Report on

Trade Finance Guarantee Issuance System

-By

Apurva
Munish
Maharshi

Chidera

Contents

Trade Finance Guarantee Issuance System

This project involves developing a Trade Finance Guarantee Issuance System using Laravel. It focuses on creating and managing guarantees such as Bank, Bid Bond, Insurance, and Surety through both manual entry and bulk data transfer. The application integrates core CRUD functionalities, secure user access, and data processing, ensuring scalability and reliability.

Technologies Used

- **PHP**: Core programming language for backend development.
- **Laravel Framework**: PHP-based framework for building the system with MVC architecture.
- **MySQL**: Database management system for storing guarantees and related data.
- **HTML/CSS (Tailwind)**: For building responsive and visually appealing frontend interfaces.
- **JavaScript**: To enhance interactivity in the user interface.
- **MVC Pattern**: Used to organize the project into Model, View, and Controller layers, ensuring a clean architecture.
- **Repository Pattern**: For managing data access logic, providing modular and testable components.
- **Blade Templates**: For creating dynamic and reusable frontend views.

Database Schema Overview

The database consists of three main tables: **User**, **Guarantee**, and **File**, each designed to support secure operations and maintain data integrity.

1. **User Table**:
   This table manages user data for authentication and role-based access. It includes fields like id, name, email (unique), role (e.g., admin, applicant), password, and timestamps for tracking creation and updates.
2. **Guarantee Table**:
   This table stores guarantee-related information. Key fields include corporate_reference_number (unique), guarantee_type (e.g., Bank, Bid Bond), nominal_amount, currency, expiry_date,

applicant_id (linked to User.id), status, reviewed_by, and review_comments.

3. **File Table**:
   This table handles bulk-uploaded files stored as binary blobs. Fields include file_name, file_type, file_blob, and uploaded_by (linked to User.id).

Relationships

- **User-Guarantee**: Users can create or review multiple guarantees.
- **User-File**: Users can upload multiple files for bulk data processing.
- **Guarantee-File**: Files are indirectly linked to guarantees through users.

This schema ensures scalability, secure data handling, and seamless role-based interactions.

Features

1. **Core Operations**:
   - Create, review, apply for, issue, and delete guarantees. ○

     Validation rules ensure data integrity

2. **Authentication and Authorization**:
   - Laravel's **Auth middleware** is utilized to secure access to guarantee management functionalities.
3. **Bulk Data Processing**:
   - Supports uploading and processing of CSV, JSON, and XML files.

- Uploaded files are stored as blobs in the database. ○ File parsing includes validation for required fields, such as Corporate Reference Number and Expiry Date.

4. **File Management**:
    - A user interface lists uploaded files with metadata (e.g., name, upload date, type).
    - Files can be deleted with user confirmation to prevent accidental removals.
5. **Dynamic Frontend**:
    - Blade templates styled with **Tailwind CSS** to create interactive and user-friendly views

Deployment of Laravel

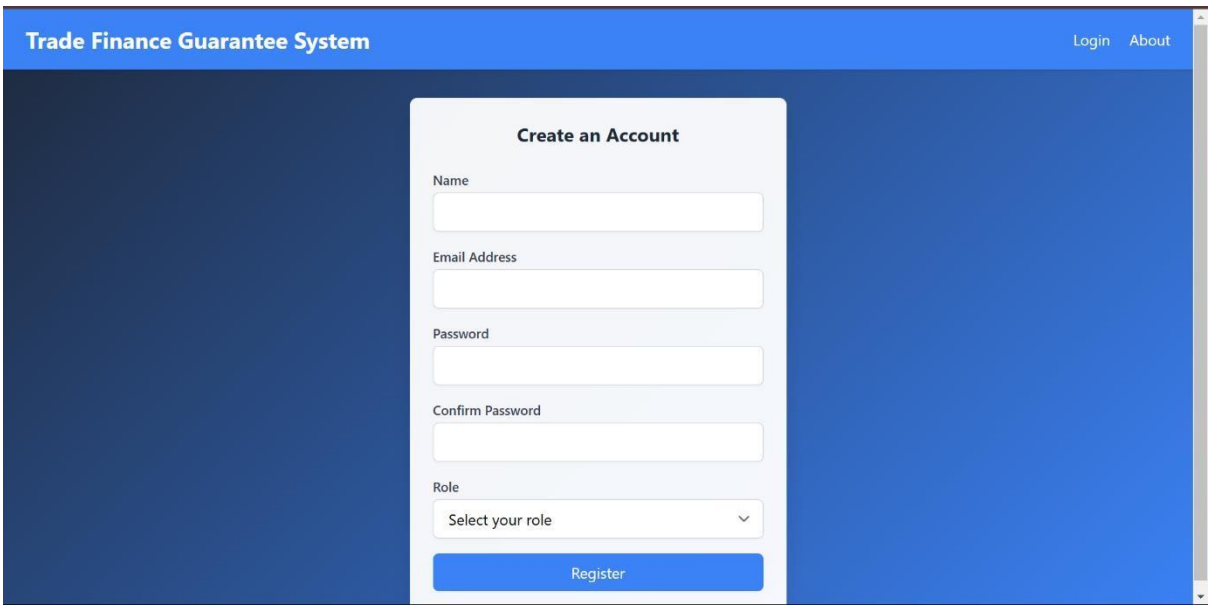
- **Install Dependencies**: ○ Run composer install to set up all required dependencies.
- **Configure the Environment**:
    - Copy .env.example to .env and update it with database credentials and other application settings.
    - Generate the application key with php artisan key:generate.
- **Database Setup**:
    - Create a new database schema in MySQL (or another supported database).

- Run php artisan migrate to set up tables and optionally php artisan db:seed to populate with sample data.

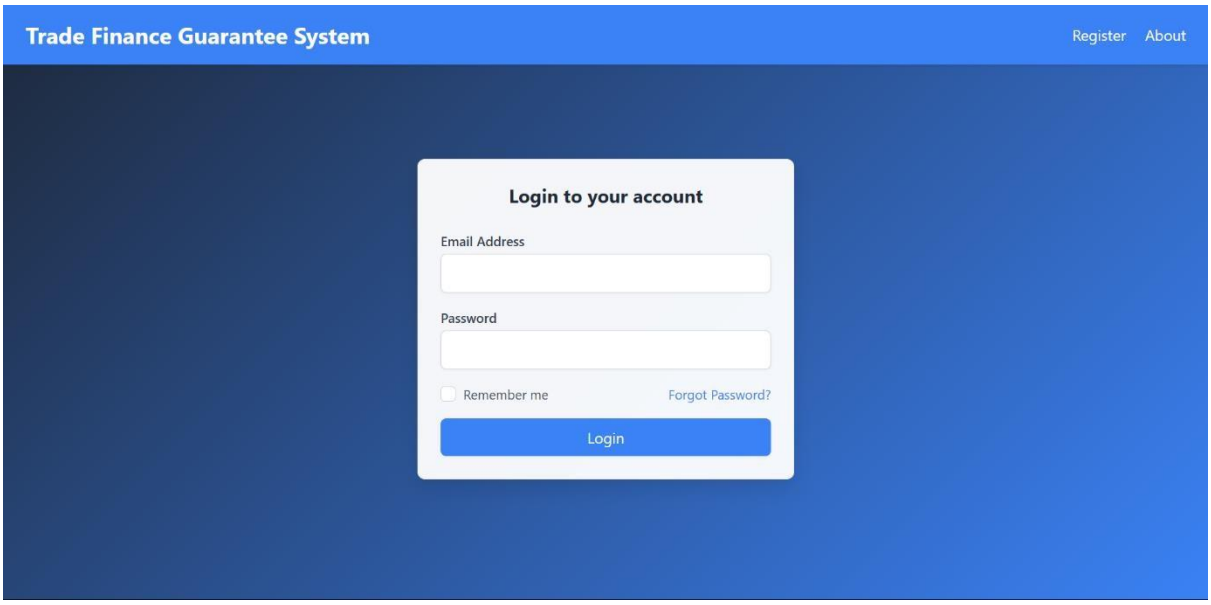
- **Run the Application**:
    - Start the development server using php artisan serve and access it at http://127.0.0.1:8000.

Screenshots

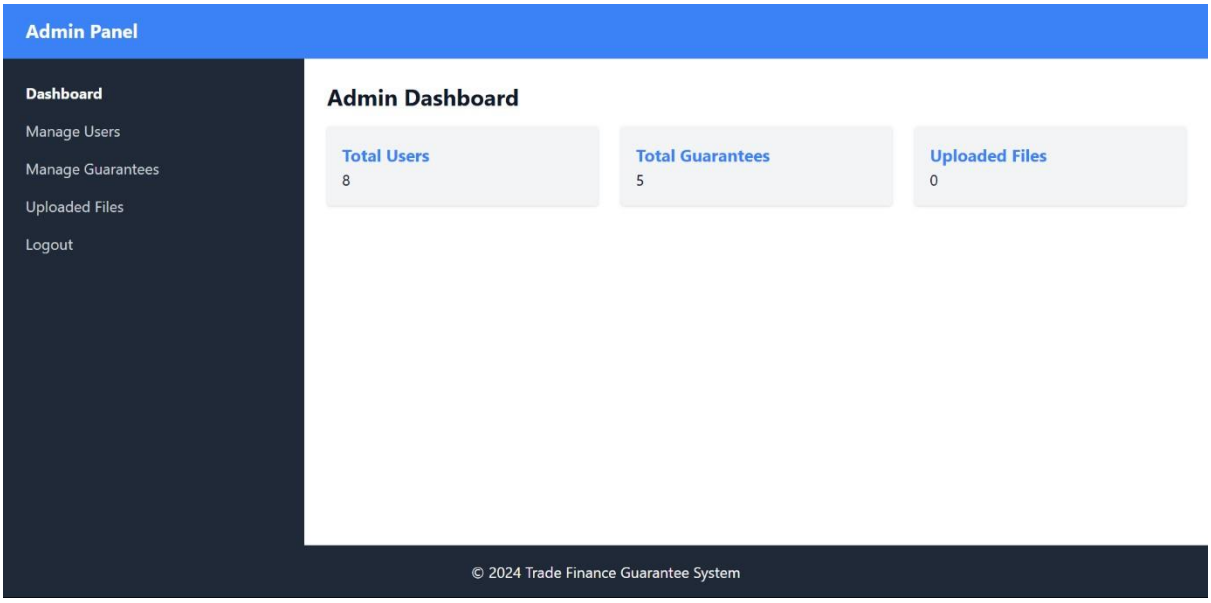
Register:



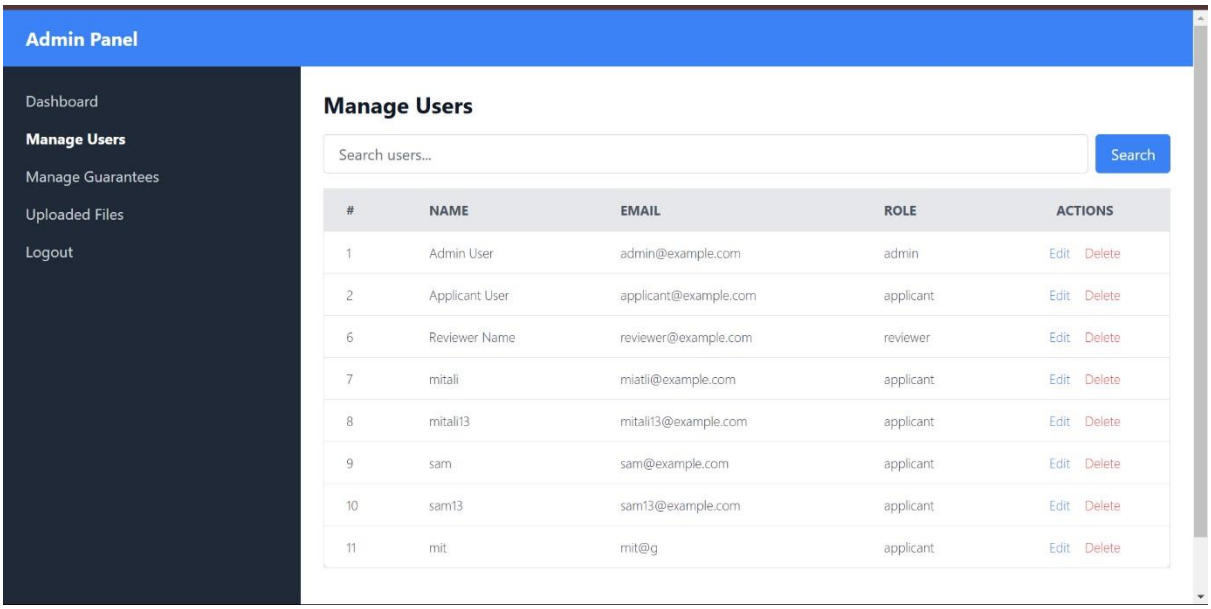
Login:



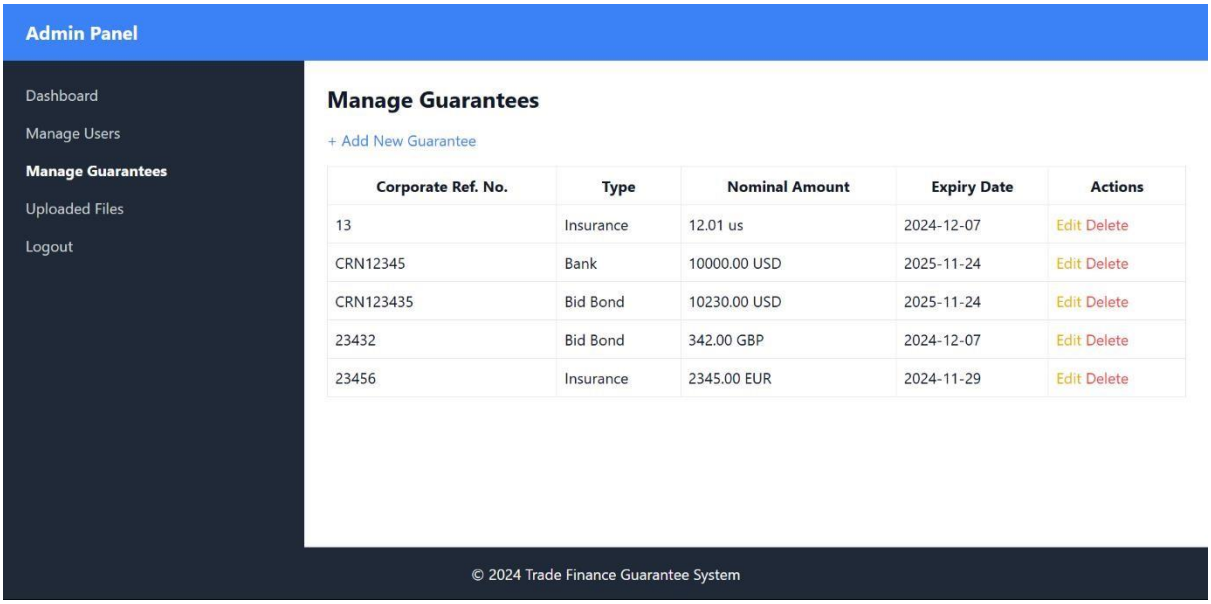
Admin Dashboard:



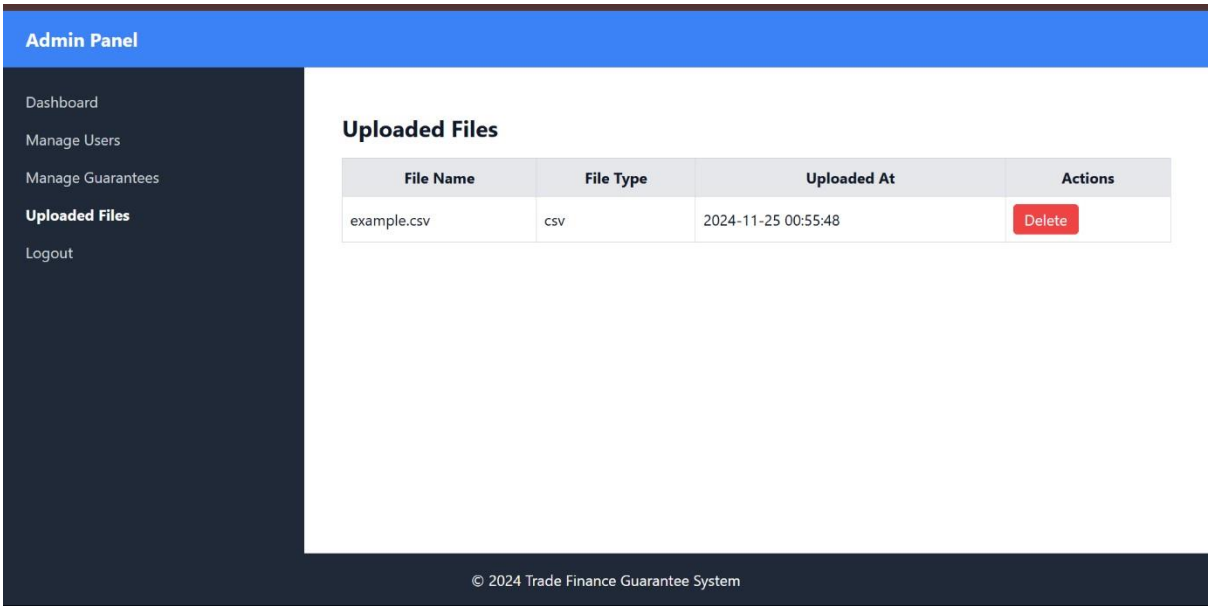
Manage Users:



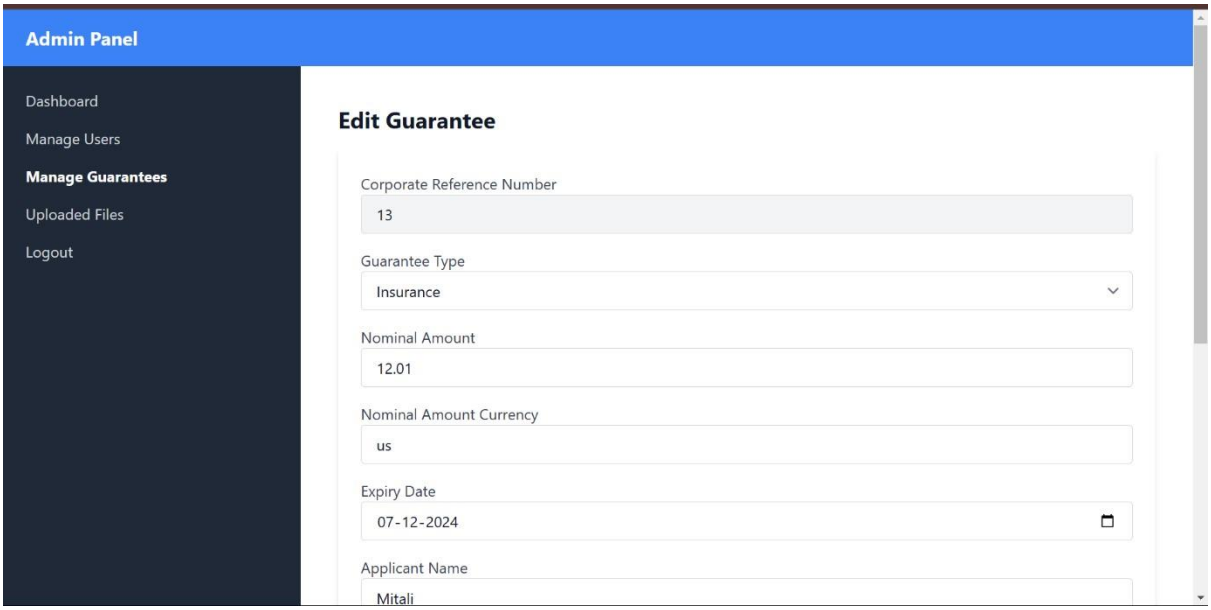
Manage Guarantees:



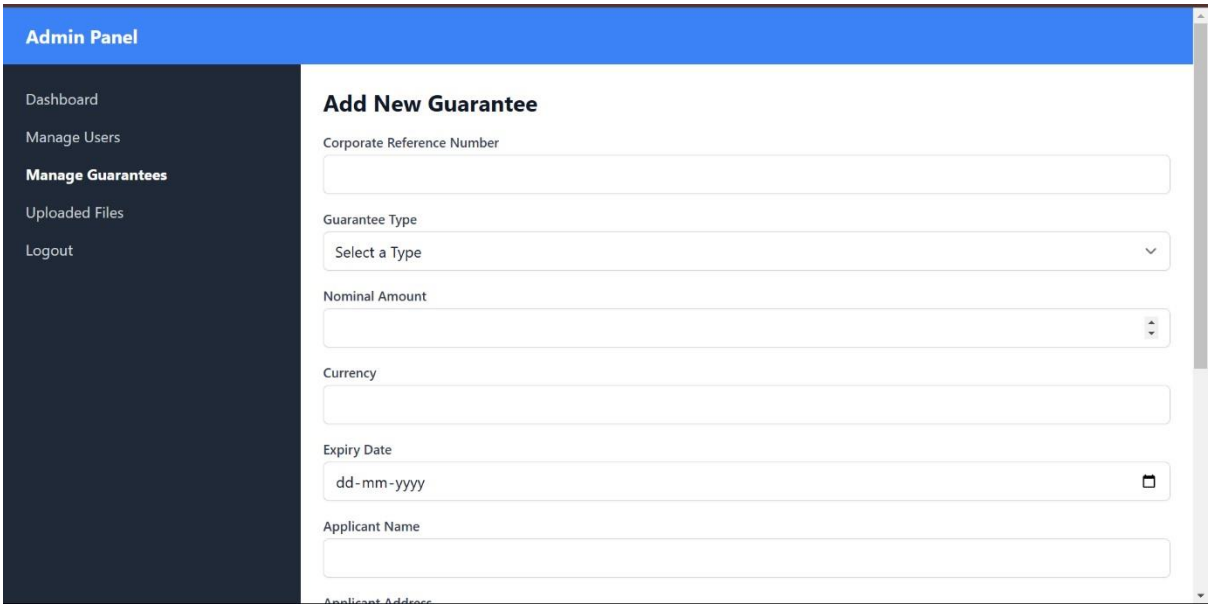
Upload data using File:



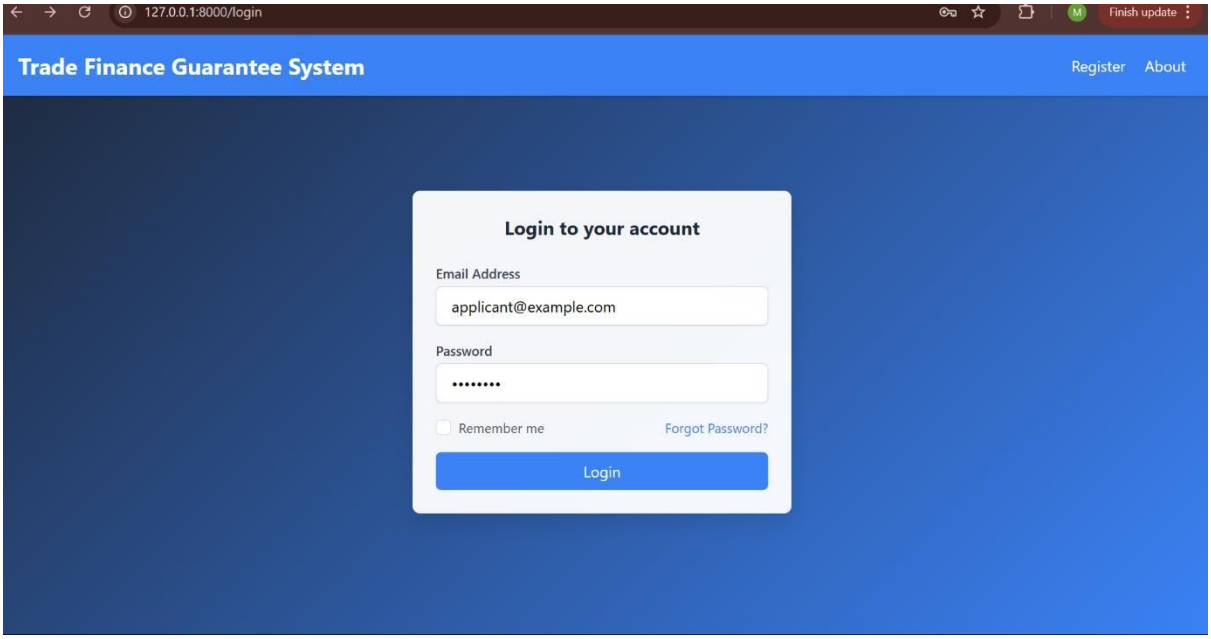
Edit Guarantee:



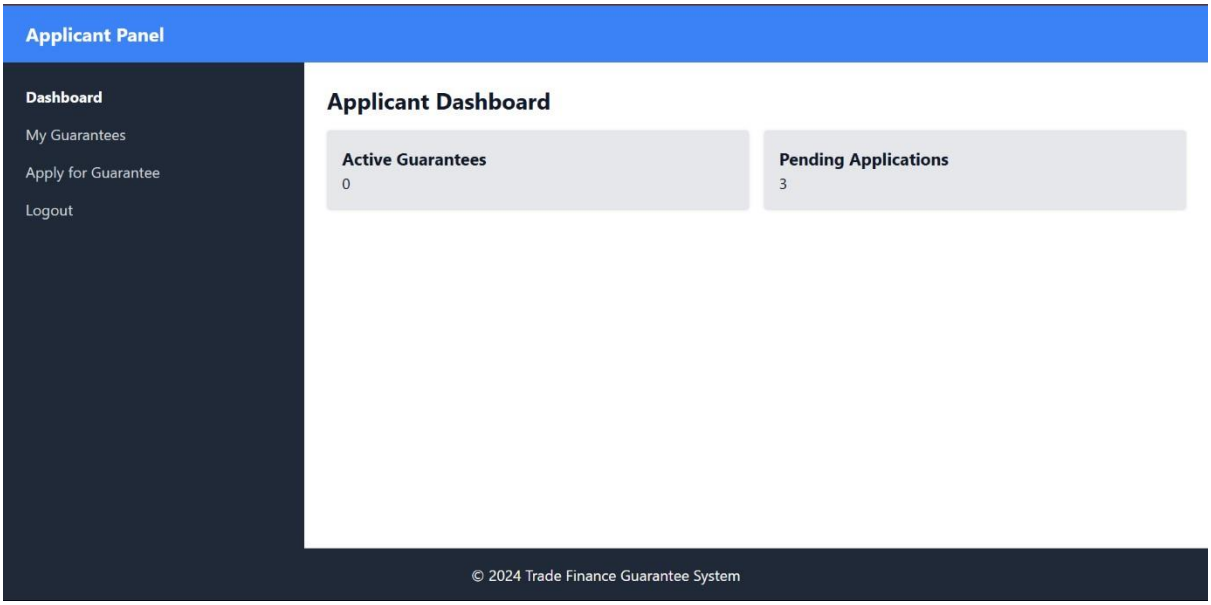
Add new Guarantee:



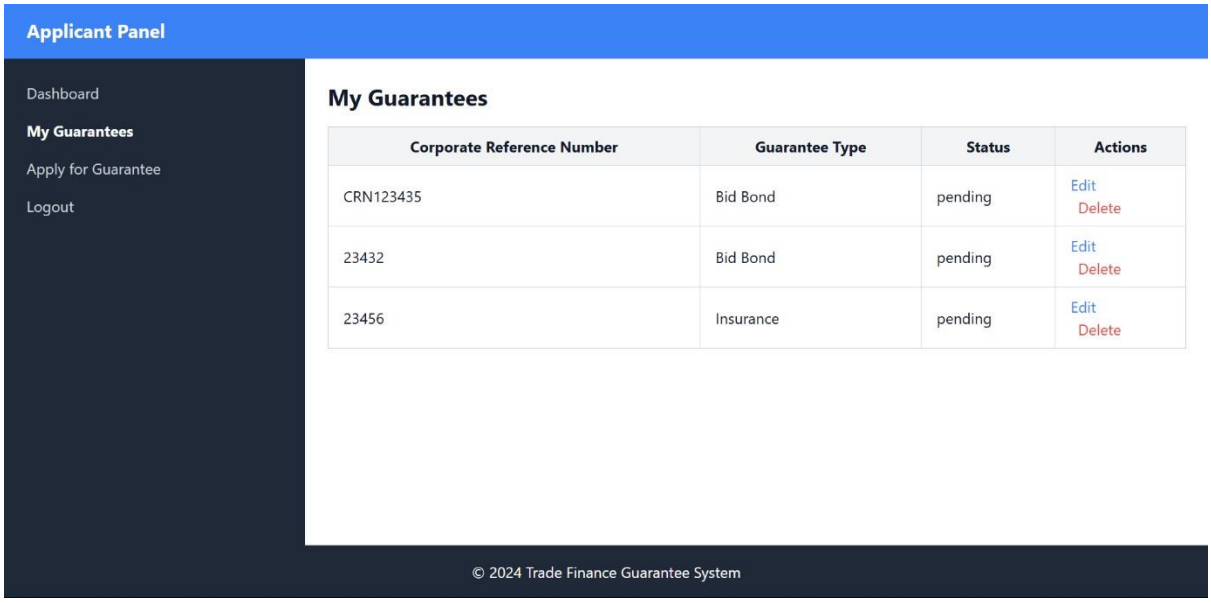
Applicant Login:



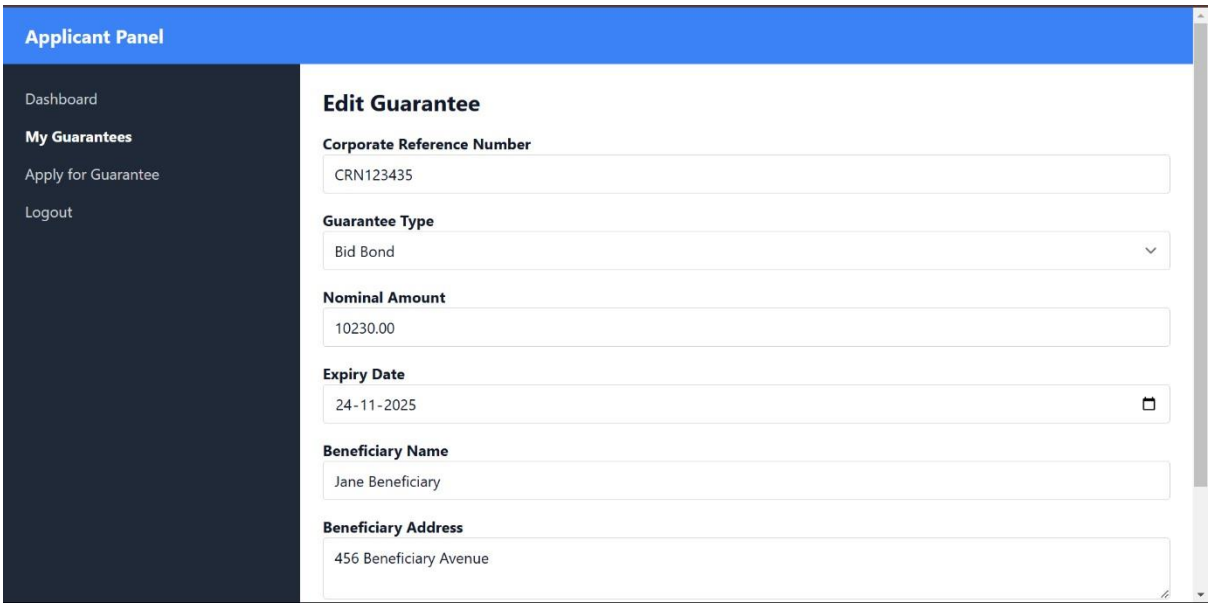
Applicant Dashboard:



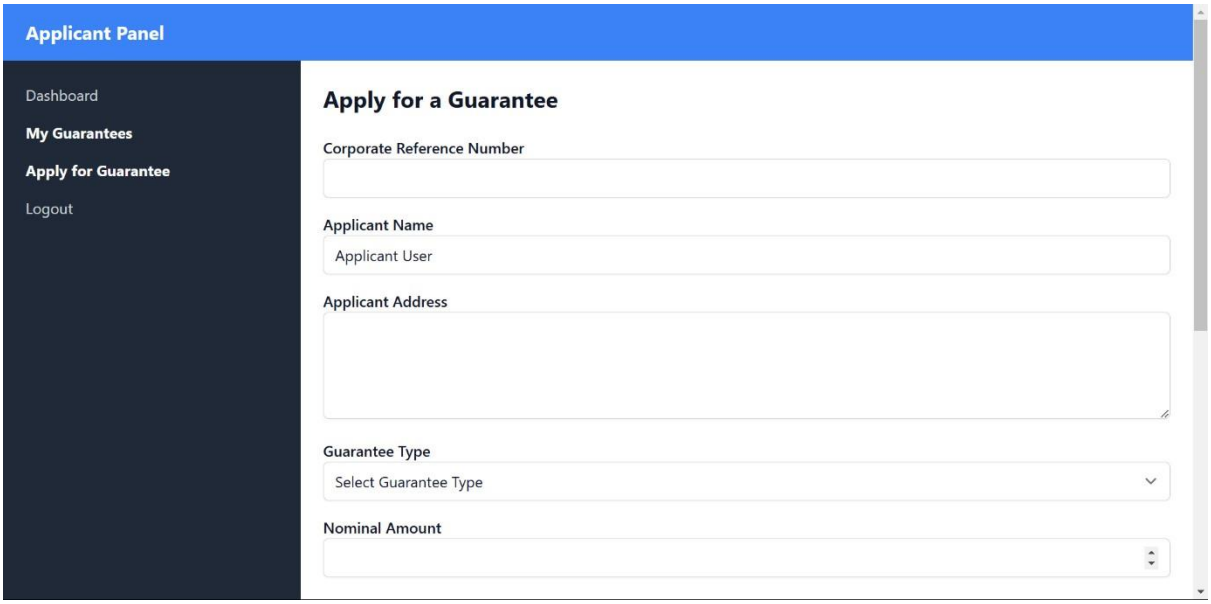
Submitted Guranatee:



Edit Guarantee:



Apply for new Guarantee:



Conclusion

The Laravel-based **Trade Finance Guarantee Issuance System** showcases an efficient and secure solution for managing guarantees with features like authentication, CRUD operations, and bulk data uploads. The use of modern technologies, clear architecture, and thorough local deployment ensures the application is reliable, functional, and ready for real-world use. This project underscores the value of organized development practices and robust testing in building scalable web applications.