

# EcoTrack - Waste Management System Documentation

---

## Author & Contributions

Developer: DABO R Yanis Axel (Y4NN), Y7 Solutions

**Role:** Full Stack Developer

As the sole developer of EcoTrack, I designed and built the entire system from the ground up. My key contributions include:

- Architecting and implementing the database with structured models and relationships
- Developing a secure authentication system with password hashing and CSRF protection
- Designing and coding user roles, including admin, operator, and reporting functionalities
- Enhancing UI/UX with pagination, dynamic interactions, and real-time updates
- Implementing data validation and security measures to protect user information
- Building an advanced reporting system with visual analytics and CSV exports

## Software Overview

### Purpose

EcoTrack is a web-based waste management system designed to track and manage waste collections efficiently. It enables organizations to monitor waste collection activities, schedule pickups, and generate comprehensive reports.

### Key Features

#### 1. User Management

- Three-tier role system:
  - **Administrator:** System management and user control
  - **Operator:** Collection and schedule handling
  - **Reporting User:** View and export reports
- Secure authentication with password hashing (bcrypt)
- CSRF protection for secure form submissions
- User profile management

#### 2. Waste Collection Management

- Record waste collections with:
  - Waste type
  - Quantity
  - Collection date
  - Location
  - Status tracking

- View, edit, and delete collection records
- Filtering, sorting, and pagination for easy navigation

3. Schedule Management

- Create collection schedules
- Assign teams to routes
- Track schedule status
- View upcoming collections

4. Advanced Reporting

- Collection statistics dashboard
- Waste type distribution visualization
- Export collection data in CSV format
- Advanced performance metrics with interactive charts

Technical Implementation

Technology Stack

- **Backend:** Python 3.12 with Flask
- **Database:** SQLite via SQLAlchemy ORM
- **Frontend:**
  - HTML5, CSS, Bootstrap 5
  - JavaScript
  - Chart.js for interactive graphs
  - SweetAlert2 for notifications
  - AJAX for dynamic updates

Project Structure

```
wms/
├── app.py                # Application configuration and initialization
├── routes.py             # Main route handlers
├── models.py             # Database models
├── requirements.txt       # Project dependencies
├──
├── blueprints/           # Feature modules
│   ├── admin.py          # Admin functionality
│   ├── auth.py           # Authentication
│   ├── operator.py        # Collection operations
│   └── reporting.py       # Reporting functionality
├──
├── static/               # Static assets
│   ├── css/
│   │   └── styles.css    # Custom styling
│   └── js/
│       └── main.js        # Core JavaScript
```

```
├── templates/                # HTML templates
│   ├── admin/                # Admin interface
│   │   ├── dashboard.html
│   │   ├── users.html
│   │   └── waste_types.html
│   ├── operator/            # Operator interface
│   │   ├── collections.html
│   │   └── schedules.html
│   ├── reporting/           # Reporting interface
│   │   ├── dashboard.html
│   │   └── reports.html
│   └── auth/                 # Authentication
│       ├── login.html
│       └── register.html
└── instance/                # Instance data
    ├── wms.db                # SQLite database with pre-configured data
    └── reports/              # Directory for generated CSV reports
```

## Security Features

- **Password Hashing:** Secure storage using bcrypt
- **CSRF Protection:** Prevents cross-site request forgery attacks
- **Role-Based Access Control:** Restricts access based on user roles
- **Form Validation:** Prevents SQL injection and XSS attacks
- **Session Management:** Secure login sessions with expiration handling

## Known Limitations

1. Basic reporting functionalities (no advanced analytics yet)
2. Limited data visualization (interactive dashboards can be improved)
3. Single database support (future versions may support multiple DBs)
4. No API endpoints for external integrations (planned for future updates)

## Important notes and Support

This software is being in progress of deployment and will be available soon on the internet for beta testing, and then open sourced, For more information, contact:

- **DABO R Yanis Axel**
- Email: [axel.studiesmail@gmail.com](mailto:axel.studiesmail@gmail.com)
- WhatsApp: [+226 67 26 08 83](https://wa.me/22667260883)

## License

This project is released under the **MIT License**, with the following permissions and conditions:

- Permission is granted to use, copy, and modify the software for **personal or educational purposes**.

- Redistribution of the software or modifications thereof is allowed, provided that the copyright notice is included in all copies or substantial portions of the software.
- **Commercial use or sublicensing of the software is not permitted without explicit permission** from the author.
- The software is provided "as is", without warranty of any kind, either express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, or 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.

By using this software, you agree to the terms outlined above.