Automations: make your systems work together

Email & marketing · https://schoolyland.com/docs/automations-connect-your-systems/

Your platform ships as one connected machine rather than a pile of plugins, so things that happen in one system can trigger actions in another automatically. Two engines handle this: **FluentCRM automations** (everything around contacts and email) and **Uncanny Automator** (cross-system recipes between your store, courses, community and more).

The pattern: trigger → actions

Every automation is the same sentence: *when X happens, do Y (and Z)*.

- *When someone buys the course product* → enroll them in the course, tag them "customer", start the onboarding email sequence.
- *When a student completes the course* → send the certificate email, invite them to the community, start the "next offer" sequence a week later.
- *When a subscription payment fails* (with WooCommerce Subscriptions, sold separately) → tag "payment-issue", email a card-update link.
- *When someone fills the contact form* → create the contact, tag by interest, notify you.

Where to build what

- **FluentCRM → Automations**: when the trigger or the action involves contacts, tags or emails. This covers most of what a course business needs: purchase triggers, form triggers, tag-based sequences.
- **Automator → Add New Recipe**: when you're connecting systems beyond email: LearnDash events, WooCommerce events, community actions, user account changes. Pick a trigger from one system, actions from any other.

Rule of thumb: start in FluentCRM; reach for Automator when FluentCRM doesn't see the event you need.

Build your first one (10 minutes)

The purchase automation is the one every business needs on day one:

1. **FluentCRM → Automations → New**, trigger: WooCommerce "order completed" for your product.
2. Action: apply a tag named after the product, such as *customer-course-x*.
3. Action: send a welcome email: where to log in, where to start, where to get help.

(Course enrollment itself is handled by the product-course link. See Sell your first product.)

Or just describe it to your AI

Automations are a good fit for the AI layer. Describe the outcome, for example "*When someone finishes the pricing course, wait 3 days and send them an invitation to book a call*", and your assistant builds the automation for you and shows you what it created. See Connect your AI.

One caution

An automation runs every time its trigger fires, so a mistake in it repeats too. Test each new automation with a real test contact before trusting it, and give everything a descriptive name ("Purchase → onboarding (Course X)") so you can still tell later what it does.

When an automation doesn't fire

Four checks find nearly every case:

1. **Is it active?** A paused automation looks identical to a broken one.
2. **Did the trigger happen as a new event?** Triggers fire on the *event*, not the state. "Tag applied" runs only when the tag is newly added; contacts who already had the tag before the automation existed won't enter it.
3. **Is the contact subscribed?** Unsubscribed or pending contacts are skipped by email steps.
4. **Did it fire but the email never arrived?** Check the automation's activity and FluentCRM's email log. If the log says sent but inboxes are empty, that's deliverability: our side. Open a ticket with

the automation name and a recipient address.