

Freshie Technical Documentation

Project Overview

Freshie is a modern e-commerce platform designed for selling clothing items. It offers a seamless user experience for browsing products, managing shopping carts, and placing orders. Additionally, it includes a robust admin panel powered by Filament for managing inventory, orders, and user accounts.

- **Website:** <https://freshlook.duckdns.org/>
- **GitHub Repository:** <https://github.com/xdkanashi/Freshie>
- **License:** MIT

Architecture and Technologies

Freshie is built as a monolithic application using the Laravel framework, following the Model-View-Controller (MVC) pattern. It integrates Filament for administrative tasks and Tailwind CSS for styling.

Backend

- **Laravel:** Version 12.7.2 (PHP framework).
- **PHP:** Version 8.2.28.
- **Livewire:** For dynamic, reactive frontend components.
- **Filament:** Version 3.2, for the admin panel.

Frontend

- **Tailwind CSS:** Utility-first CSS framework for responsive design.
- **Vite:** Asset bundler for compiling CSS and JavaScript.
- **Blade:** Laravel's templating engine.

Database

- **MySQL:** Relational database for storing application data.

Server

- **Nginx:** Web server for serving the application.
- **PHP-FPM:** Handles PHP requests.

Tools

- **Composer:** PHP dependency manager.
- **npm:** Frontend dependency manager.
- **Git:** Version control system.

Features

User-Facing Features

- **Product Catalog:** Browse clothing items with details (name, price, discount).
- **Shopping Cart:** Add, update, or remove items; view cart totals.
- **Checkout:** Place orders (requires user authentication).
- **User Authentication:** Register, log in, and manage profiles.
- **Responsive Design:** Optimized for all devices.

Admin Panel (Filament)

- **Dashboard:** Displays site metrics (e.g., site visits).
- **Resource Management:**
 - **Clothes:** Manage product inventory.
 - **Orders:** Track and update order statuses.
 - **Users:** Administer user accounts and roles.
 - **Coupons:** Create and manage discount codes.
 - **Contacts:** Handle customer inquiries.
- **Access Control:** Restricted to users with `is_admin = 1`.

Installation and Setup

Prerequisites

- PHP ≥ 8.2
- Composer
- Node.js ≥ 16
- npm
- MySQL ≥ 5.7
- Nginx or Apache
- Git

Setup Instructions

1. Clone the Repository:

```
git clone https://github.com/xdkanashi/Freshie.git
cd Freshie
```

2. Install PHP Dependencies:

```
composer install
```

3. Install Frontend Dependencies:

```
npm install
```

4. Configure Environment:

- Copy the example .env file:

```
cp .env.example .env
```

- Update .env with your database credentials:

```
APP_NAME=Freshie
APP_URL=https://yourdomain.com
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=freshie_db
DB_USERNAME=your_username
DB_PASSWORD=your_password
```

5. Generate Application Key:

```
php artisan key:generate
```

6. Run Migrations:

```
php artisan migrate
```

7. Seed Database (Optional):

```
php artisan db:seed
```

8. Build Frontend Assets:

```
npm run build
```

9. Set Permissions:

```
sudo chown -R www-data:www-data /var/www/Freshie/storage
sudo chown -R www-data:www-data /var/www/Freshie/bootstrap/cache
sudo chmod -R 775 /var/www/Freshie/storage
sudo chmod -R 775 /var/www/Freshie/bootstrap/cache
```

10. Configure Web Server (Nginx Example):

```
server {
    listen 80;
    server_name yourdomain.com;
    root /var/www/Freshie/public;

    index index.php index.html;

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

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php8.2-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

11. Restart Nginx:

```
sudo systemctl restart nginx
```

12. Create Admin User:

```
php artisan make:filament-user
```

13. Access the Application:

- Website: <https://yourdomain.com>
- Admin Panel: <https://yourdomain.com/admin>

Local Environment Setup (After Building Frontend Assets)

1. Set File Permissions:

```
chmod -R 775 storage bootstrap/cache  
chown -R $(whoami):$(whoami) storage bootstrap/cache
```

2. Start Local Server:

```
php artisan serve
```

Access at <http://127.0.0.1:8000>.

3. Create Admin User:

```
php artisan make:filament-user
```

4. Access:

- Website: <http://127.0.0.1:8000>
- Admin Panel: <http://127.0.0.1:8000/admin>

Database Schema

The database freshie consists of the following tables:

cache:

- **key**: Primary key (varchar, not null).
- **value**: Cache value (mediumtext, not null).
- **expiration**: Expiration timestamp (int, not null).

cache_locks:

- **key**: Primary key (varchar, not null).
- **owner**: Lock owner (varchar, not null).
- **expiration**: Expiration timestamp (int, not null).

cart:

- **id:** Primary key (bigint, auto-increment).
- **user_id:** Foreign key (users.id, on delete cascade).
- **clothe_id:** Foreign key (clothes.id, on delete cascade).
- **size:** Item size (varchar, not null).
- **quantity:** Item quantity (int, default 1).
- **created_at, updated_at:** Timestamps.

clothes:

- **id:** Primary key (bigint, auto-increment).
- **name:** Product name (varchar, not null).
- **category:** Product category (varchar, not null).
- **description:** Product description (text, nullable).
- **price:** Price (decimal(8,2), not null).
- **is_sold_out:** Sold out flag (tinyint, default 0).
- **pre_order:** Pre-order flag (tinyint, default 0).
- **discount:** Discount percentage (decimal(5,2), nullable).
- **size:** Available sizes (varchar, nullable).
- **created_at, updated_at:** Timestamps.

clothes_images:

- **id:** Primary key (bigint, auto-increment).
- **clothes_id:** Foreign key (clothes.id, on delete cascade).
- **image:** Image path (varchar, nullable).
- **created_at, updated_at:** Timestamps.

contacts:

- **id:** Primary key (bigint, auto-increment).
- **name:** Contact name (varchar, not null).
- **phone:** Contact phone (varchar, nullable).
- **subject:** Inquiry subject (varchar, not null).
- **email:** Contact email (varchar, not null).
- **comment:** Message (text, nullable).

- **created_at, updated_at:** Timestamps.

coupons:

- **id:** Primary key (bigint, auto-increment).
- **coupon_code:** Unique coupon code (varchar, not null).
- **coupon_type:** Type (enum: 'percentage', 'fixed').
- **discount_percentage:** Discount percentage (decimal(5,2), nullable).
- **delivery_discount_amount:** Delivery discount (decimal(10,2), nullable).
- **free_item_id:** Free item ID (bigint, nullable).
- **is_active:** Active flag (tinyint, default 1).
- **valid_from, valid_until:** Validity dates (date, nullable).
- **min_order_amount:** Minimum order amount (decimal(10,2), nullable).
- **usage_limit:** Usage limit (int, nullable).
- **description:** Coupon description (text, nullable).
- **usage_count:** Usage count (int, default 0).
- **created_at, updated_at:** Timestamps.

failed_jobs:

- **id:** Primary key (bigint, auto-increment).
- **uuid:** Unique identifier (varchar, not null).
- **connection, queue:** Job connection and queue (text, not null).
- **payload, exception:** Job payload and exception (longtext, not null).
- **failed_at:** Failure timestamp (timestamp, not null).

jobs:

- **id:** Primary key (bigint, auto-increment).
- **queue:** Queue name (varchar, not null).
- **payload:** Job payload (longtext, not null).
- **attempts:** Attempt count (tinyint, not null).
- **reserved_at, available_at, created_at:** Timestamps (int, nullable/required).

job_batches:

- **id:** Primary key (varchar, not null).
- **name:** Batch name (varchar, not null).
- **total_jobs, pending_jobs, failed_jobs:** Job counts (int, not null).
- **failed_job_ids:** Failed job IDs (longtext, not null).
- **options:** Batch options (mediumtext, nullable).
- **cancelled_at, created_at, finished_at:** Timestamps (int, nullable/required).

leaderboard:

- **id:** Primary key (bigint, auto-increment).
- **user_id:** Foreign key (users.id, unique, on delete cascade).
- **score:** User score (int, not null).
- **created_at, updated_at:** Timestamps.

migrations:

- **id:** Primary key (int, auto-increment).
- **migration:** Migration name (varchar, not null).
- **batch:** Batch number (int, not null).

orders:

- **id:** Primary key (bigint, auto-increment).
- **order_number:** Unique order number (varchar, not null).
- **customer_name:** Customer name (varchar, not null).
- **status:** Order status (enum: 'pending', 'processing', 'completed', 'canceled').
- **notes:** Order notes (text, nullable).
- **created_at, updated_at:** Timestamps.

password_reset_tokens:

- **email:** Primary key (varchar, not null).
- **token:** Reset token (varchar, not null).
- **created_at:** Timestamp (nullable).

sessions:

- **id:** Primary key (varchar, not null).
- **user_id:** Foreign key (users.id, nullable).
- **ip_address:** User IP (varchar, nullable).
- **user_agent:** Browser agent (text, nullable).
- **payload:** Session data (longtext, not null).
- **last_activity:** Last activity timestamp (int, not null).

users:

- **id:** Primary key (bigint, auto-increment).
- **username:** Unique username (varchar, not null).
- **email:** Unique email (varchar, not null).
- **password:** Hashed password (varchar, not null).
- **is_admin:** Admin flag (tinyint, default 0).
- **game_limit_enabled:** Game limit flag (tinyint, default 1).
- **last_game_played_at, next_game_time:** Game timestamps (timestamp, nullable).
- **game_interval_hours:** Game interval (int, default 24).
- **remember_token:** Remember token (varchar, nullable).
- **created_at, updated_at:** Timestamps.

File Structure

```
Freshie/
```

```
|— app/
```

```
|   |— Filament/
```

```
|   |   |— Resources/
```

```
|   |   |   |— ClotheResource.php
```

```
|   |   |   |— ContactResource.php
```

```
|   |   |   |— CouponResource.php
```

```
|   |   |   |— OrderResource.php
```

```
|   |   |   |— UserResource.php
```

```
|   |   |— Widgets/
```

```
|   |   |   |— SiteVisitsWidget.php
```

```
| | — Http/
| | | — Controllers/
| | | — Middleware/
| | | | — CheckIfAdmin.php
| | — Models/
| | | — User.php
| | | — Clothe.php
| | | — Cart.php
| | | — Contact.php
| | | — Coupon.php
| | | — Order.php
| — database/
| | — migrations/
| | — seeders/
| | | — DatabaseSeeder.php
| | | — UserSeeder.php
| | | — ClotheSeeder.php
| | | — CartSeeder.php
| | | — ContactSeeder.php
| | | — CouponSeeder.php
| | | — OrderSeeder.php
| — public/
| | — build/
| | | — assets/
| | | — manifest.json
| — resources/
| | — css/
| | | — filament/admin/theme.css
| | — views/
| | | — layouts/
| | | | — app.blade.php
```

```
| | | — welcome.blade.php
| — routes/
| | | — web.php
| — .env.example
| — composer.json
| — package.json
| — vite.config.js
| — README.md
```

Usage

User Interface

1. **Browse Products:** View clothing items on the homepage.
2. **Add to Cart:** Select size/quantity and add to cart.
3. **Checkout:** Log in, review cart, and place orders.
4. **Contact Support:** Submit inquiries via the contact form.
5. **Leaderboard:** View user rankings based on scores.

Admin Panel

1. **Access:** Log in at /admin with admin credentials.
2. **Manage Resources:**
 - **Clothes:** Add/edit products.
 - **Orders:** Update order statuses.
 - **Users:** Manage accounts.
 - **Coupons:** Create discount codes.
 - **Contacts:** Review inquiries.
3. **Dashboard:** View site metrics.

Contribution Guidelines

1. Fork the repository.
2. Create a branch: git checkout -b feature/your-feature.
3. Push: git push origin feature/your-feature.
4. Open a pull request.

Standards:

- Follow PSR-12 coding standards.
- Update documentation as needed.
- Add tests for new features.

Troubleshooting

- **Error 500:** Check storage/logs/laravel.log. Fix permissions:

```
sudo chown -R www-data:www-data storage bootstrap/cache
sudo chmod -R 775 storage bootstrap/cache
```

Filament Access Issues: Ensure `is_admin = 1` for the user.

- **Styles Missing:** Rebuild assets:

```
npm run build
```

Database Errors: Run migrations:

```
php artisan migrate
```

License

Licensed under the MIT License. [See for details.](#)

MIT License

Copyright (c) 2025 xdkanashi

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.