## Email Automation Schedule Setup (phpMyAdmin/WordPress)

This guide outlines how a solopreneur can automate the delivery of the **Bloating Myth‑Busting Email Course** using WordPress with phpMyAdmin for database management. It leverages WordPress’s built‑in cron system and an SMTP plugin for reliable email delivery.

### 1. Configure Email Deliverability

WordPress uses PHP’s mail() function by default, which is basic and not authenticated. Emails sent via this function are more likely to be marked as spam and may not reach recipients[[1]](https://wpmailsmtp.com/how-to-send-automated-email-in-wordpress/#:~:text=This%20is%20because%20WordPress%20uses,supported%20by%20all%20hosting%20providers). To improve deliverability, install and configure the **WP Mail SMTP** plugin. This plugin routes WordPress emails through a third‑party SMTP service (e.g., SendLayer, Gmail, or SMTP.com) to ensure your messages are authenticated and reach the inbox[[2]](https://wpmailsmtp.com/how-to-send-automated-email-in-wordpress/#:~:text=This%20is%20because%20WordPress%20uses,supported%20by%20all%20hosting%20providers).

### 2. Create Database Tables

In phpMyAdmin, create two tables:

* **users** – stores subscriber data (user\_id, name, email, signup\_date).
* **course\_schedule** – stores course send schedule (id, user\_id, course\_id, day\_number, scheduled\_send\_datetime, sent\_status). The course\_id field allows you to reuse the same table for multiple nurture sequences.

Populate course\_schedule when a user signs up: insert five rows, one for each email day, with scheduled send dates computed relative to the signup time (e.g., signup + 1 day, signup + 2 days, etc.).

### 3. Set Up WordPress Cron Events

WordPress has a built‑in cron system for scheduled tasks. A tutorial on automatic email sending explains that you need to:

1. **Schedule a cron event** that runs at regular intervals (e.g., every hour)[[3]](https://xnau.com/automatically-sending-an-email-using-the-wordpress-cron/#:~:text=1,was%20sent%20to%20that%20person).
2. **Query the database** during each run to find schedule rows where sent\_status = 0 and scheduled\_send\_datetime <= now()[[4]](https://xnau.com/automatically-sending-an-email-using-the-wordpress-cron/#:~:text=When%20the%20cron%20is%20triggered%2C,qualify%20for%20the%20reminder%20email).
3. **Check for duplicates:** Ensure each email is sent only once by updating sent\_status immediately after sending[[3]](https://xnau.com/automatically-sending-an-email-using-the-wordpress-cron/#:~:text=1,was%20sent%20to%20that%20person).
4. **Send the email** using wp\_mail() (or a mailing library) with the appropriate template and personalized data[[5]](https://xnau.com/automatically-sending-an-email-using-the-wordpress-cron/#:~:text=Sending%20the%20Email).
5. **Log the send** by setting sent\_status = 1 and recording the sent timestamp, so future cron runs don’t resend the same email[[6]](https://xnau.com/automatically-sending-an-email-using-the-wordpress-cron/#:~:text=Once%20someone%20has%20been%20reminded%2C,time%20we%20sent%20the%20email).

Here is a simplified PHP snippet to schedule and process the cron event:

// Schedule the cron event when plugin/theme is activated  
function fnm\_schedule\_email\_cron() {  
 if (!wp\_next\_scheduled('fnm\_send\_course\_emails')) {  
 wp\_schedule\_event(time(), 'hourly', 'fnm\_send\_course\_emails');  
 }  
}  
register\_activation\_hook(\_\_FILE\_\_, 'fnm\_schedule\_email\_cron');  
  
// Cron callback: send pending course emails  
function fnm\_send\_course\_emails() {  
 global $wpdb;  
 $table\_name = $wpdb->prefix . 'course\_schedule';  
 $rows = $wpdb->get\_results( $wpdb->prepare(  
 "SELECT \* FROM $table\_name WHERE sent\_status = %d AND scheduled\_send\_datetime <= NOW()",  
 0  
 ) );  
 foreach ($rows as $row) {  
 $user = get\_userdata($row->user\_id);  
 $email\_content = fnm\_get\_email\_template($row->day\_number, $user);  
 wp\_mail($user->user\_email, $email\_content['subject'], $email\_content['body']);  
 // mark as sent  
 $wpdb->update( $table\_name, [ 'sent\_status' => 1, 'sent\_timestamp' => current\_time('mysql') ], [ 'id' => $row->id ] );  
 }  
}  
add\_action('fnm\_send\_course\_emails', 'fnm\_send\_course\_emails');

### 4. Manage Templates

Define a function (e.g., fnm\_get\_email\_template($day, $user)) that returns the subject and body for each day. Pull content from the JSON file generated for the course to keep templates separate from logic. Use placeholders like [FirstName] to personalize the salutation.

### 5. Scale for Multiple Courses

The presence of a course\_id column in course\_schedule allows you to schedule different courses simultaneously. When a new course is added, insert rows with the appropriate course\_id and email count. Your cron callback should select rows based on both course\_id and unsent status.

### 6. Testing & Monitoring

* Use a plugin such as **WP Crontrol** to view and manually trigger scheduled cron events during testing[[7]](https://xnau.com/automatically-sending-an-email-using-the-wordpress-cron/#:~:text=It%E2%80%99s%20a%20good%20idea%20to,to%20trigger%20on%20it%E2%80%99s%20own).
* Install an email logging plugin to record outgoing messages and verify that emails are sent as expected[[7]](https://xnau.com/automatically-sending-an-email-using-the-wordpress-cron/#:~:text=It%E2%80%99s%20a%20good%20idea%20to,to%20trigger%20on%20it%E2%80%99s%20own).
* Always test with a small group before rolling out to all subscribers. Add error logging in your cron callback so you can diagnose issues without sending unintended emails[[8]](https://xnau.com/automatically-sending-an-email-using-the-wordpress-cron/#:~:text=Testing%20the%20Plugin).

By combining phpMyAdmin for scheduling, WordPress cron for automation, and a reliable SMTP service, a solopreneur can deliver a professional, scalable nurture sequence without manual intervention.

[[1]](https://wpmailsmtp.com/how-to-send-automated-email-in-wordpress/#:~:text=This%20is%20because%20WordPress%20uses,supported%20by%20all%20hosting%20providers) [[2]](https://wpmailsmtp.com/how-to-send-automated-email-in-wordpress/#:~:text=This%20is%20because%20WordPress%20uses,supported%20by%20all%20hosting%20providers) How to Send Automated Email in WordPress

<https://wpmailsmtp.com/how-to-send-automated-email-in-wordpress/>

[[3]](https://xnau.com/automatically-sending-an-email-using-the-wordpress-cron/#:~:text=1,was%20sent%20to%20that%20person) [[4]](https://xnau.com/automatically-sending-an-email-using-the-wordpress-cron/#:~:text=When%20the%20cron%20is%20triggered%2C,qualify%20for%20the%20reminder%20email) [[5]](https://xnau.com/automatically-sending-an-email-using-the-wordpress-cron/#:~:text=Sending%20the%20Email) [[6]](https://xnau.com/automatically-sending-an-email-using-the-wordpress-cron/#:~:text=Once%20someone%20has%20been%20reminded%2C,time%20we%20sent%20the%20email) [[7]](https://xnau.com/automatically-sending-an-email-using-the-wordpress-cron/#:~:text=It%E2%80%99s%20a%20good%20idea%20to,to%20trigger%20on%20it%E2%80%99s%20own) [[8]](https://xnau.com/automatically-sending-an-email-using-the-wordpress-cron/#:~:text=Testing%20the%20Plugin) Automatically Sending an Email using the WordPress Cron – xnau webdesign

<https://xnau.com/automatically-sending-an-email-using-the-wordpress-cron/>