

1. Introduction

Streamit - Video Streaming offers a seamless way to explore and enjoy a wide variety of entertainment content. With an extensive library of movies, TV shows, videos, and live channels, users can easily find and stream content anytime, anywhere. Streamit provides personalised recommendations based on user ratings and preferences and allows users to manage their own watchlist. It also offers the ability to continue watching shows or movies from where users left off. Designed with a user-friendly interface, Streamit makes it easy to navigate and discover trending content.

Key Features:

- Browse and watch movies, TV shows, videos, and live channels
- Personalised recommendations based on your watch history
- Create and manage your watchlist
- Continue watching content from where you left off
- Filter content by genre, language, and popularity
- Dark mode for a comfortable viewing experience
- Multiple sign-in options including Google and OTP
- Multi-language support for a global audience

Streamit is your go-to platform for an immersive streaming experience, offering high-quality content with ease of access and management.

Visit our platform: <https://apps.iqonic.design/streamit-laravel>

2. Introduction of Laravel

What is Laravel?

Laravel is a PHP-based web framework for building high-end web applications with clear and elegant syntax. It offers a robust collection of tools and provides application architecture. Moreover, it incorporates characteristics of technologies like ASP.NET MVC, CodeIgniter, Ruby on Rails, and others. As an open-source framework, Laravel helps developers save time and reduce the need to build websites from scratch.

Key Features:

- Browse and watch movies, TV shows, videos, and live channels
- Personalised recommendations based on your watch history
- Create and manage your watchlist
- Continue watching content from where you left off
- Filter content by genre, language, and popularity
- Dark mode for a comfortable viewing experience
- Multiple sign-in options including Google and OTP
- Multi-language support for a global audience

Streamit is your go-to platform for an immersive streaming experience, offering high-quality content with ease of access and management.

Click [here](#) to install Laravel on a Laptop or Desktop.

Requirements

- PHP 8.2 or greater
 - OpenSSL PHP Extension
 - PDO PHP Extension
 - Mbstring PHP Extension
 - Tokenizer PHP Extension
 - XML PHP Extension
 - Ctype PHP Extension
 - JSON PHP Extension
 - GD PHP Extension (or Imagick PHP Extension)
 - PHP Fileinfo extension
 - PHP Zip Archive
 - Rewrite Module (Apache or Nginx)
-

php.ini Requirements

- `open_basedir` must be disabled
 - `output_buffering` must be enabled
 - `output_buffering = On`
-

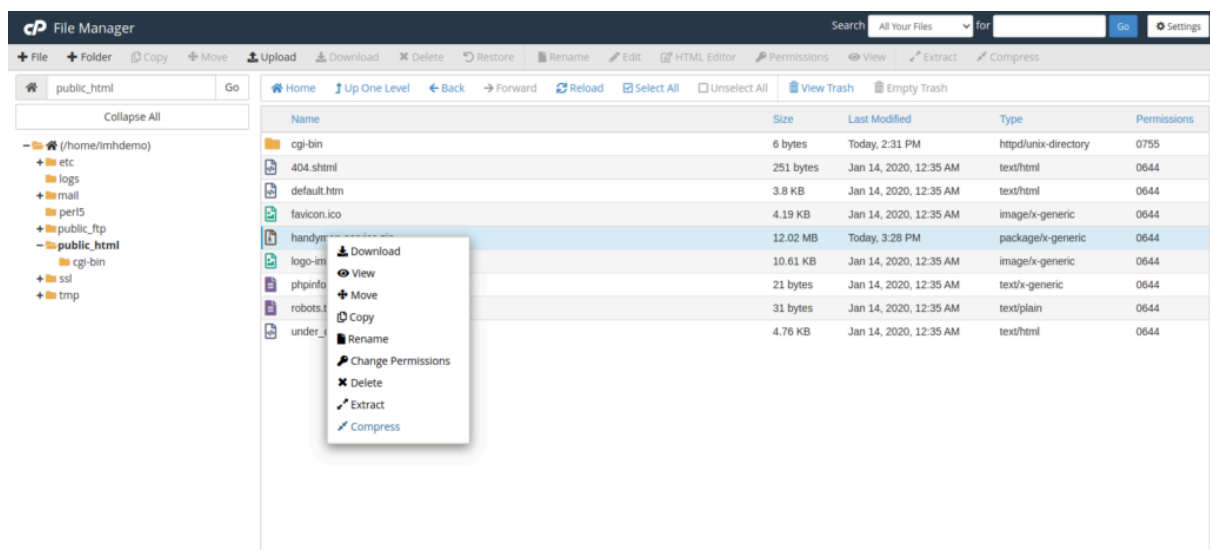
File and Folder Permissions

- `/bootstrap` 775
- `/storage` 775

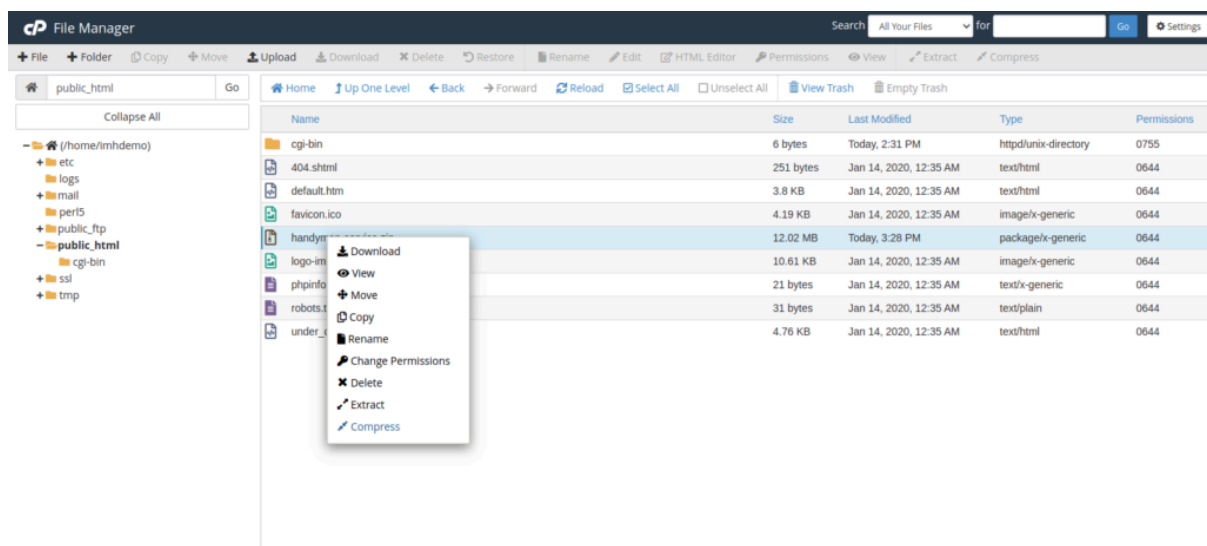
3. Basic Server Setup

1. Retrieve the project zip file from CodeCanyon, then extract it.

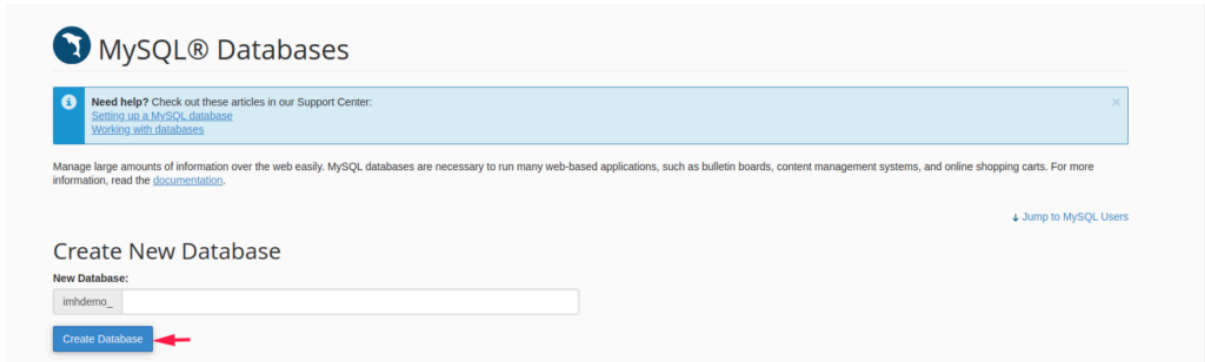
Proceed to the following path: streamit_app_source_code -> admin-panel, upload the zip file to your cPanel, and extract it to the public_html directory.



2. Extract the zip file to that directory.



3. Create a new database from your server MySQL database.



The screenshot shows the 'MySQL® Databases' management interface. At the top, there's a header with the MySQL logo and a navigation bar. Below the header, there's a blue banner with a help icon and text: 'Need help? Check out these articles in our Support Center: [Setting up a MySQL database](#) [Working with databases](#)'. Below this, a paragraph of text explains the purpose of MySQL databases. A link 'Jump to MySQL Users' is on the right. The main section is titled 'Create New Database'. It contains a text input field labeled 'New Database:' with the value 'imhdemo_'. Below the input field is a blue button labeled 'Create Database', which is highlighted with a red arrow.

4. Create a DB user and database, then assign them to each other.



The screenshot shows the 'MySQL Users' management interface. At the top, there's a header with the MySQL logo and a navigation bar. Below the header, there's a blue banner with a help icon and text: 'Need help? Check out these articles in our Support Center: [Setting up a MySQL database](#) [Working with databases](#)'. Below this, a paragraph of text explains the purpose of MySQL databases. A link 'Jump to MySQL Users' is on the right. The main section is titled 'Add New User'. It contains several form fields: 'Username' with the value 'imhdemo_ root', 'Password' with a masked input, and 'Password (Again)' with a masked input. Below these fields is a 'Strength' indicator showing a green bar and the text 'Very Strong (100/100)'. To the right of the strength indicator is a 'Password Generator' button. At the bottom of the form is a blue button labeled 'Create User', which is highlighted with a red arrow.

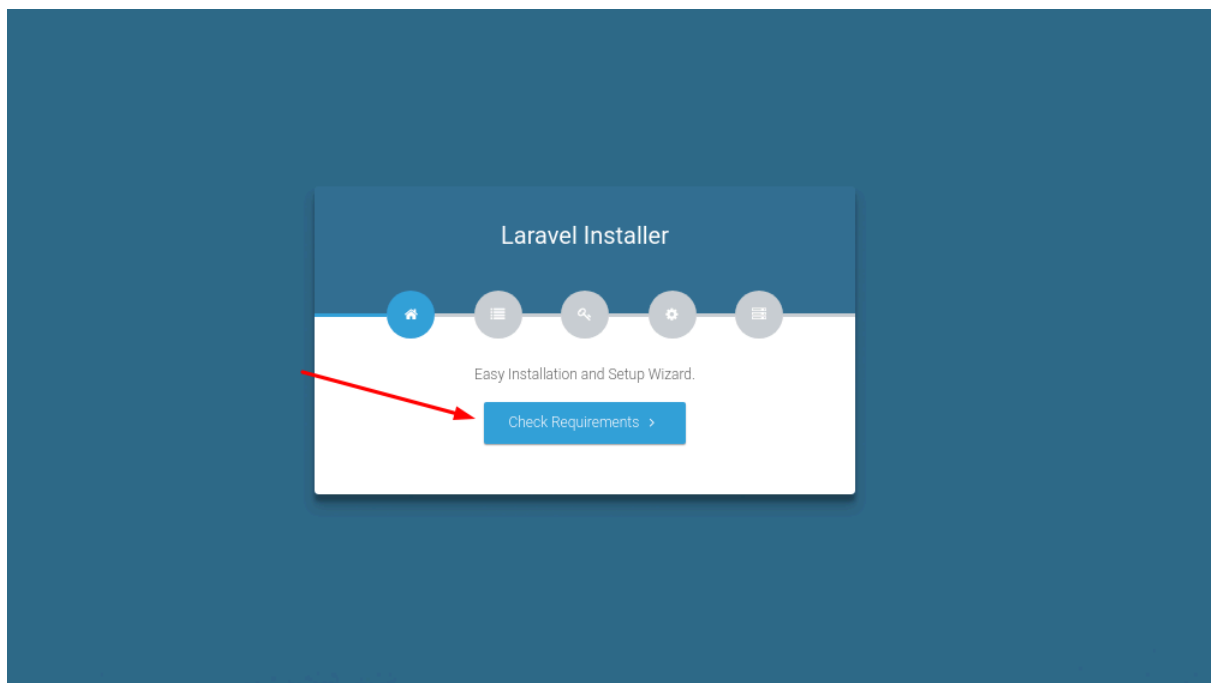
Successful!!

After completing all the steps, just hit your URL in the browser to access the site.

4. Installation

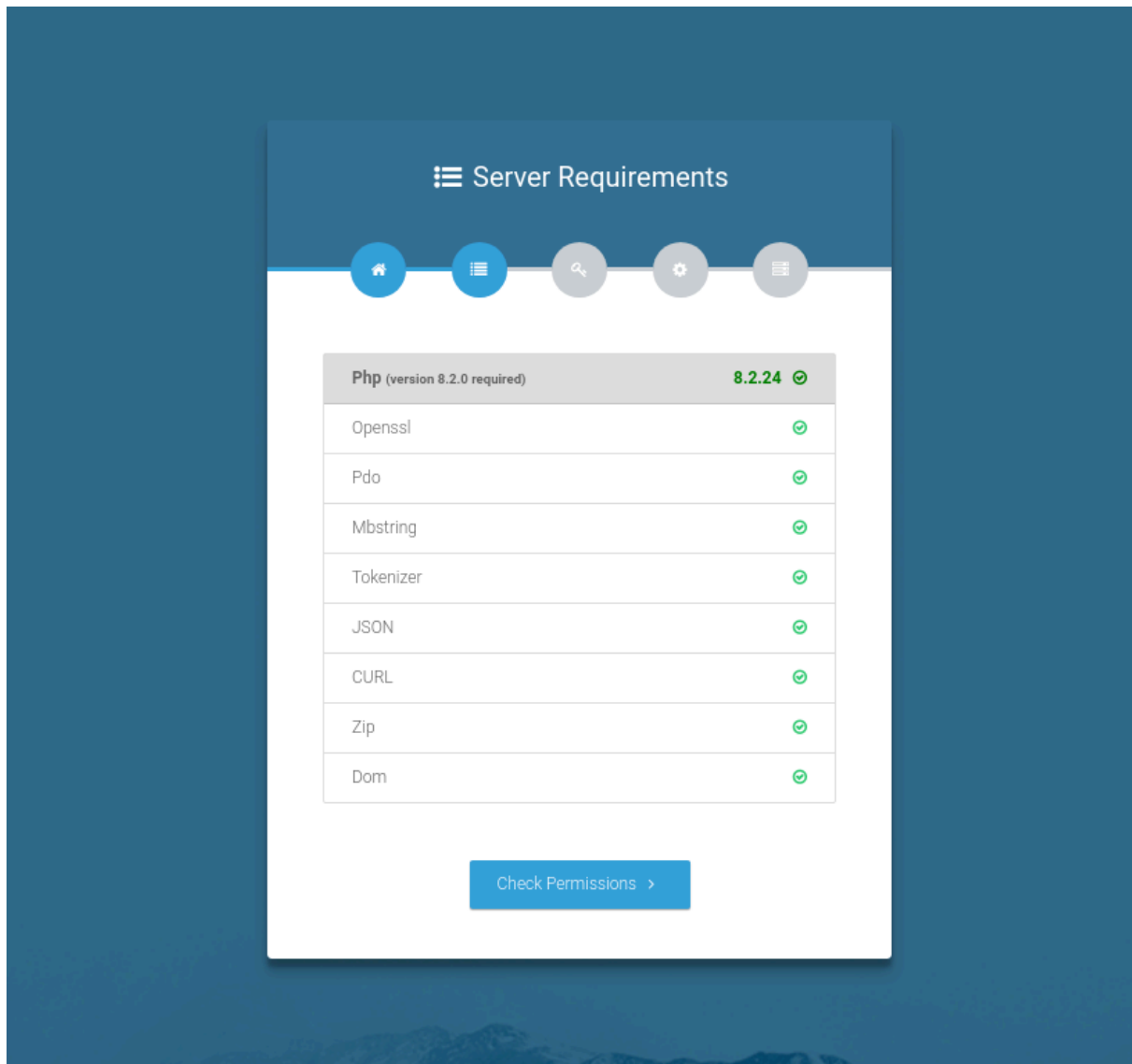
Automatic Installation and Setup

1. Go to: <https://yourdomainname.com/install>
2. You will see the welcome screen of Streamit setup wizard. Click "Check Requirements" to continue.



3. In the "Server Requirements" step, you can see if your server meets Streamit's requirements.

If all requirements are checked, you are ready to move on to the next step. Click the Check Permission button.



4. In this step, you'll verify that you are in the correct directory and that the files and folders within it have the necessary file permissions.

After granting all the necessary permissions, click the Configure Environment button.

Permissions



Storage/framework/	 775
Storage/logs/	 775
Bootstrap/cache/	 775

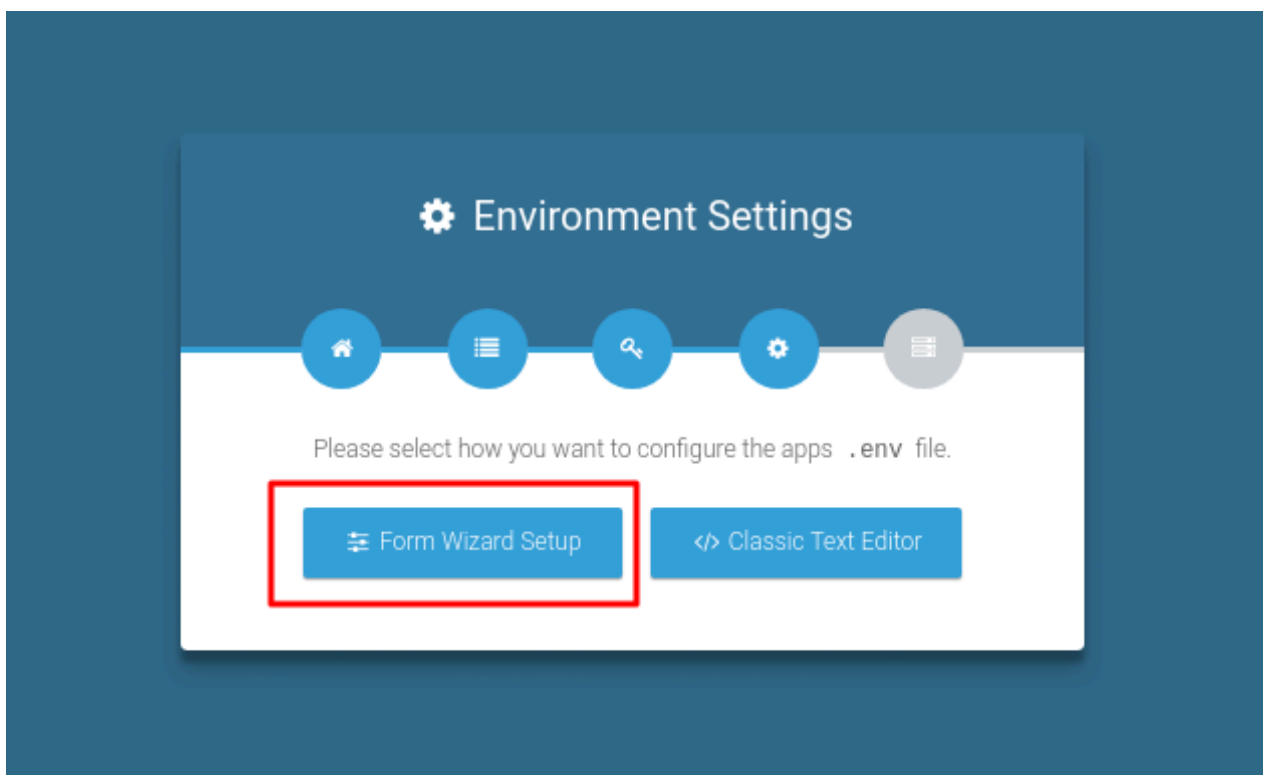
[Configure Environment >](#)

5. In this step, you have two options to configure the environment:

1. Use the wizard setup.
2. Utilise the classic text editor.

Option -1 - To configure the environment using the wizard setup.

1. Click the Form Wizard setup button.

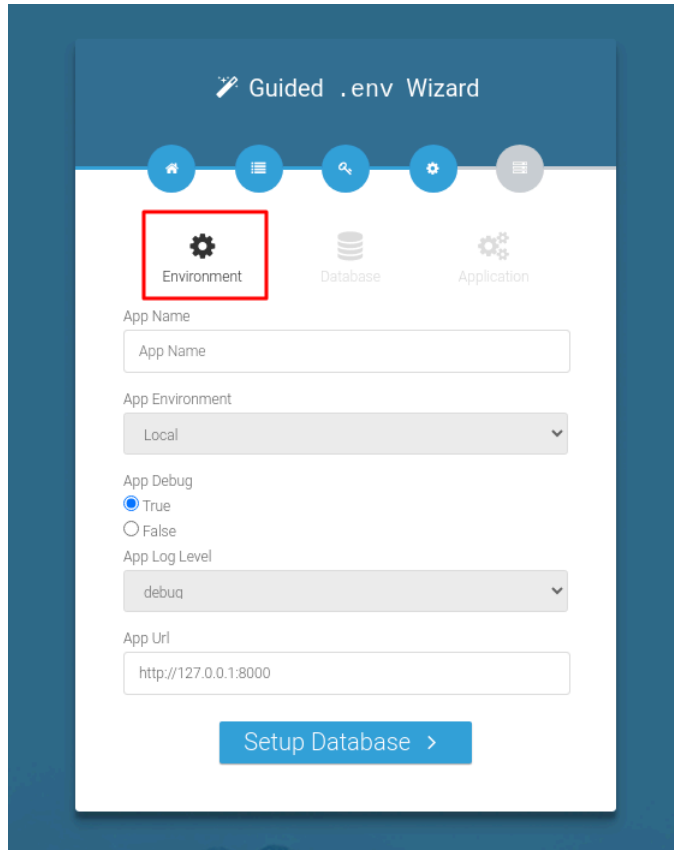


2. In this step, you will enter your site's information.

- App Name: Enter the name of your site/project in this field.
- App Environment: Choose the environment.
- App Debugging Mode: Choose the debugging mode.

– App URL: Enter the URL of your site/project in this field. Please enter it as: `https://yourdomain.com` and do not add a “/” at the end.

After filling all the fields, click the Setup Database button.



The image shows a 'Guided .env Wizard' interface. At the top, there are five circular icons: a home icon, a list icon, a magnifying glass icon, a gear icon (which is highlighted with a red box and labeled 'Environment'), and a document icon. Below these icons are three tabs: 'Environment', 'Database', and 'Application'. The 'Environment' tab is active. The form contains the following fields:

- App Name:** A text input field with the placeholder text 'App Name'.
- App Environment:** A dropdown menu with 'Local' selected.
- App Debug:** Radio buttons for 'True' (selected) and 'False'.
- App Log Level:** A dropdown menu with 'debug' selected.
- App Url:** A text input field with the placeholder text 'http://127.0.0.1:8000'.

At the bottom of the form is a blue button labeled 'Setup Database >'.

3. In this step, you will enter the information of the database you created .

– Database Connection: Choose your database connection type.

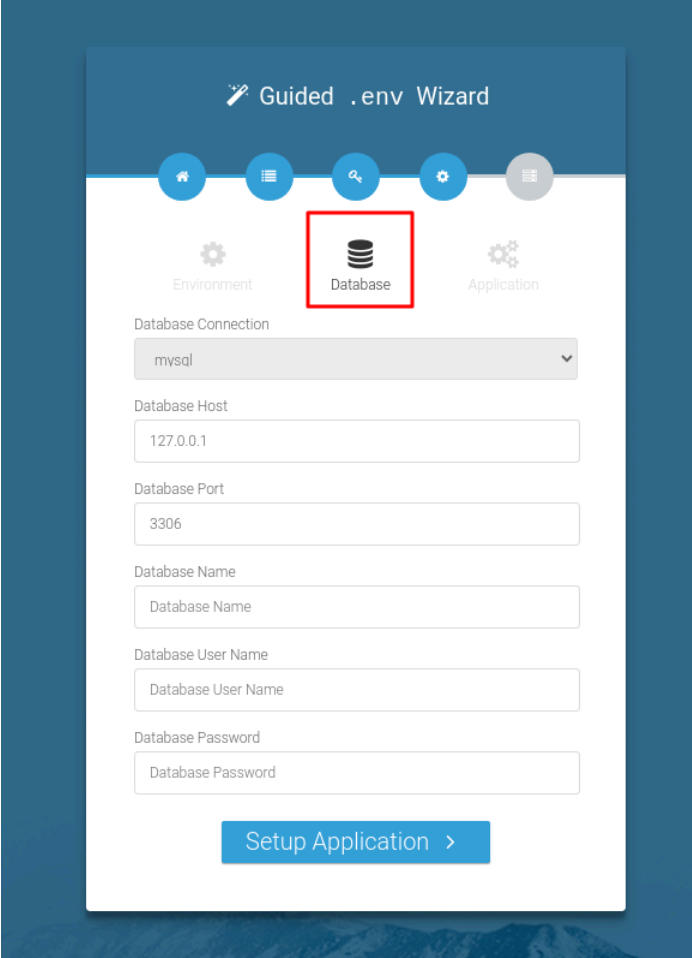
– Database Host: Enter your database host address. By default, it is 127.0.0.1 or localhost.

– Database Port: Enter the access port of your database. By default, it is 3306.

– Database User Name: Enter the username of your database.

– Database Password: Enter the password of your database.

- Database Name: Enter the name of your database.
After filling all the fields, click the Setup Application button



The image shows a 'Guided .env Wizard' interface. At the top, there are five circular icons: a home icon, a list icon, a magnifying glass icon, a gear icon, and a document icon. Below these icons are three main sections: 'Environment' (gear icon), 'Database' (cylinder icon, highlighted with a red box), and 'Application' (gear icon). The 'Database' section is active and contains the following fields:

- Database Connection:** A dropdown menu with 'mysql' selected.
- Database Host:** A text input field containing '127.0.0.1'.
- Database Port:** A text input field containing '3306'.
- Database Name:** A text input field containing 'Database Name'.
- Database User Name:** A text input field containing 'Database User Name'.
- Database Password:** A text input field containing 'Database Password'.

At the bottom of the form is a blue button labeled 'Setup Application >'.

4. In this step, you will input the necessary configurations for log settings, email, Pusher, and Redis.

After filling all the fields, click the save and install button.

Guided .env Wizard



Environment



Database



Application

Broadcasting, Caching, Session, & Queue

Broadcast Driver 

log

Cache Driver 

file

Session Driver 

Redis Driver

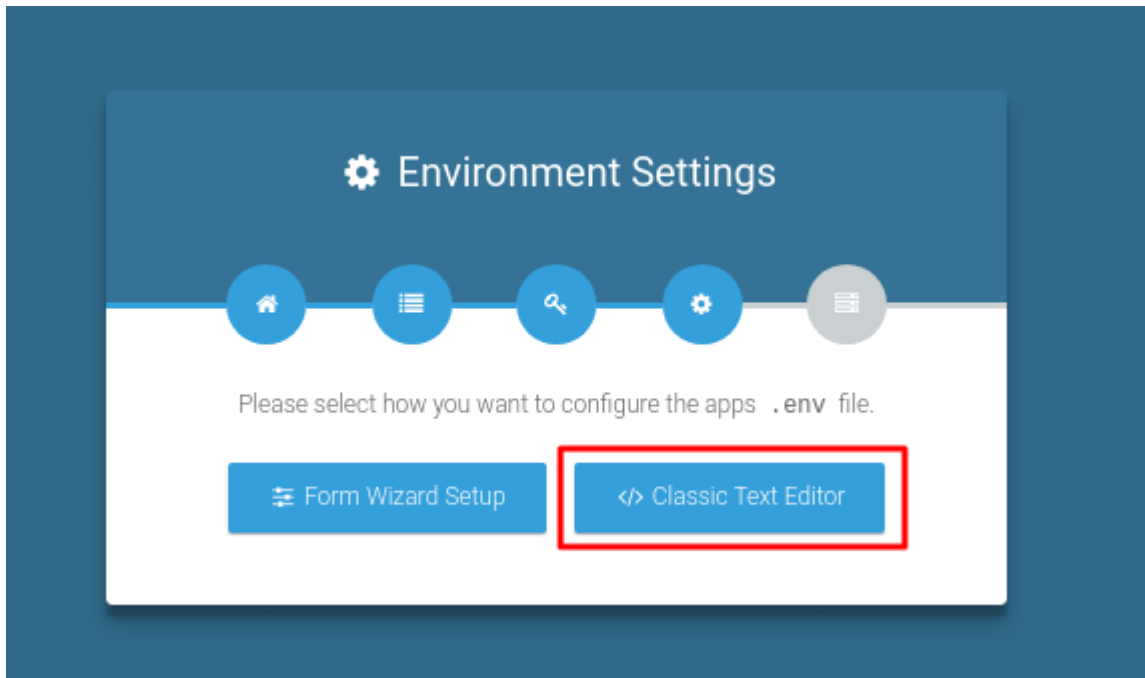
Mail

Pusher

Install >

Option 2: To configure the environment using the classic text editor, simply follow these steps

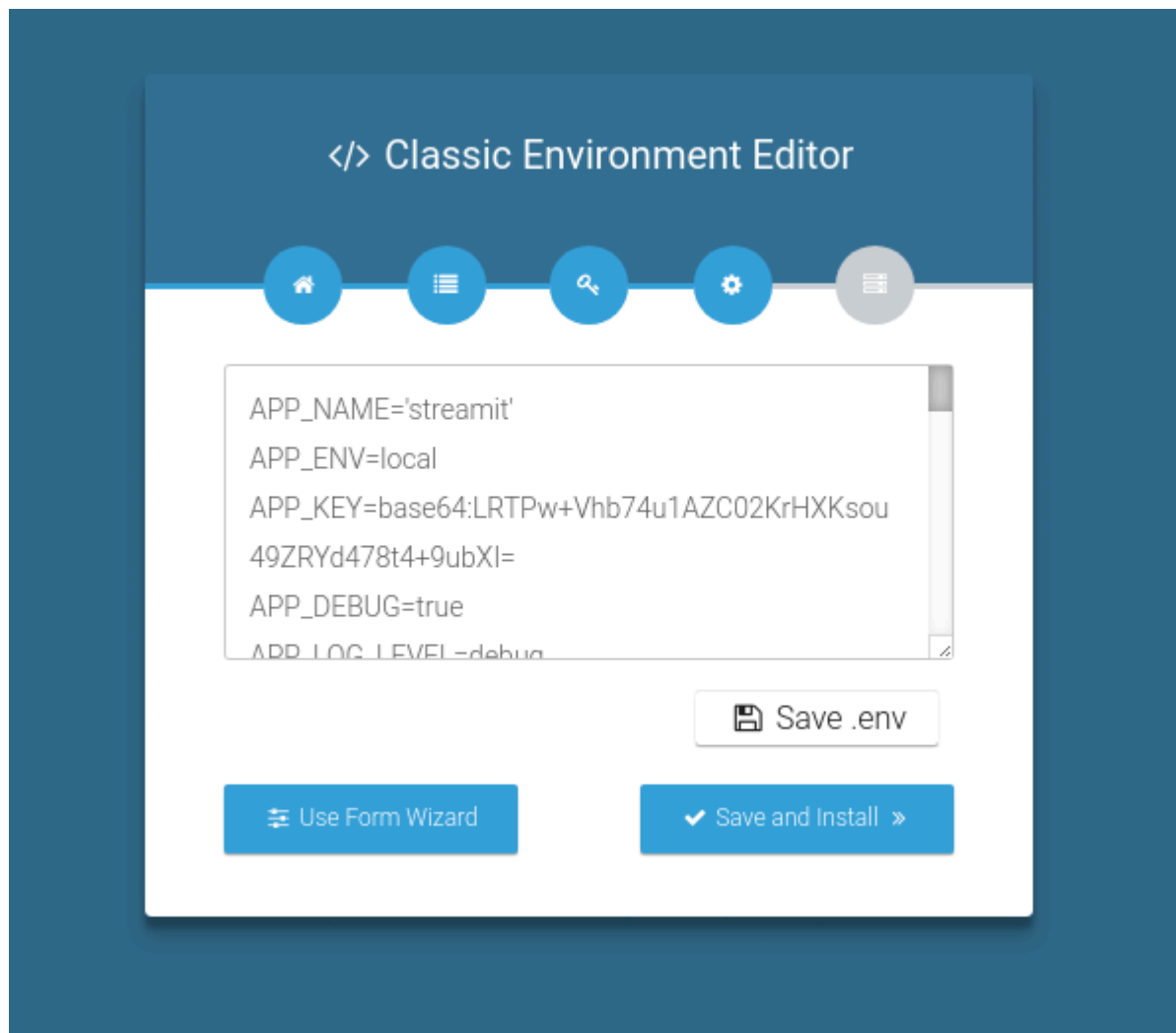
1. Click the "Classic Text Editor" button.



2. In this step, you have to configure the environment file (`.env`) with the necessary information.

After filling the necessary information, click the Save .env button.

After clicking the Save .env button, click the save and install button.



Manual Installation and Setup

You're setting up the development environment to create your web application.

1. Start command prompt window or terminal and change directory to streamit-laravel-web

- `cd streamit-laravel-web`

2. For installing vendor Run in terminal or CMD:

- `composer install`

3. Copy .env file Run in terminal or CMD:

- `cp .env.example .env`

4. For generating key Run in terminal or CMD:

- `php artisan key:generate`

5. Create database with name "streamit" in your sqlyog ,laragon,xampp,wamp

- Open and edit the /.env file and provide your server details:

`DB_CONNECTION="mysql"`

`DB_HOST="127.0.0.1"`

DB_PORT="3306"

DB_DATABASE="YOUR DATABASE NAME"

DB_USERNAME="YOUR USERNAME"

DB_PASSWORD="YOUR PASSWORD"

- If you want a dummy data into the application, you have to set this key to `true`

IS_DUMMY_DATA=false

- Then run the below commands

- php artisan migrate
- php artisan db:seed

OR

- php artisan migrate:fresh --seed

6. For installing node_modules Run in terminal or CMD:

- npm install

7. For building css and js Run in terminal or CMD:

- npm run dev

OR

- npm run build

OR

- npm run watch

8. To run project

- php artisan serve

9. Login Credentials

For Admin Login

<https://apps.iqonic.design/streamit-laravel/admin/login>

Username:admin@streamit.com

Password:12345678

5. Configuration

Mail Configuration

1. Open and edit the /.env file and provide following details:

MAIL_DRIVER="YOUR MAIL DRIVER"

MAIL_HOST="YOUR MAIL HOST"

MAIL_PORT="YOUR MAIL PORT"

MAIL_USERNAME="YOUR MAIL ID"

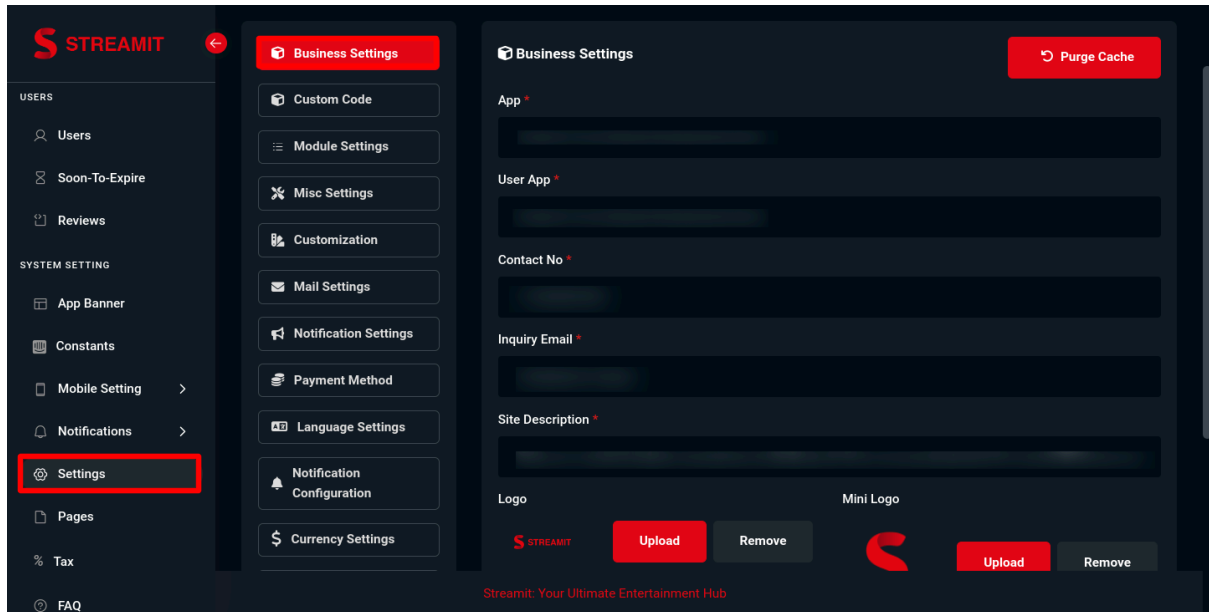
MAIL_PASSWORD="YOUR MAIL PASSWORD"

MAIL_ENCRYPTION="YOUR MAIL ENCRYPTION"

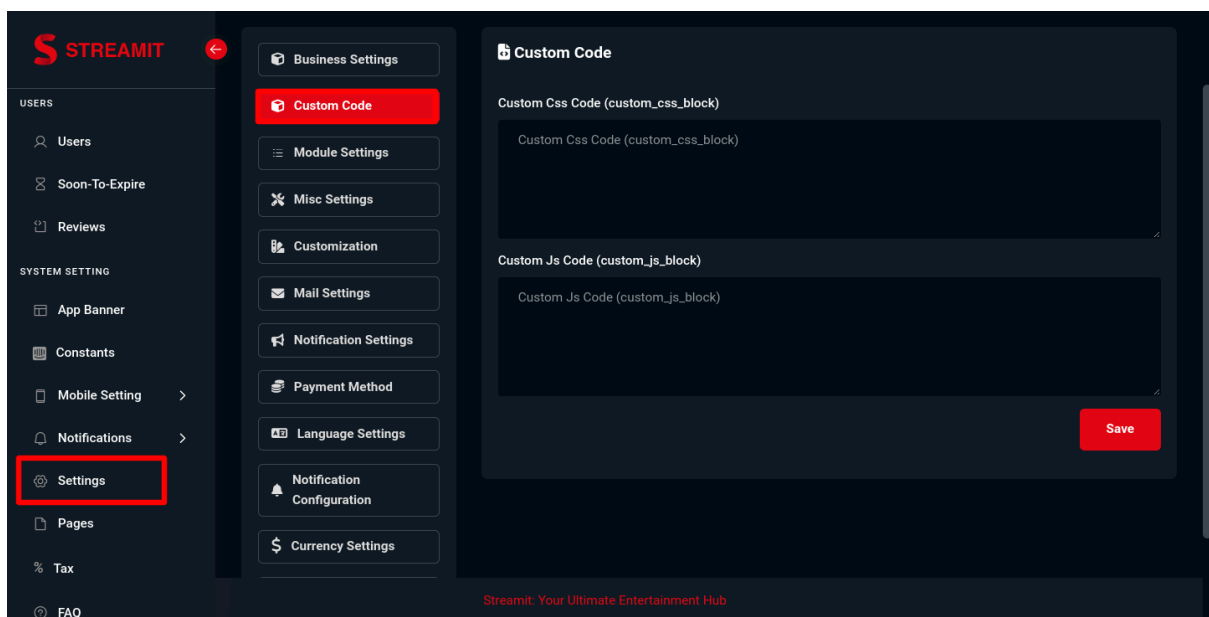
2. Enter mail host name, mail port, mail username(Mail ID), mail password and email encryption type.

General Settings Configuration

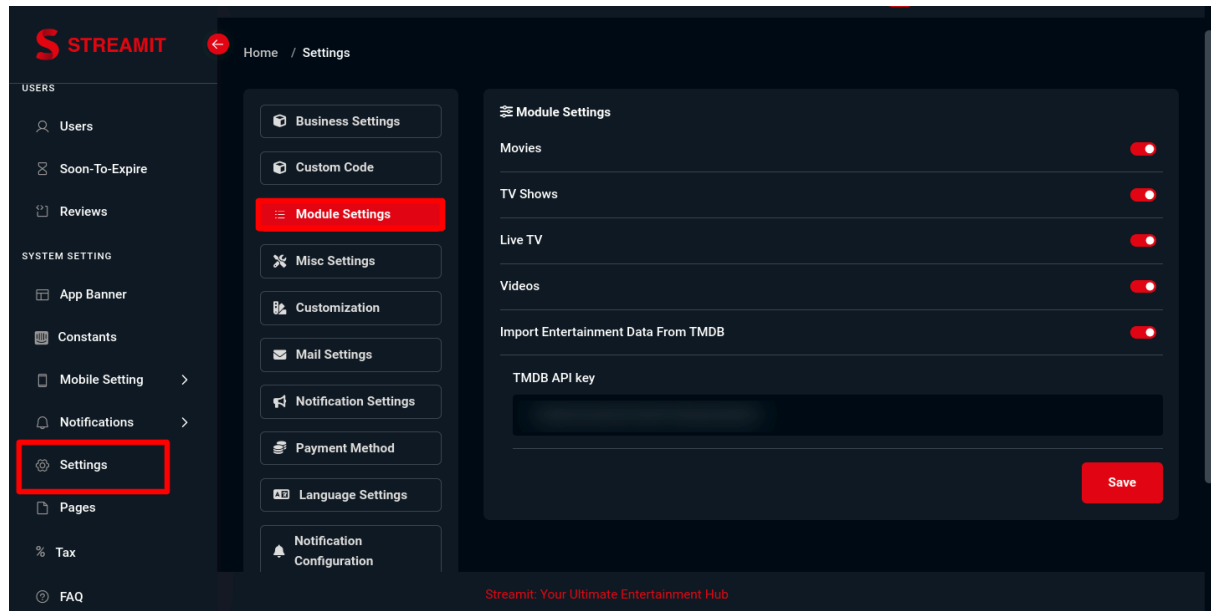
Business Setting - Change your App Name, Footer Text, Copyright Text, Helpline Number, Inquiry Email, Site Description and Logos.



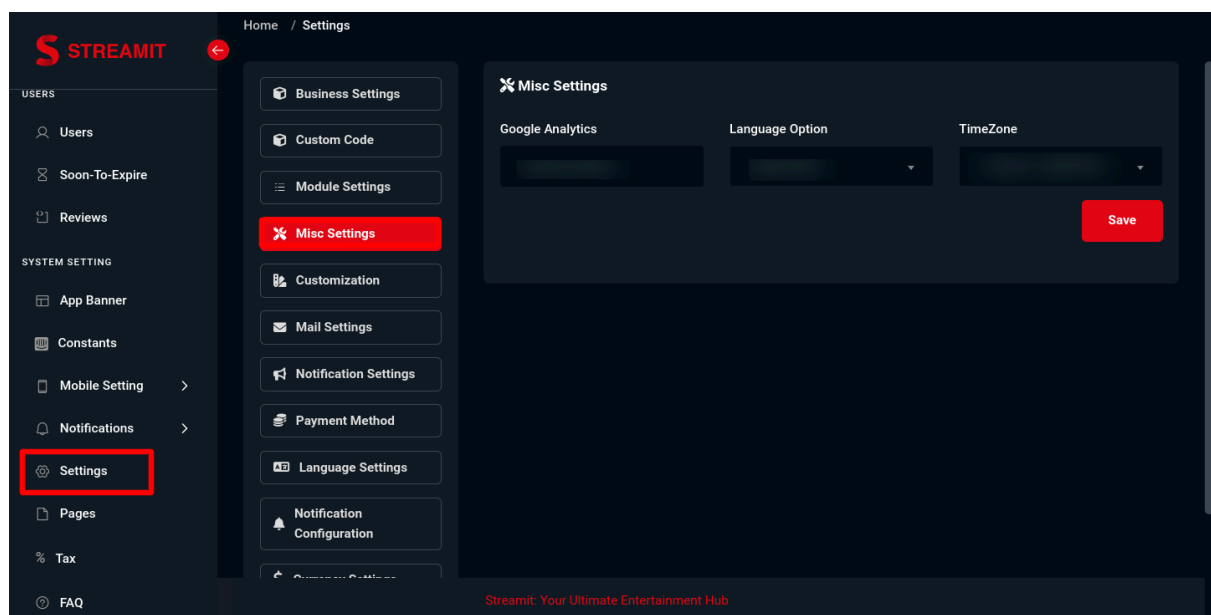
Custom Code- The Custom Code section allows you to add custom programming code to your app or platform. This is useful when you need additional functionality or want to modify how certain parts of the platform behave.



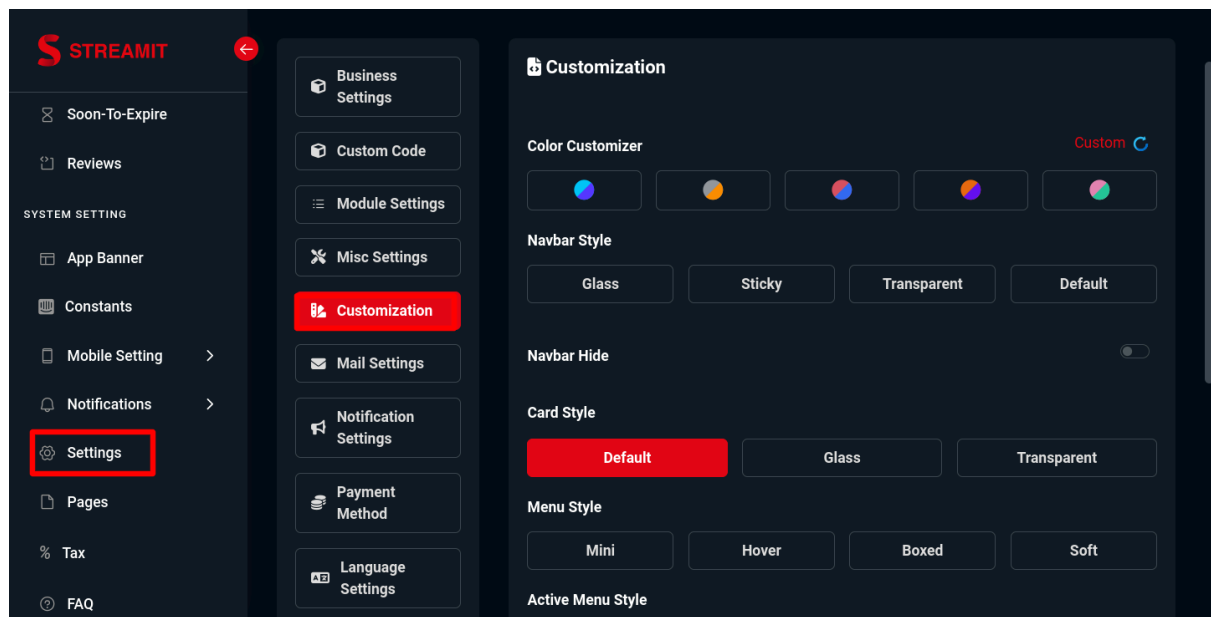
Module Setting - This section lets you manage and configure the different modules or features available in your app or system. Like Movies, TV Shows, Live TV, Videos. You can set module settings



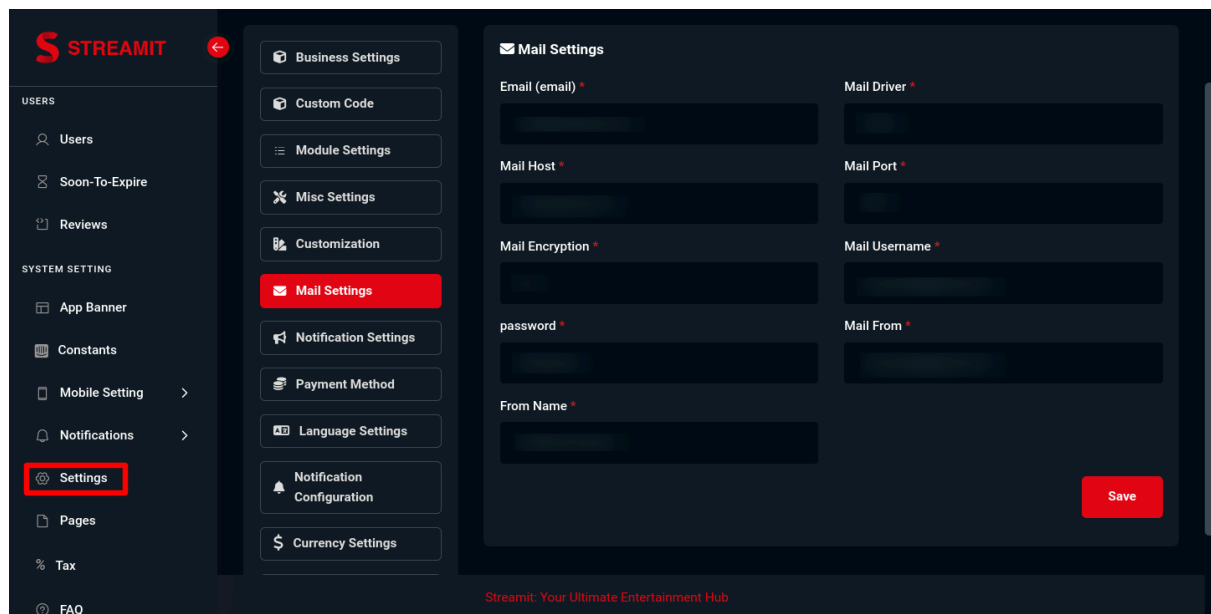
Misc Setting- This is Global Settings.



Customization- This is for Customizing look & feel of Theme.



Mail Setting- Configure Mail Settings.



Notification Setting- Configure when you want send notifications.

Streamit

Users

Soon-To-Expire

Reviews

System Setting

App Banner

Constants

Mobile Setting

Notifications

Settings

Pages

Tax

FAQ

Business Settings

Custom Code

Module Settings

Misc Settings

Customization

Mail Settings

Notification Settings

Payment Method

Language Settings

Notification Configuration

Currency Settings

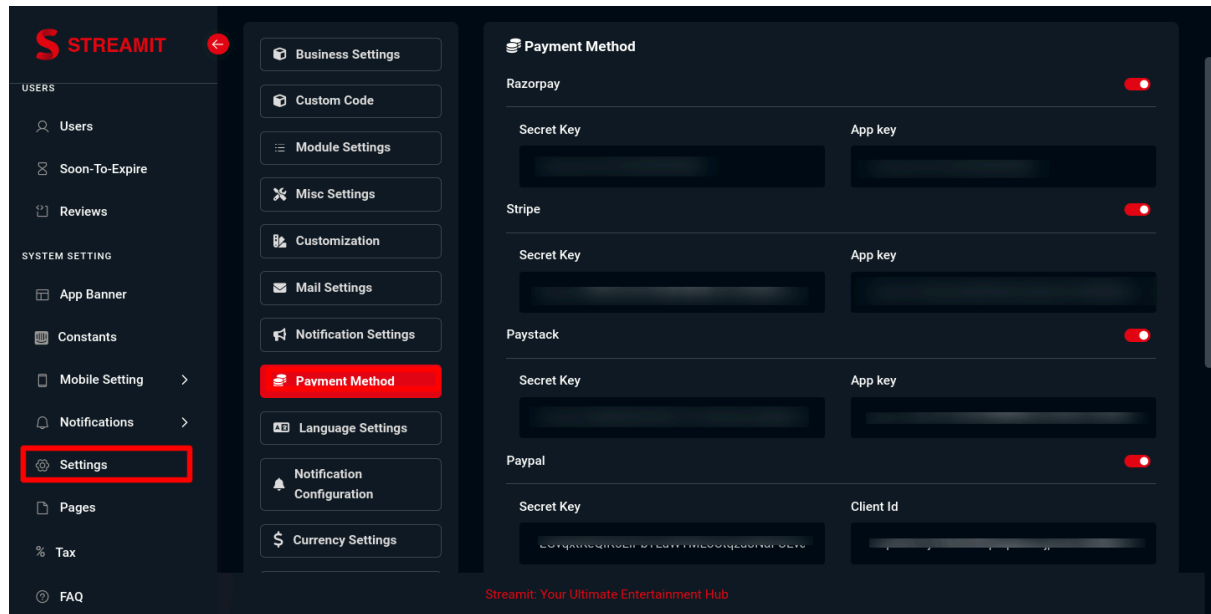
Notification Settings

Type	Templates	Mail	Mobile	Custom Webhook
change_password	Change Password	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
forget_email_password	Forget Email/Password	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
tv_show_add	TV Show Added!	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
movie_add	Movie Added!	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
episode_add	Episode Added!	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
season_add	Season Added!	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
new_subscription	New User is subscribe!	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
cancle_subscription	A User is cancle subscribe!	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

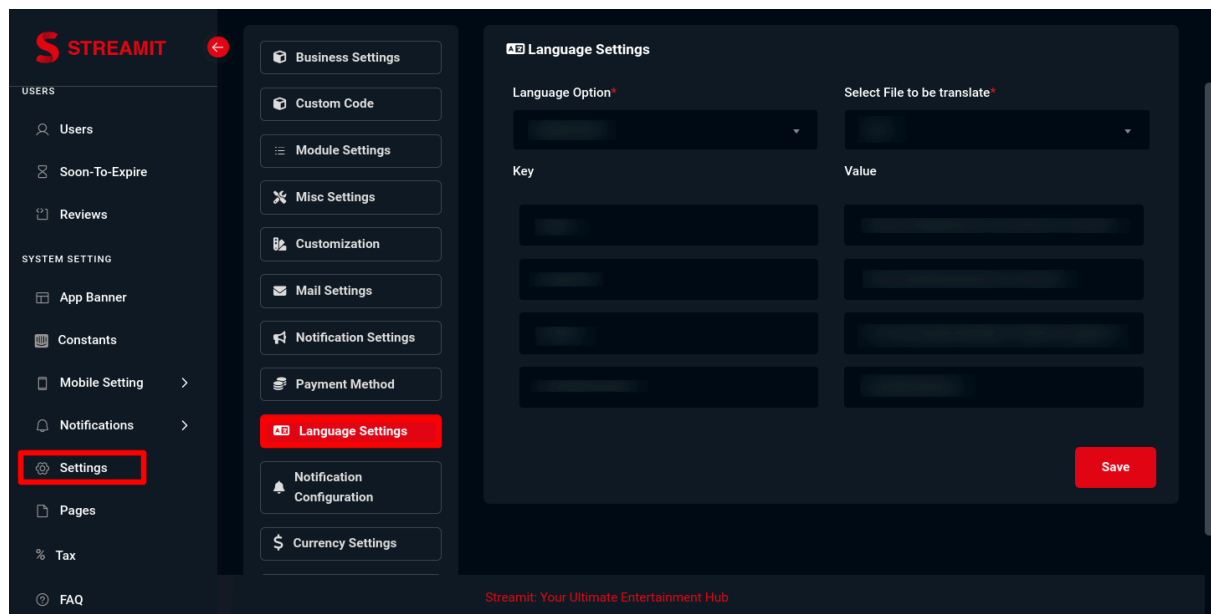
Save

Streamit. Your Ultimate Entertainment Hub

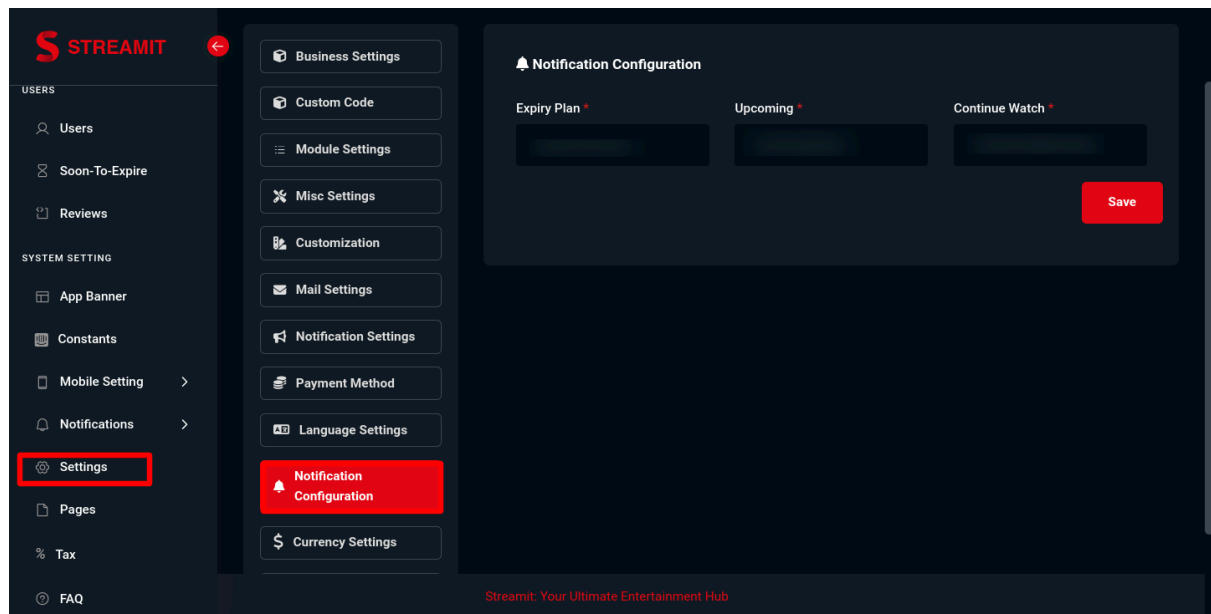
Payment Method- Enable/Disable any Payment Method from here.



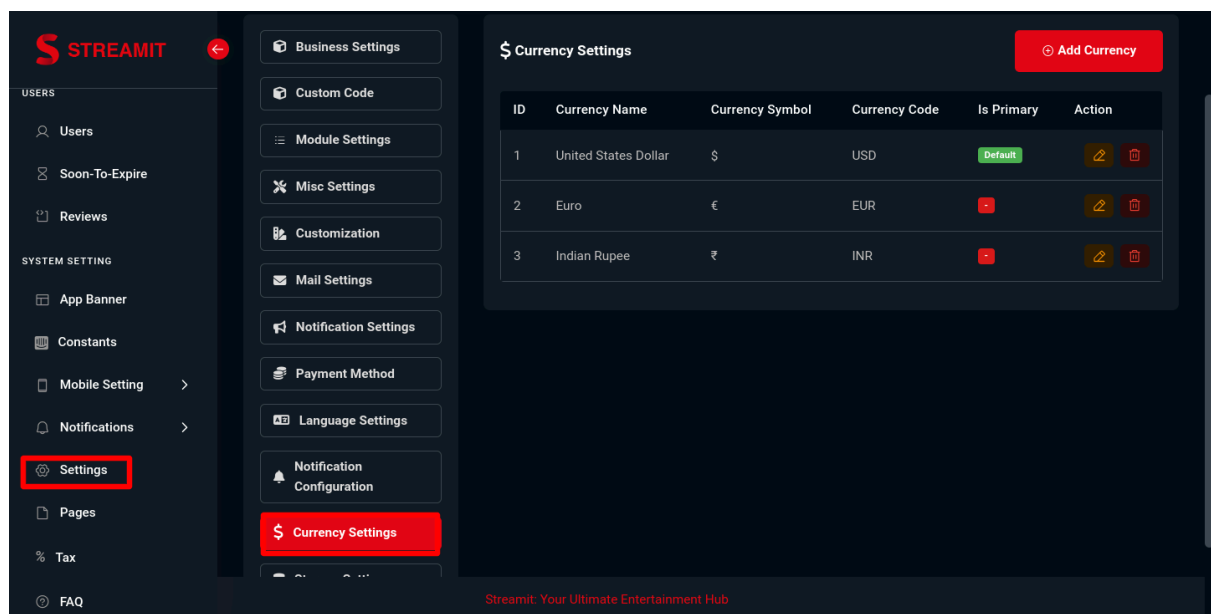
Language Setting- If you want to change any translation than utilize this settings.



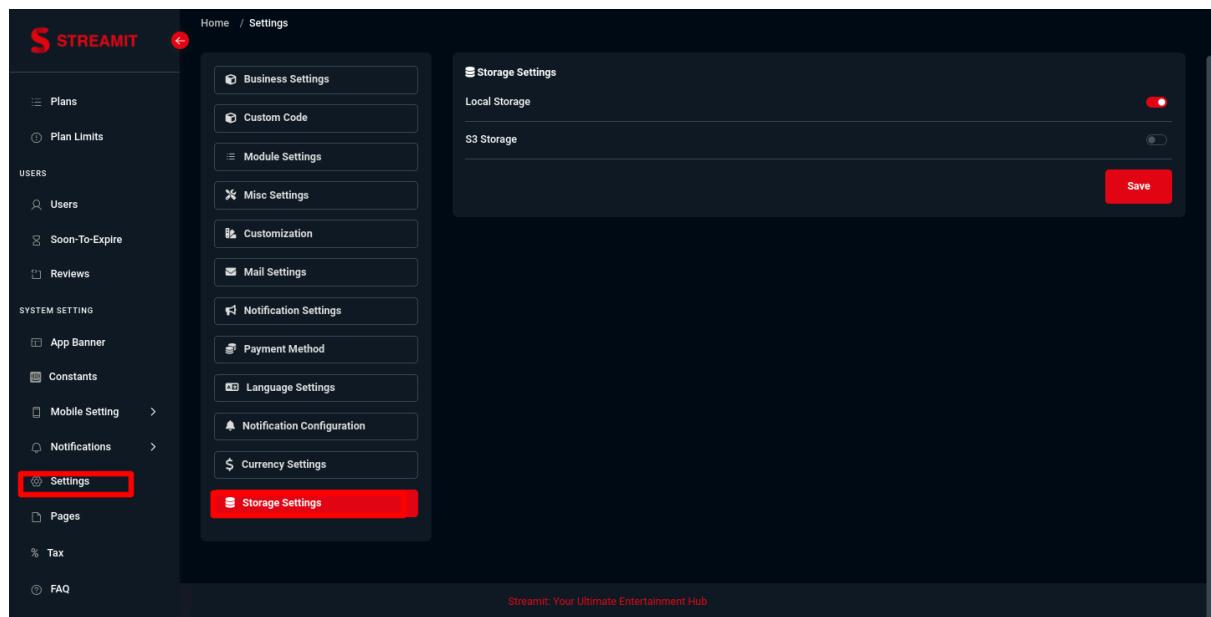
Notification Configuration- This setting allows you to manage how and when notifications are sent to users.



Currency Setting- You can set Global Currency settings.



Storage Setting- You can set Storage settings.



6. Payment Methods

Razorpay Gateway

Welcome to Razorpay! In this guide, you will learn how to sign up for a Razorpay account and obtain API keys necessary for integrating Razorpay into your website or application.

Razorpay Payment

Razorpay is the only payments solution in India that allows businesses to accept, process, and disburse payments with its product suite. It gives you access to all payment modes, including credit card, debit card, net banking, UPI, and popular wallets such as JioMoney, Mobikwik, Airtel Money, FreeCharge, Ola Money, and PayZapp. Manage your marketplace, automate NEFT/RTGS/IMPS bank transfers, collect recurring payments, and share invoices with customers – all from a single platform. Fast forward your business with Razorpay.

[Click here](#) for more details about the Razorpay payment gateway.

If you have an existing Razorpay Developer Account, [Log in](#) to your account.

Otherwise, [Sign up](#) and create a new business account.

Obtain API Keys

1. Click the link to [login](#).
2. After logging in, click on the **Settings** option. We've included a screenshot below. Then, select the **API keys** section.
3. Click on the **regenerate test key** option, and a dialog will open, displaying the **Key Id** and **Key Secret**.

If you close the **TEST MODE** switch, you will get the Live **Key Id** and **Key Secret**.

Add Generated Keys To Admin Panel

Now that you've successfully generated the keys, navigate to the Admin Panel and follow these steps:

- Navigate to **Settings** → **Payment Method**.
- Enable the Razorpay Payment switch.
- Paste the **Secret Key** under the *Secret Key* field and the **App Key** under the *App Key* field.
- Save the changes.

Congratulations! Your Razorpay configuration is complete.

Test Razorpay

Here, we provide some testing credentials for the app.

If you wish to add your own payment gateway, consider purchasing our paid support for the same.

Use Only in Test Mode

You can use these test cards to make payments in test mode only. Using them in live mode will result in a "card issuer is invalid" or "invalid card input" error.

Test Cards Details for Razorpay Payment

You can use the following test card in test mode to ensure the integration is working correctly. Use any CVC, postal code, and future expiration date.

Card Number	Expiry Date	CVV	Cardholder Name
4111 1111 1111 1111	12/25	123	Test User

PayStack Gateway

This guide walks you through signing up for a PayStack account and obtaining the necessary API keys to integrate PayStack payments into your website or application.

What is PayStack?

PayStack is a leading payment processing platform enabling businesses in Africa to accept payments globally.

Why Choose PayStack?

- 1) Registered and unregistered businesses can leverage PayStack's services.
- 2) Registered businesses enjoy full access to features, while unregistered businesses (Starter Businesses) have limited access.
- 3) See requirements for:

Registered Businesses: <https://support.paystack.com/en/articles/2128898>

Starter Businesses: <https://support.paystack.com/en/articles/2128898>

Important Note:

Before proceeding, ensure your currency is supported by PayStack:

<https://support.paystack.com/en/articles/2130690>

Sign Up/Log In

- **Existing Account:** Log in to your PayStack Developer Account: <https://dashboard.paystack.com/>
- **New Account:** Sign up for a new business account: <https://dashboard.paystack.com/>

Sign Up Process:

1. Provide your business name, email, country, and password: <https://dashboard.paystack.com/>
2. Complete the signup process by providing additional information.
3. Once complete, you'll be redirected to your dashboard.

Testing Integration:

- Start testing your integration and explore features in Test Mode.

Important Note:

- Your account is initially in Test Mode and cannot receive real money.

- To receive real payments, activate your account: <https://dashboard.paystack.com/>

Obtaining API Keys

Obtain Test API Keys

1. Navigate to **Dashboard** → **Settings** <https://paystack.com/developers>
2. Click on **API Keys & Webhooks** tab.
3. Locate the **API Configuration - Test Mode** section.
4. Find your **Test Secret Key** and **Test Public Key**.

Obtaining Live API Keys

Follow the same steps above to obtain Live API Keys after completing your testing and integration.

Generating New Live Secret Key

1. Log in to your PayStack Developer Dashboard: <https://dashboard.paystack.com/>
2. Navigate to **Settings** <https://paystack.com/developers>
3. Click on **API Keys & Webhooks**.
4. Under your Live and Test Secret Keys, click on **Generate new secret key**.

Important Note:

Generating a new secret key will require updating it on your website or server to process transactions.

Adding Generated Keys to Admin Panel

1. Navigate to **Settings** → **Payment Method** in your Admin Panel.
2. Enable the switch for **PayStack** Payment.
3. Paste the PayStack **Secret Key** under **Secret Key** and **App Key** under **App Key** fields.
4. Submit the changes.

Congratulations! Your PayStack configuration is complete.

Testing PayStack Payment

Test Credentials (for Test Mode Only):

Important Note:

Use these test cards for testing purposes only. Using them in Live Mode will result in errors.

Description	Card Number	Expiry Date (MM/YY)	CVV	IN	OTP	Third Validation
No validation (reusable)	084 0840 8408 081	12/25	08	111		
IN validation	078 5078 5078 078	12/25	1	111		
IN + OTP validation	060 6666 6666 666	12/25	23	234	23456	Validation Data3

FlutterWave Gateway

This guide explains how to sign up for a FlutterWave account and obtain the necessary API keys to integrate FlutterWave payments into your website or application.

Sign Up for FlutterWave

1. **Visit the FlutterWave Website:** Go to <https://app.flutterwave.com/register>.
2. **Sign Up:** Click the "Sign Up" button in the top right corner.
3. **Provide Your Details:** Fill out the registration form with your email address, password, and other required information.
4. **Verify Your Email:** Check your inbox for a verification message from FlutterWave and click the link to verify your account.
5. **Complete Your Profile:** Once verified, log in and complete your profile details (business name, address, contact details).

Congratulations! You've created a FlutterWave account. Now, let's get your API keys.

Obtain API Keys

FlutterWave provides API keys for integrating their payment gateway. There are two account modes:

- **Live Mode:** Processes real transactions with real money.
- **Test Mode:** Uses test cards and bank accounts for development and testing (no real money involved).

Get an API Key for the First Time

1. **Log in:** Access your FlutterWave dashboard (<https://dashboard.flutterwave.com/login>).
2. **Settings:** Click "Settings" and then "API Keys" under the Developers tab.
3. **Generate Secret Key:** Click "Generate Secret Key" to obtain your live Secret Key. An authentication code will be sent to your email.
4. **Enter Authentication Code:** Check your email for the 7-digit code and enter it to complete the generation process.

Generate New API Keys

If you already have API keys and need new ones:

1. **Settings:** Log in to your dashboard and navigate to "Settings" -> "API Keys".
2. **Generate New Keys:** Click "Generate New Key" to obtain new live API keys.
3. **Set Expiration (Optional):** Set an expiration period for your current keys (optional).
4. **Enter Authentication Code:** Enter the code sent to your email and click "Generate new API keys".

There are three types of API keys:

- **Secret Key:** The most powerful key, authorizing any action on your account. Keep it private.
- **Public Key:** Used in "public" scenarios.
- **Encryption Key:** Used with the direct card charge endpoint (refer to encryption guide).

Add Generated Keys to Admin Panel

1. Navigate to **Settings** → **Payment Method** in your Admin Panel.
2. Enable the switch for **FlutterWave** Payment.
3. Paste the **Secret Key** under **Secret Key** and **Public Key** under **App Key**.
4. Submit the changes.

Congratulations! Your FlutterWave configuration is complete.

Test FlutterWave Payment

Use the following test card details for testing purposes (only in Test Mode)

Update Guide

Here are the steps to update the Streamit project.

1. Download the updated zip file and extract it.
2. Go to the main directory of the Streamit project.
3. Run the following command:
 - `php artisan migrate`

This command will migrate the database and update any necessary changes.

FAQs

Frequently Asked Questions

1. Symlink() has been disabled for security reasons?

When you get an error while run `php artisan storage:link`, please follow the below steps:

Second way, just fire this command

```
ln -s /home/your_application/storage/app/public /home/your_application/public/storage
```

2. Migrate database issue?

If you are facing a problem migrating database using command the please follow the below steps.

Go to the main dir Streamit-Laravel

then find Streamit-Laravel.sql file and import to the database.

3. Forgot password swift_transport exception?

If you are facing any swift transport type issue while sending mail,

then, please set the mail config in your `.env` file

please follow the below step.

`MAIL_DRIVER=your mail driver`

`MAIL_HOST=your mail host`

`MAIL_PORT=your mail port`

`MAIL_USERNAME=your mail id`

`MAIL_PASSWORD=your mail password`

`MAIL_ENCRYPTION=your mail encryption 1`

Help & Support

Iqonic Design is happy to assist you with any questions or concerns you have about our products.

Support Channels:

- **Help Desk:** Submit a ticket through our dedicated support portal at [Desky](#).
- **Email:** Contact us directly via email at hello@iqonic.design.

Support Policy:

Before contacting support, please review our documentation. Many common issues can be resolved by following our guides. To receive technical support, you must have a valid purchase code obtained from your Themeforest "Download" page (<https://codecanyon.net/>).

Need Assistance?

If your questions extend beyond the documentation, submit a ticket through our support portal (<https://iqonic.desky.support/>). We aim to respond within 24-48 hours, prioritizing tickets by order of receipt.

Important Reminders:

- Search our Knowledge Base for answers to common questions before opening a ticket.
- Pre-sales inquiries and general concerns can be directed to CodeCanyon ([invalid URL removed]) or hello@iqonic.design.
- We appreciate positive reviews on CodeCanyon.

Notes:

- Technical support covers theme-related features only. Customization requests are not included.
- Support excludes third-party plugin integration or compatibility issues beyond our development scope.
- Our support team operates during business hours (Monday-Saturday, 9:00-18:00 GMT+05:30). Please consider our time zone when expecting responses.

We are here to help! Don't hesitate to reach out if you need assistance.

Source & Details

Tools

- [Gulp](#)
- [Node.js](#)

Images

- [Pexels](#)
- [Freepik](#)
- [Pixabay](#)

Scripts

- [jQuery](#)
- [Select2](#)
- [Quill](#)
- [Flatpickr](#)
- [Rater](#)
- [Swiper](#)
- [FS Lightbox](#)
- [Waypoints](#)
- [Sweetalert2](#)
- [Smooth Scrollbar](#)
- [Nouislider](#)
- [Counterup2](#)
- [Cropper.js](#)
- [Choice.js](#)
- [Popper.js](#)
- [Counter](#)
- [ApexCharts](#)

CSS & Fonts

- [Bootstrap](#)
- [Google Fonts](#)

