

AirLink Advisor
Your Global Journey, Our Guidance

*Dream
Plan
Move Forward*

Visa & Immigration Consultancy Platform

Complete Laravel PHP Script

- Visa Services Management
- Coaching & Exam Preparation
- Consultant & Team Management
- Blogs, FAQ, Testimonials
- Lead & Enquiry Management
- Multilingual & SEO Ready
- Powerful Admin Panel
- Fully Responsive Design

Explore Global Opportunities With AirLink Advisor

Laravel MySQL Desktop | Tablet | Mobile

AirLink Advisor

Hostinger Installation & Deployment Guide

Laravel 10 • PHP 8.1/8.2 • MySQL/MariaDB • airlinkadvisor.com

Prepared from the supplied project ZIP and SQL database export. This guide covers upload, database import, environment setup, permissions, cache, admin access, email/social login, SSL, troubleshooting, backup and update workflow.

1. Project Snapshot

Item	Detected in supplied build
Framework	Laravel 10
PHP requirement	^8.1 ^8.2
Production URL	https://airlinkadvisor.com
Database	u631869004_airlinkadvisor
DB user	u631869004_airlinkadvisor
Vendor folder	Included (vendor/autoload.php present)
Root routing	.htaccess routes requests through server.php / public assets
Admin route	/admin/login
User login route	/login
Database dump	u631869004_airlinkadvisor.sql

Important: the supplied .env contains live credentials and third-party keys. Keep it private, never publish it, and rotate secrets when transferring the project to a new owner/server.

2. Hostinger Server Requirements

In hPanel, use a hosting plan that supports PHP 8.1 or 8.2, MySQL/MariaDB, mod_rewrite, file permissions and cron jobs. Recommended PHP extensions include OpenSSL, PDO/MySQL, Mbstring, Tokenizer, XML, Ctype, JSON, BCMath, Fileinfo, cURL, GD/Imagick and ZIP.

Setting	Recommended
PHP	8.2 (8.1 is also supported by composer.json)
memory_limit	256M or higher
upload_max_filesize	64M+ if admins upload large media
post_max_size	64M+
max_execution_time	120-300 seconds for imports/uploads
HTTPS	Enabled with Hostinger SSL

3. Fresh Domain Installation - Step by Step

1. Create or connect the domain

Add **airlinkadvisor.com** to Hostinger and point the domain DNS to the hosting account. Wait until the domain resolves correctly.

2. Open File Manager

Go to the domain document root, normally **public_html**. Back up or remove only the default placeholder page.

3. Upload the project ZIP

Upload the supplied project ZIP and extract its contents directly into the domain root. The root should contain **artisan**, **app**, **bootstrap**, **config**, **public**, **resources**, **routes**, **storage**, **vendor**, **.env** and **.htaccess**.

4. Keep hidden files

Do not lose **.env** or **.htaccess**. Hostinger File Manager can hide dotfiles, so enable hidden-file display if needed.

5. Do not move only /public

This build already includes a root **.htaccess** that serves public assets and routes dynamic requests via **server.php**. Keep the supplied directory structure intact unless you intentionally reconfigure the document root.

4. Create and Import the Database

In Hostinger hPanel → Databases → MySQL Databases, create the database and user. For this supplied build the configured names are shown below. If your Hostinger account uses different names, update **.env** to match.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=u631869004_airlinkadvisor
DB_USERNAME=u631869004_airlinkadvisor
DB_PASSWORD=YOUR_DATABASE_PASSWORD
```

Open phpMyAdmin, select the new database, choose **Import**, upload **u631869004_airlinkadvisor.sql**, then run the import. The SQL export is MariaDB/MySQL compatible and contains the application tables and data.

If phpMyAdmin rejects a large upload, increase the available import limit if your plan allows it, or use Hostinger database import/SSH tools.

5. Configure .env for Production

```
APP_NAME='AirLink Advisor'
APP_ENV=production
APP_DEBUG=false
APP_URL=https://airlinkadvisor.com

CACHE_DRIVER=file
SESSION_DRIVER=file
QUEUE_CONNECTION=sync

# Configure real SMTP before testing forms/password reset
MAIL_MAILER=smtp
MAIL_HOST=your.smtp.host
MAIL_PORT=587
MAIL_USERNAME=your-mailbox
MAIL_PASSWORD=your-mail-password
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS=info@airlinkadvisor.com
MAIL_FROM_NAME='AirLink Advisor'
```

The supplied build already contains an APP_KEY. Keep it for this existing database installation. Generate a new key only when intentionally creating a fully independent installation and after confirming no encrypted application values depend on the current key.

6. Writable Folders and Permissions

Laravel must be able to write cache, sessions, logs and uploaded runtime files. On Hostinger, directories normally work at 755, while Laravel writable directories may require 775 depending on ownership. Files normally remain 644. Never set the entire site to 777.

Path	Purpose
storage/	Logs, sessions, cache, framework data
storage/framework/	Laravel session/cache/view files
storage/logs/	Application logs
bootstrap/cache/	Compiled configuration/services cache
public/uploads/	Site/admin uploaded media when used by this build

7. Clear Laravel Caches

If Hostinger Terminal/SSH is available, run these commands from the project root after editing .env:

```
php artisan optimize:clear
php artisan config:clear
php artisan cache:clear
php artisan route:clear
php artisan view:clear
```

You do not normally need **composer install** because the supplied ZIP already contains **vendor/autoload.php**. Run Composer only if vendor is missing or you intentionally change dependencies.

8. SSL and Domain Finalization

Enable Hostinger SSL, force HTTPS, then confirm APP_URL is exactly <https://airlinkadvisor.com>. Test the home page, assets, form submissions, admin login, images and internal links. If assets load over HTTP, clear Laravel/browser/CDN caches and recheck the application URL.

9. Admin and User Access

Area	URL / note
Admin login	https://airlinkadvisor.com/admin/login
Admin dashboard	https://airlinkadvisor.com/admin/dashboard after authentication
User login	https://airlinkadvisor.com/login
Existing admin account	SQL contains admin@gmail.com. Password is stored as a secure hash; this guide does not invent a plaintext password

10. Email, Contact Forms and Password Reset

The supplied .env still uses a local Mailpit configuration. Before production use, replace it with a real SMTP provider. Then test contact/enquiry notifications, registration emails (if enabled), password reset and any admin notification mail.

Recommended test: submit the website enquiry form with an email address you control, verify it appears in Admin → Leads/Enquiries (or the project equivalent), and confirm any configured notification email is received.

11. Google / Social Login

The project includes Laravel Socialite and Google login routes. The supplied configuration contains a Google redirect URI for `airlinkadvisor.com`. In Google Cloud Console, ensure the authorized redirect URI exactly matches the application callback path and rotate client secrets if the package is transferred to another owner.

```
GOOGLE_REDIRECT_URI=https://airlinkadvisor.com/user/callback/google
```

12. Cron / Scheduler

Laravel applications often use scheduled jobs, but the exact need depends on enabled modules. If this project uses Laravel scheduler tasks, configure a Hostinger cron job once per minute:

```
* * * * * php /home/USERNAME/domains/airlinkadvisor.com/public_html/artisan schedule:run >> /dev/null 2>&1
```

Replace USERNAME and path with the actual Hostinger account path. If no scheduled commands are registered, this cron is harmless but optional.

13. Production Security Checklist

- APP_DEBUG=false and APP_ENV=production.
- Keep .env inaccessible from the browser and out of public downloads/backups.
- Rotate database, SMTP, Google/social and payment gateway secrets when ownership changes.
- Use HTTPS everywhere and secure cookies/sessions.
- Use strong admin passwords and remove unused admin accounts.
- Back up files and database before updates.
- Do not expose phpMyAdmin or backup ZIP files publicly.
- Review storage/logs/laravel.log for errors but do not leave logs web-accessible.

14. Common Troubleshooting

Problem	Checks / fix
500 Internal Server Error	Verify PHP 8.1/8.2, .env values, APP_KEY, storage/bootstrap permissions, vendor folder and Laravel log. Run
CSS/images missing	Confirm root .htaccess is present, public assets exist, APP_URL uses HTTPS, and mod_rewrite is enabled.
Database connection error	Check DB_DATABASE, DB_USERNAME, DB_PASSWORD, DB_HOST; confirm the DB user is assigned to the
Admin login opens but credentials fail	Use the known project admin password or perform a secure password reset; do not replace the hash with guess
Form email not received	Replace Mailpit with real SMTP; verify port/encryption, mailbox credentials and sender address.
Upload fails	Check upload_max_filesize/post_max_size, public/uploads write access and file validation limits.
419 / CSRF error	Clear sessions/cache, verify HTTPS/session cookie settings and storage/framework write access.
Class not found / vendor error	Confirm vendor/autoload.php exists. If missing, run composer install --no-dev --optimize-autoloader with PHP 8

15. Backup and Update Workflow

1. Create a full ZIP backup of the working website files.
2. Export the current production database from phpMyAdmin.
3. Apply code changes to a staging copy first when possible.
4. Preserve .env, uploaded media and any server-specific configuration.
5. After code updates, run php artisan optimize:clear.
6. Test home, services, coaching/products, blog, consultants, contact/enquiry, user login and admin dashboard.
7. Only then replace the production build.

16. Final Go-Live Test

- Homepage loads without PHP/Laravel errors
- Visa Services pages and cards open correctly
- Coaching / products pages open correctly
- Blog/news list and details open
- Consultants/team page loads images and profiles
- Contact/enquiry submits and is visible to admin
- Admin dashboard loads and counts/charts do not error
- User login/logout works
- Mobile/tablet/desktop layouts are responsive
- Logo, favicon and social/SEO preview assets display
- HTTPS is forced with no mixed-content warnings
- Backup taken after successful launch