Python Developer Assessment - DCC Integration

This assignment tests your ability to connect Python with a DCC application (Maya or Blender). You'll build a plugin and a server to manage object transforms and a simple inventory.

Part 1: DCC Plugin (Maya or Blender - *choose one,* (Optional, but recommended))
Solutions with this part included will be prioritized.

1. **Interface:** Create a plugin with:
    a. Object selection (how you choose objects in Maya/Blender).
    b. Transform controls (for position, rotation, scale).
    c. Endpoint dropdown (selects which server function to use).
    d. Submit button (sends data to the server).
2. **Functionality:**
    a. Transform controls update when you change object transforms in the DCC.
    b. The plugin sends the selected object's transform data to the server when you click "Submit."

Part 2: Local Server (Flask or FastAPI)

1. **Endpoints:** Create a server with these functions:
    a. `/transform`: Takes all transforms (position, rotation, scale).
    b. `/translation`: Takes only position.
    c. `/rotation`: Takes only rotation.
    d. `/scale`: Takes only scale.
    e. `/file-path`: Returns the DCC file's path. `/file-path?projectpath=true` returns the project folder path.
    f. `/add-item`: Adds an item to a database (name, quantity).
    g. `/remove-item`: Removes an item from the database (by name).
    h. `/update-quantity`: Updates an item's quantity (name, new quantity).
2. **Behavior:**
    a. 10-second delay for all responses.
    b. Logs received requests to the terminal.
    c. Use correct status codes (200, 400, 404).

Part 3: Database (SQLite)

1. **Inventory:** Use a SQLite database to store items and quantities.
2. **Server Interaction:** The server updates the database based on the `/add-item`, `/remove-item`, and `/update-quantity` requests.

Part 4: PyQt/PySide UI

1. **Inventory Display:** Create a UI that shows the inventory from the database.
2. **Purchase/Return:** Add buttons to buy/return items (updates the database and the DCC plugin's display).
3. **Responsiveness:** The UI should not freeze while waiting for server responses.

Requirements:

- Python proficiency.
- DCC Python API knowledge (Maya or Blender).
- REST API experience (Flask or FastAPI).
- SQLite database skills.
- Git for version control.

Bonus:

- Single-binary packaging (PyInstaller).
- Advanced UI features.
- Testing.

Submission

Submission form:

https://forms.office.com/Pages/ResponsePage.aspx?id=Mk3i-9P_ukSMeKBIfw3_deWZqt_DAWdPoUAGTvjAuxpUOFNZNU5ZQ0taSkxENDIyNDBOWjlGNVZDUC4u

Contact

Siddesh Sharma – siddesh@vigaet.com

Frequently Asked Questions about the Assignment.

- **Are Viga internships paid?** Yes, all Viga internships are paid.
- **What is the expected timeframe for completing the assignment?** We encourage you to complete the assignment within one week of receiving it.
- **What if I need more time?** While we aim for a one-week turnaround, the submission window is open indefinitely. You are welcome to take a few extra days if needed. There's no need to request an extension.