

ViZuTrade Installation Guide

A beginner-friendly professional guide for installing ViZuTrade on cPanel hosting using the built-in web installer.

Project ViZuTrade	Format PDF + Markdown	Focus cPanel installation
-----------------------------	---------------------------------	-------------------------------------

Contents

- 1. Screenshot Placeholders
- 2. 1. Introduction
- 3. 2. Server Requirements
- 4. 3. Uploading Project to cPanel
- 5. 4. File Permissions
- 6. 5. Starting Installer
- 7. 6. Requirements Check Page
- 8. 7. Permissions Check Page
- 9. 8. Database Setup Page
- 10. 9. App Setup Page
- 11. 10. Running Installation

- 12. 11. Installation Complete
- 13. 12. Admin Login
- 14. 13. Common Errors and Fixes
- 15. 14. Security Notes
- 16. 15. Final Checklist
- 17. Quick Installer Flow Summary
- 18. Support Note

Purpose: This guide explains how to install ViZuTrade on cPanel hosting using the built-in web installer.

Audience: Beginners, site owners, and administrators who want to upload the ViZuTrade Laravel project to hosting and complete installation from a browser.

Brand: ViZuTrade uses a clean blue and white trading dashboard style.

Screenshot Placeholders

This documentation includes screenshot placeholders such as:

[SCREENSHOT: Server Requirements Page]

When you capture real screenshots later, place them in:

`docs/screenshots/`

Recommended screenshot file names:

- ? 01-welcome-page.png
 - ? 02-server-requirements-page.png
 - ? 03-folder-permissions-page.png
 - ? 04-database-setup-page.png
 - ? 05-app-setup-page.png
 - ? 06-installation-progress-page.png
 - ? 07-installation-complete-page.png
-

1. Introduction

ViZuTrade is a Laravel-based trading journal and platform management system. It includes user authentication, trading tools, trade tracking, subscriptions, admin controls, support tickets, maintenance mode, API settings, and dashboard features for traders.

This guide teaches you how to install ViZuTrade on cPanel hosting using the web installer. The installer lets you complete setup from a browser instead of manually editing every Laravel configuration file.

The installer helps you:

- ? Check server requirements
 - ? Check folder permissions
 - ? Enter database details
 - ? Test the database connection
 - ? Create or update the `.env` file
 - ? Generate the Laravel `APP_KEY`
 - ? Run migrations
 - ? Run seeders
 - ? Create the first admin account
 - ? Lock the installer after installation
-

2. Server Requirements

Before uploading ViZuTrade, make sure your hosting account supports Laravel.

Required PHP Version

ViZuTrade requires:

? PHP 8.2 or higher

In cPanel, you can usually check this from:

[cPanel -> Select PHP Version](#)

If your hosting is using an older PHP version, change it to PHP 8.2 or higher.

Required PHP Extensions

The installer checks these extensions:

? OpenSSL

? PDO

? PDO MySQL

? Mbstring

? Tokenizer

? XML

? ctype

? JSON

? BCMath

? Fileinfo

? GD or Imagick

? Curl

? Zip

Most good Laravel hosting plans include these by default. If one is missing, enable it from cPanel PHP extensions or ask your hosting provider.

MySQL Requirement

ViZuTrade uses MySQL for the database.

You need:

? A MySQL database

? A MySQL database user

? The database user assigned to the database

? Full permissions for that database user

In cPanel, this is usually done from:

[cPanel -> MySQL Databases](#)

Composer Requirement

Composer is used to install Laravel PHP dependencies.

If you are uploading a complete project with the `vendor` folder already included, you may not need to run Composer on cPanel.

If your hosting supports terminal access, the recommended command is:

```
composer install --no-dev --optimize-autoloader
```

Run this command inside the main ViZuTrade project folder, not inside the `public` folder.

Node.js Requirement

Node.js is used to build frontend assets such as Tailwind CSS and Vite files.

For production hosting, you should normally build assets before uploading:

```
npm install  
npm run build
```

Run these commands locally inside the ViZuTrade project folder before zipping the project. After the build is complete, upload the project with the generated `public/build` folder.

cPanel Hosting Requirements

Your hosting should allow:

- ? PHP 8.2+
- ? MySQL database creation
- ? Writable `storage` folder
- ? Writable `bootstrap/cache` folder
- ? Domain or subdomain pointing to Laravel `public` folder
- ? File Manager or FTP access
- ? Terminal access if you want to run Composer commands on the server

3. Uploading Project to cPanel

This section explains how to prepare and upload ViZuTrade.

Step 1: Prepare the Project Locally

On your computer, open the ViZuTrade project folder.

Run:

```
composer install  
npm install  
npm run build
```

What these commands do:

- ? `composer install` installs Laravel PHP dependencies.
- ? `npm install` installs frontend dependencies.
- ? `npm run build` creates production-ready CSS and JavaScript files.

Step 2: Zip the Project

Select all ViZuTrade project files and create a ZIP file.

Do not include unnecessary development folders if your hosting will install dependencies itself. For beginner cPanel upload, it is often easier to include:

- ? `app`
- ? `bootstrap`
- ? `config`
- ? `database`
- ? `public`
- ? `resources`
- ? `routes`
- ? `storage`
- ? `vendor`
- ? `composer.json`
- ? `composer.lock`
- ? `package.json`
- ? `package-lock.json`
- ? `.env.example`

Do not upload your local `.env` file if it contains private passwords or API keys.

Step 3: Upload to cPanel File Manager

Open:

`cPanel -> File Manager`

Upload the ZIP file to a folder outside `public_html` if your hosting allows it. A good structure is:

```
home/yourcpaneluser/vizustrade/  
home/yourcpaneluser/public_html/
```

Then extract the ZIP file.

Step 4: Laravel Public Folder Setup

Laravel should serve files from the `public` folder, not the project root.

Best setup:

```
vizustrade/ -> full Laravel project  
public_html/ -> points to vizustrade/public
```

Some cPanel hosts allow you to set the domain document root to:

```
/home/yourcpaneluser/vizustrade/public
```

If your host does not allow changing the document root, you may need to move the contents of ViZuTrade's `public` folder into `public_html` and update `index.php` paths. The safer option is to ask your hosting provider to point the domain to Laravel's `public` folder.

4. File Permissions

ViZuTrade needs permission to write cache files, logs, sessions, uploads, and the installer lock file.

Required Writable Folders

Make sure these are writable:

```
storage
bootstrap/cache
```

Recommended permissions:

```
storage: 775
bootstrap/cache: 775
```

Some shared hosting may require `755` or `775` depending on server ownership. Avoid `777` unless your hosting provider specifically tells you to use it.

`.env` File Permission

The installer creates or updates the `.env` file.

If `.env` does not exist, the project root must be writable.

If `.env` already exists, the `.env` file must be writable.

Recommended:

```
.env: 644
```

During installation, if the installer cannot write `.env`, fix permissions and try again.

5. Starting Installer

After uploading the project and pointing your domain to the Laravel `public` folder, open your website URL in a browser:

```
https://yourdomain.com
```

If ViZuTrade is not installed yet, it should automatically redirect to:

```
https://yourdomain.com/install
```

On the Welcome page, click:

```
Start Installation
```

6. Requirements Check Page

The requirements page checks if your hosting has the PHP version and extensions needed by ViZuTrade.

If all checks pass, continue to the next step.

If any item fails, contact your hosting provider or enable the missing PHP extension in cPanel.

[SCREENSHOT: Server Requirements Page]

Place the real screenshot later at:

```
docs/screenshots/02-server-requirements-page.png
```

7. Permissions Check Page

The permissions page checks if Laravel can write to important folders.

The installer checks:

? `storage`

? `bootstrap/cache`

? `.env` file or project root

If a folder is not writable, update permissions in cPanel File Manager.

[SCREENSHOT: Folder Permissions Page]

Place the real screenshot later at:

```
docs/screenshots/03-folder-permissions-page.png
```

8. Database Setup Page

The database setup page connects ViZuTrade to MySQL.

Create your database first in:

```
cPanel -> MySQL Databases
```

Then enter the database details into the installer.

DB Host

Usually:

```
localhost
```

Some hosts use a remote database hostname. If `localhost` fails, check your cPanel database page or ask your hosting provider.

DB Port

Usually:

```
3306
```

DB Database

Enter the full database name created in cPanel.

Example:

```
cpaneluser_vizustrade
```

DB Username

Enter the full database username created in cPanel.

Example:

```
cpaneluser_vizuuser
```

DB Password

Enter the password for the database user.

The installer does not display your password again after submission.

Test Connection Button

Click:

```
Test Connection
```

If the connection is successful, click:

```
Save and Continue
```

[SCREENSHOT: Database Setup Page]

Place the real screenshot later at:

<docs/screenshots/04-database-setup-page.png>

9. App Setup Page

The app setup page creates your basic application settings and first admin account.

App Name

Enter:

```
ViZuTrade
```

App URL

Enter your website URL.

Example:

```
https://yourdomain.com
```

Do not add a slash at the end.

Admin Name

Enter the name of the first admin user.

Example:

```
ViZuTrade Admin
```

Admin Email

Enter the email address you will use to log in as admin.

Example:

```
admin@yourdomain.com
```

Admin Password

Create a strong password. The password should be at least 8 characters and include uppercase letters, lowercase letters, and numbers.

Example format:

```
StrongPass123
```

Use your own secure password. Do not use the example password on a live website.

[SCREENSHOT: [App Setup Page](#)]

Place the real screenshot later at:

`docs/screenshots/05-app-setup-page.png`

10. Running Installation

After saving the app setup, click:

```
Run Installation
```

The installer will do these tasks:

Create or Update `.env`

The `.env` file stores private environment settings such as:

- ? App name
- ? App URL
- ? Database host
- ? Database name
- ? Database username
- ? Database password

Generate `APP_KEY`

Laravel needs `APP_KEY` to encrypt sessions, passwords, and secure data.

The installer generates this automatically.

Run Migrations

Migrations create database tables such as:

- ? users
- ? trades
- ? goals
- ? trading accounts
- ? API settings
- ? subscription plans
- ? payments
- ? support tickets
- ? site settings

Run Seeders

Seeders add important default data such as:

- ? Trading symbols
- ? Subscription plans
- ? Payment gateways

Create Admin Account

The installer creates the admin account using the name, email, and password entered on the App Setup page.

[SCREENSHOT: [Installation Progress Page](#)]

Place the real screenshot later at:

`docs/screenshots/06-installation-progress-page.png`

11. Installation Complete

When installation is finished, ViZuTrade creates this lock file:

```
storage/app/installed
```

This lock file is important because it disables the installer after installation.

You should see buttons for:

? Login

? Admin Dashboard

Click Login and sign in with the admin account created during installation.

[SCREENSHOT: Installation Complete Page]

Place the real screenshot later at:

```
docs/screenshots/07-installation-complete-page.png
```

12. Admin Login

Open:

```
https://yourdomain.com/login
```

Enter:

? Admin email

? Admin password

After login, admin users should go to:

```
https://yourdomain.com/admin/dashboard
```

The admin dashboard is for platform management only. Admin users should not use the admin account as a normal trader account.

If you want to use ViZuTrade as a trader, create a separate normal user account.

13. Common Errors and Fixes

500 Server Error

Possible causes:

? Wrong PHP version

? Missing PHP extension

? File permission issue

? `.env` missing

? Laravel cache problem

Fix:

1. Check PHP version is 8.2 or higher.

2. Check required extensions.
3. Make `storage` and `bootstrap/cache` writable.
4. Clear cache from terminal if available:

```
php artisan optimize:clear
```

Database Connection Failed

Possible causes:

- ? Wrong DB host
- ? Wrong DB name
- ? Wrong DB username
- ? Wrong DB password
- ? Database user not assigned to database

Fix:

1. Confirm database exists in cPanel.
2. Confirm user exists.
3. Assign the user to the database.
4. Give the user all privileges.
5. Try `localhost` as DB host.
6. Click Test Connection again.

Permission Denied

Possible causes:

- ? `storage` is not writable
- ? `bootstrap/cache` is not writable
- ? `.env` cannot be created or updated

Fix:

Set permissions:

```
storage: 775
bootstrap/cache: 775
.env: 644
```

If the problem continues, ask your hosting provider to correct file ownership.

APP_KEY Missing

Laravel needs `APP_KEY`.

The installer should generate it automatically.

If it is missing and you have terminal access, run:

```
php artisan key:generate
```

Then clear cache:

```
php artisan optimize:clear
```

Migration Failed

Possible causes:

- ? Database connection failed
- ? Database user does not have enough permissions
- ? Tables already exist from an old failed installation

Fix:

1. Check database credentials.
2. Give the database user all privileges.
3. If this is a fresh install, empty the database and run the installer again.

Be careful: emptying a database deletes data.

Storage Link Issue

If uploaded screenshots or public files do not show, create Laravel storage link.

Run:

```
php artisan storage:link
```

If your cPanel does not allow symbolic links, ask your hosting provider for help.

CSS Not Loading

Possible causes:

- ? `npm run build` was not run
- ? `public/build` folder was not uploaded
- ? Domain is not pointing to Laravel `public` folder
- ? Browser cache

Fix:

1. Run locally:

```
npm install  
npm run build
```

2. Upload the generated `public/build` folder.
3. Clear browser cache.
4. Confirm your domain points to Laravel `public`.

Login Not Working

Possible causes:

- ? Admin user was not created
- ? Wrong password
- ? Session folder not writable
- ? App URL is wrong

Fix:

1. Confirm `storage/framework/sessions` is writable.
 2. Confirm `APP_URL` in `.env` matches your website.
 3. Try password reset if mail is configured.
 4. If needed, rerun installation on a fresh database.
-

14. Security Notes

Follow these security rules for a live website.

Do Not Upload `.env` to GitHub

The `.env` file contains private data such as:

- ? Database password
- ? API keys
- ? App secrets
- ? Payment gateway keys

Keep `.env` private.

Keep API Keys Private

Never put API keys inside Blade files, JavaScript files, screenshots, or GitHub.

API keys should stay in:

- ? `.env`
- ? Encrypted database settings managed from the admin panel

Installer Lock

After installation, the installer creates:

```
storage/app/installed
```

Do not delete this file unless you intentionally want the installer to run again.

Use a Strong Admin Password

Use a password that is:

- ? At least 12 characters for production
- ? Not reused from another website

? Includes uppercase, lowercase, numbers, and symbols

Set Debug Mode Off

For production, `.env` should contain:

```
APP_ENV=production
APP_DEBUG=false
```

Do not run a live website with debug mode enabled because it can expose sensitive error details.

15. Final Checklist

Use this checklist before announcing the website is ready.

- Project files uploaded to cPanel
- Domain points to Laravel `public` folder
- PHP version is 8.2 or higher
- Required PHP extensions are enabled
- MySQL database created
- MySQL user created
- MySQL user assigned to database
- `storage` folder is writable
- `bootstrap/cache` folder is writable
- Installer welcome page opens
- Server requirements passed
- Folder permissions passed
- Database connection tested successfully
- App setup completed
- Installation completed
- `.env` file created or updated
- `APP_KEY` generated
- Migrations completed
- Seeders completed
- Admin account created
- `storage/app/installed` lock file created
- Installer is blocked after install
- Admin login works
- Admin dashboard opens
- Website frontend opens
- Maintenance mode tested

Payment mode checked

CSS and JavaScript load correctly

Quick Installer Flow Summary

```
Upload project
Point domain to /public
Open website URL
Start installer
Check requirements
Check permissions
Enter database details
Test database connection
Enter app and admin details
Run installation
Login as admin
```

Support Note

If installation fails, read the error message carefully and check the relevant section in this guide. Most beginner installation problems come from one of these areas:

- ? Wrong database credentials
- ? Missing PHP extension
- ? Incorrect file permissions
- ? Domain not pointing to the Laravel `public` folder
- ? Frontend assets not built before upload