01-30 Meeting: Custom Integration for Acuity and Stripe

Key Points

- Acuity hosts events, using Acuity (via Squarespace) for booking and intake forms, and Stripe for payments. The primary challenges are that the two systems do not integrate well and Acuity's built-in payment system does not handle taxes or "pay-what-you-want" pricing models, which are crucial for their events.
- The current workflow requires attendees to register in Acuity, receive a confirmation email, and then click a separate link to pay via Stripe. This process is disjointed and leads to discrepancies where some people book without paying, and others pay without booking.
- Brian Y. spends a significant amount of time (sometimes half a day per event) manually cross-referencing data from Acuity and Stripe in spreadsheets to confirm payments, track attendance, and manage logistics like accommodations and dietary restrictions. This manual process is becoming unsustainable as the number of events increases.
- Naveed N. has been experimenting with a workaround using Zapier and Google Sheets to create a "state-based system" that tracks whether a booking has been paid for or if a payment has a corresponding booking. The goal is to automate reminders for incomplete registrations.
- The team discussed the need for a more robust, centralized system—potentially a custom dashboard with a backend database (like PostgreSQL)—to automate the entire process. This system would bridge Acuity and Stripe, centralize attendee data, and automate communications like registration reminders and day-of-event text messages with logistical details.
- Another significant pain point is the lack of a centralized portal for co-creators/co-hosts to view event registration data (attendee lists, payment status, etc.), which currently creates an efficiency drain on Brian Y. who has to manually provide this information.
- Brandon A. proposed developing a custom solution using the Acuity and Stripe APIs. This would be a "code first" approach, creating a more permanent and integrated system than the Zapier workaround.

Decisions Made

- The team will move forward with exploring a custom-built solution to integrate Acuity and Stripe, automate workflows, and create a centralized dashboard.
- Brandon A. will research potential solutions, analyze the current system's data, and present 2-3 different options for the team to consider.
- Brian Y. will grant Brandon A. access to their Acuity, Stripe, and Zapier accounts to facilitate research and development.

Action Items

Tasks

Task	Responsible Party	Deadline	Notes
Grant Brandon A. access to Stripe, Squarespace (Acuity), and Zapier accounts.	Brian Y.	Later today	Use the specified Gmail address.
Research and analyze the current systems (Acuity, Stripe, Zapier).	Brandon A.	N/A	To understand the data and current workflow.
Develop and present 2-3 potential solution options to the team.	Brandon A.	N/A	Options should address the integration, automation, and dashboard needs.
Develop a Minimum Viable Product (MVP) or prototype for testing.	Brandon A.	By late February	The goal is to test the new system alongside the current one before the busy season in March.
Provide a Standard Operating Procedure (SOP) for the new system.	Brandon A.	N/A	To be delivered with the final product.

Deadlines

- **Late February**: A prototype (MVP) of the new system should be ready for testing.
- **March**: The ideal timeframe to have a new system live and tested, as the number of events starts to pick up significantly.

Follow-Up Actions

- Brian Y. will provide Brandon A. with access to all relevant platforms (Stripe, Squarespace/Acuity, Zapier) later today.
- Brandon A. will research options and then circle back with Brian Y. to present different approaches and discuss project scope and cost.
- The team will test the new prototype system in late February to ensure it is functioning correctly before the main event season begins.