how to (1) launch the server-side environment (e.g., docker-compose up) and (2) run the Electron application in a development environment (e.g., npm run dev).

  A brief demo video showcasing the features.
- Source code hosted on GitHub, including a detailed README file with setup instructions.
- Bonus: Basic design document that outlines the architecture and basic workflows

Performance Expectations

Voyis Imaging Inc.
120 Randall Drive, Unit E
Waterloo ON, Canada, N2V 1C6

+1 519.489.0005

UI Smoothness: Ensure the display and interaction for the image and viewing gallery are smooth under a reasonable test data load.

Scalability Design (Open-ended): We do not require your application to actually handle large-scale images for this submission. *However, you might detail in README.md (or a bonus design document) the following:

1. What design decisions did you make to prepare for future large-scale data (e.g., 100k+ images)?

2. What specific optimization techniques (e.g., in the UI, API, or Database) would you implement to handle large-scale syncing and rendering?

Notes: Candidates are free to use any available tools out there including AI/LLM tools. If your project meets the requirements, we will invite you a face-to-face interview in our office to talk about the project and ask you additional technical questions.