HireBoard

APPLICANT TRACKING SYSTEM

Installation Guide

Install and configure HireBoard on your server.

Version 1.0.0

June 2026

1. Overview

HireBoard is a self-hosted PHP application built on the **Laravel 13** framework. It runs on any standard PHP hosting that meets the requirements below — including most shared hosting, VPS, and dedicated servers. Installation has three parts:

1. Upload the files and serve the application from its `public/` folder.
2. Make a few folders writable.
3. Open the application in your browser and complete the web installer.

> **Ready to run.** The package ships with all PHP dependencies included, so you do *not* need Composer or any build tools to install it — you only need a server that meets the version requirements below.

2. Server Requirements

> **Read this first.** HireBoard 1.0 requires **PHP 8.3.0 or higher** and runs on **Laravel 13**. This is a hard requirement — the bundled dependencies will not run on PHP 8.2 or lower, and the application will stop on start-up with a Composer platform error if it is launched on an unsupported PHP version. Confirm your PHP version *before* uploading.

PHP & Laravel

PHP 8.3.0 or higher, on the **Laravel 13** framework (bundled — nothing to install). Most control panels let you select the PHP version per website; make sure yours is set to 8.3 or newer. If your host is on an older version, read Running on PHP below 8.3 before you continue.

Required PHP extensions

The web installer checks all of these for you on its first screen:

Extension	Purpose
PDO & PDO MySQL	Database connectivity
Mbstring	Multibyte string handling
OpenSSL	Encryption
Tokenizer	Framework internals
JSON	Data encoding
cURL	Outbound requests

Extension	Purpose
Fileinfo	File type detection for uploads
BCMath	Precise calculations
Zip	Creating and restoring backups
GD	Image handling
XML	Document generation
Ctype	Character validation

Database

MySQL 5.7+ or **MariaDB 10.3+**. You will need a database, a database user, and that user's password. The installer can create the database for you if your database user has permission to do so.

Web server

Apache (with `mod_rewrite`) or Nginx. The application **must** be served from its `public/` directory — see Document Root & the public/ Folder.

Running on PHP below 8.3

If your server's PHP is older than 8.3 and you control it, the cleanest fix is simply to raise the PHP version in your control panel. If you cannot raise the account-wide version, you have two further options.

Option 1 — Set a higher PHP version for just this site/folder. Many LiteSpeed / CloudLinux shared hosts let you run a different PHP version for a single domain, subdomain, or folder by adding a handler line to that folder's `.htaccess`. For example, to force PHP 8.3 for the HireBoard folder only:

```
<FilesMatch "\.(php|phtml)$">
    SetHandler application/x-httpd-alt-php83___lsphp
</FilesMatch>
```

> **The handler string is host-specific.** The exact value (`application/x-httpd-alt-php83___lsphp` on CloudLinux, `application/x-lsphp83` on some LiteSpeed builds) depends on your host — check your provider's "change PHP version per directory" documentation. Note also that this changes only the *web* PHP version; the command-line `php` over SSH may still be your account default, so run artisan with the matching binary path when needed.

Option 2 — Downgrade HireBoard to PHP 8.2 / Laravel 12. If you cannot obtain PHP 8.3 at all, HireBoard can be re-resolved to run on PHP 8.2 by lowering the framework to Laravel 12. In broad terms this means setting `"php": "^8.2"` and `laravel/framework: "^12"` in `composer.json`, pinning the

build platform to 8.2, running `composer update` on a machine with PHP 8.2, and re-testing the whole application. It normally works, but it is a non-trivial change you perform at your own discretion.

> **The PHP 8.2 / Laravel 12 downgrade is unsupported.** If you choose to downgrade, you do so **entirely at your own risk**. The downgrade is **not covered by item support** and is not included with your purchase — any issues arising from it are your responsibility. If you would like it done and verified for you, we offer it as a **paid customization**: contact contact@mes-dev.com.

---

3. Installation Types

HireBoard can be installed in two scenarios. The application and the web installer are identical in both; only the environment around them differs. **In both cases the version pre-requirements above are mandatory — confirm PHP 8.3+ before you begin.**

Local installation (your own machine)

Use this to evaluate, test, or develop against HireBoard on your computer before going live.

Pre-requirements (local):

- A local PHP **8.3+** environment — for example Laragon, XAMPP configured with PHP 8.3, Herd, or any setup where `php -v` reports 8.3 or higher.
- A local **MySQL or MariaDB** server (e.g. the one bundled with Laragon/XAMPP).
- The required PHP extensions listed in Section 2 (most local stacks enable them by default).

1. Unzip the package into your local web directory (for example `C:\laragon\www\hireboard` or `~/Sites/hireboard`).
2. Confirm your local PHP is 8.3+ with `php -v`.
3. Create an empty local database and a database user.
4. From the project root, start Laravel's built-in server: `php artisan serve`. This serves the `public/` folder for you at `http://127.0.0.1:8000` with clean URLs — no `.htaccess` or document-root setup needed locally.
5. Open `http://127.0.0.1:8000/install` and complete the web installer (Section 7).

> **Why `php artisan serve` is easiest locally.** It points at `public/` automatically, so the `public/` -folder configuration in Section 6 only matters for a live server (or if you prefer to run a local Apache/Nginx vhost instead).

Live installation (online server)

Use this for production — shared hosting, a VPS, or a dedicated server reachable on your domain.

Pre-requirements (live):

- Hosting with **PHP 8.3+** selectable for your domain or subdomain, and the extensions from Section 2.
- **MySQL 5.7+ / MariaDB 10.3+**, with a database and user you can create from your panel.
- The ability to point your domain's **document root at the `public/` folder** — or, if your host won't allow that, the ability to use the included `.htaccess` rewrite (Section 6, Method B).
- An **SSL certificate** so the site is served over `https://` (most hosts offer free Let's Encrypt).

1. Upload and unzip the package on your server (see Before You Begin).
2. Serve the app from `public/` using Method A or Method B in Section 6.
3. Set the writable folder permissions in Section 5.
4. Open your domain in a browser and complete the web installer (Section 7).

---

4. Before You Begin

1. Download the HireBoard package and unzip it.
2. Upload its contents to your server — typically into your hosting account's web directory (for example `public_html`) or a folder above it.
3. Serve the application from its `public/` folder (see Document Root & the public/ Folder).

> **Security:** only the `public/` directory should ever be reachable from the web. Keeping the application files above the web root protects your configuration (`.env`), dependencies (`vendor/`), and data (`storage/`).

---

5. Folder Permissions

Before running the installer, make sure the following directories are **writable** by the web server. The installer verifies these on its first screen and shows you any that need attention.

- `storage/` (and everything inside it)
- `storage/app/`
- `storage/app/public/`
- `storage/framework/` (cache, sessions, views)
- `storage/logs/`
- `bootstrap/cache/`

On most shared hosting a permission of `755` is sufficient. If your host runs PHP under a different user, you may need `775`. Your hosting control panel's file manager can set these, or via SSH:

```
chmod -R 775 storage bootstrap/cache
```

6. Document Root & the public/ Folder

HireBoard — like every Laravel application — must be served from its `public/` folder, never from the project root. The project root holds `.env`, `vendor/`, and `storage/`, which must never be reachable from the web. There are two supported ways to achieve this; choose the one your host allows.

Method A — Point the document root at `public/` (recommended)

This is the cleanest and most secure layout: the web server's document root resolves directly to `.../hireboard/public`, and the rest of the application sits outside the web root.

> **This does not put `/public` in your URLs.** A common misconception: pointing the document root at `public/` *removes* the need for `/public` — your site loads at `https://your-domain.com/` with clean URLs. (You only ever see `/public` in the address bar when the document root is wrong and a redirect or rewrite is exposing it.)

Apache — point the virtual host's `DocumentRoot` at `public/`. HireBoard ships the necessary `public/.htaccess`, so with `mod_rewrite` enabled and `AllowOverride All` set, clean URLs work out of the box.

```
<VirtualHost *:80>
    ServerName your-domain.com
    DocumentRoot /var/www/hireboard/public

    <Directory /var/www/hireboard/public>
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Nginx:

```
server {
    listen 80;
    server_name your-domain.com;
    root /var/www/hireboard/public;
    index index.php;

    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }

    location ~ \.php$ {
        fastcgi_pass unix:/var/run/php/php8.3-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

Shared hosting (cPanel, hPanel, and similar) — most panels let you set the document root per domain or subdomain. When adding or editing the domain/subdomain, set its document root (or folder) to the application's `public` directory, e.g. `.../hireboard/public`. This is the cleaner option whenever your host offers it.

Method B — Rewrite into `public/` (when you can't change the document root)

Some shared hosts fix the document root to a folder you cannot repoint (such as `public_html`). In that case, upload the **entire application** into that folder and add a small `.htaccess` at the **project root** that forwards every request into `public/`.

File: `<project root>/.htaccess` — at the top level, next to `app/` and `public/` (this is *not* the `.htaccess` inside `public/`, which you leave untouched).

```
<IfModule mod_rewrite.c>
    RewriteEngine On

    # Don't re-enter public/ (prevents a redirect loop on LiteSpeed)
    RewriteCond %{REQUEST_URI} !^/public/ [NC]

    # Internal rewrite into public/ — no R flag, so the URL stays clean
    RewriteRule ^(.*)$ public/$1 [L]
</IfModule>
```

This forwards requests into `public/` internally, so visitors never see `/public` and your application files are protected by the bundled rules. Method A remains more secure when available, because it keeps the application files outside the web root entirely.

Keeping URLs clean

HireBoard generates every link from your **application URL**, so as long as that value is correct your links stay clean under either method. The web installer writes it for you on the Company step.

> **If you ever see `/public` in your links** (most likely with Method B), make sure `APP_URL` in your `.env` matches your real domain exactly — for example `APP_URL=https://your-domain.com` with **no** `/public` and **no** trailing slash — then clear the cache (your panel's cache tool, or `php artisan optimize:clear` via SSH).

---

7. The Web Installer (Recommended)

With the files uploaded and the application served from `public/`, open your domain (or `http://127.0.0.1:8000` locally) in a browser. HireBoard automatically redirects you to the installer at `/install` and guides you through six steps.

Step 1 — Requirements

The installer checks your PHP version, the required extensions, and folder permissions. Everything must show a green check before you can continue. Fix anything marked red (raise your PHP version, enable the extension with your host, or correct a folder permission) and refresh.

Step 2 — Database

Enter your database host, port, name, username, and password, then click **Test Connection**. HireBoard verifies the credentials and tells you whether the database already exists, is empty, or already contains tables. If it does not exist yet, the installer will create it for you during setup (provided your user has permission).

> **Installing into an existing database?** If the database already contains tables, the installer asks you to confirm before continuing, so existing data is never modified by accident. Using a fresh, empty database is always safest.

Step 3 — Setup

Click to run setup. HireBoard creates the database if needed, builds all the tables, installs every module, and generates a unique application security key for your installation. This step is automatic — just wait for it to finish.

Step 4 — Admin Account

Create your administrator account by entering a name, email, and password. This is the account you will use to sign in and manage the system.

> **No default passwords.** Your admin account is created here with a password *you* choose — the package ships with no default credentials.

Step 5 — Company

Enter your company name, choose your timezone, and confirm your **application URL** (your live domain). Enter it exactly — for example `https://your-domain.com`, with no `/public` and no trailing slash. This value is saved as `APP_URL` and is what keeps every generated link clean.

Step 6 — Finish

The installer locks itself (so it cannot be run again), clears caches, and shows a success screen. Click through to the sign-in page and log in with the admin account you created.

> **That's it — HireBoard is installed.** The installer also creates the link that serves your uploaded files, so résumés, logos, and offer PDFs work immediately.

---

8. After Installation

- **Use HTTPS.** Install an SSL certificate (most hosts offer free Let's Encrypt) and serve the site over `https://`.
- **Debug stays off.** The shipped configuration runs in production mode with debugging disabled — leave it that way on a live site.
- **Configure email** so notifications can be sent (next section).
- **Turn on automatic backups** under **Backups → Schedule & retention**.

---

9. Email (SMTP) Setup

To let HireBoard send email, sign in as the administrator and go to **Settings → Mail**. Enter your SMTP server, port, username, password, and encryption, then send a test email to confirm it works. These settings apply to the whole application, so all notifications use your account.

10. Automatic Backups

HireBoard's scheduled backups are controlled entirely from the admin panel — there is **no cron job to configure**. Under **Backups → Schedule & retention**, set automatic backups to Daily or Weekly and choose how many to keep. Backups run automatically while the site is in use.

> **Advanced (optional).** If you prefer precise timing on a server you control, a console command `php artisan backup:run --keep=N` is also available to place on a system cron. This is entirely optional — the in-app schedule is all most installations need.

11. Updating HireBoard

1. **Back up first** — create and download a backup from the Backups area.
2. Replace the application files with the new version, keeping your `.env` file and your `storage/` contents.
3. Clear caches (your host's control panel or, via SSH, the standard Laravel cache-clear commands).

Always read the release notes (`CHANGELOG.md`) included with each update for any version-specific steps.

12. Troubleshooting

A Composer / PHP version error on start-up

A message such as *"Composer detected issues in your platform: Your Composer dependencies require a PHP version >= 8.3.0"* means the site is running on an **older PHP version**. Raise your site's PHP to 8.3+ in your control panel, or apply one of the options in Running on PHP below 8.3. Remember the web PHP version and the SSH command-line `php` can differ.

"/public" appears in my page links

This happens when the app is served via the Method B rewrite but `APP_URL` is not set to your clean domain. Set `APP_URL` in `.env` to your real domain with no `/public` and no trailing slash, then run `php artisan optimize:clear` (or clear the cache from your panel). See Keeping URLs clean.

A "500 Server Error" or blank page

Almost always a permissions issue. Confirm the folders in Section 5 are writable. Check `storage/logs/laravel.log` for the specific error.

Uploaded files (logos, résumés) don't display

The link that serves uploaded files is missing. The installer creates it automatically; if files were uploaded out of order, recreate it by running `php artisan storage:link`, or ask your host to create a symbolic link from `public/storage` to `storage/app/public`.

"Could not connect to the database"

Re-check the host, port, database name, username, and password. On many shared hosts the database host is `localhost` rather than `127.0.0.1`, and database/user names are prefixed by your account name.

Pages other than the homepage return 404

URL rewriting is not active. On Apache, enable `mod_rewrite` and ensure `AllowOverride All`. On Nginx, confirm the `try_files` directive shown above.

The installer keeps reappearing or won't start

The installer runs until an install-lock file exists. If setup was interrupted, complete all six steps. If you need to re-run it deliberately, remove the file `storage/installed` and reload.

---

13. Manual Installation (Advanced)

If you have command-line access and prefer to install manually:

1. Upload the files and serve the application from `public/` (Section 6).
2. If you removed the bundled dependencies, run `composer install --no-dev --optimize-autoloader` on PHP 8.3+.
3. Copy `.env.example` to `.env` and run `php artisan key:generate`.
4. Set your database credentials and your `APP_URL` in `.env`.
5. Visit `/install` and complete the wizard, or run the migrations and seeders manually if you know the framework.
6. Run `php artisan storage:link`.

> The web installer remains the supported, recommended path even on servers with command-line access — it performs every one of these steps for you and validates the environment as it goes.

14. Support & Contact

We're here to help. Please use the channel that matches your request so we can respond quickly:

- **Product support (for purchased items)** — support@mes-dev.com. Include your CodeCanyon **purchase code** and your HireBoard version so we can assist you faster.
- **General inquiries, customizations, or a new application** — contact@mes-dev.com.

Product support covers installation help, bug fixes, and questions about documented features, in line with the item's support policy on CodeCanyon. Environment changes you make yourself — such as the PHP 8.2 / Laravel 12 downgrade described in Section 2 — fall outside support, but are available as a paid customization.