

System Documentation for SearchForOrganics.com

1. Introduction

SearchForOrganics.com is a web-based search engine designed to help users find certified organic products. This document provides an overview of the system architecture, configuration details, and maintenance procedures for the SearchForOrganics.com platform.

2. System Architecture

Components

- **Frontend**: The frontend is built using HTML, CSS, and JavaScript. It provides the user interface for searching and browsing organic products.
- **Backend**: The backend is powered by a server-side application developed in Python using the Django framework. It handles user requests, processes search queries, and communicates with the database.
- **Database**: The system uses a relational database (e.g., PostgreSQL) to store product information, user data, and search indexes.
- **Search Engine**: The search engine is implemented using Google Programmable Search Engine (Custom Search Engine) to provide customized organic product search results.

Architecture Diagram

![System Architecture Diagram](insert_diagram_url_here)

3. Configuration

Server Configuration

- **Hosting**: The system is hosted on a cloud server (e.g., AWS EC2) to ensure scalability and reliability.
- **Web Server**: The web server (e.g., Nginx) is configured to serve the frontend and route requests to the backend application.
- **Database Server**: The database server (e.g., PostgreSQL) is configured to store and manage data efficiently.

Application Configuration

- **Django Settings**: Django settings are configured to connect to the database, manage static files, and handle security settings.
- **Google Custom Search Engine**: The Google Programmable Search Engine is configured with custom search parameters and integrated into the frontend.

Environmental Variables

- **Secrets**: API keys, database credentials, and other sensitive information are stored in environmental variables for security.

4. Maintenance Procedures

Regular Backups

- **Database Backups**: Regularly backup the database to prevent data loss in case of a server failure.
- **Codebase Backups**: Maintain a backup of the codebase to revert to a stable version in case of a deployment issue.

Security Updates

- **System Updates**: Regularly update the operating system and software packages to patch security vulnerabilities.
- **Application Updates**: Keep the application up to date with the latest security patches and bug fixes.

Monitoring and Logging

- **Monitoring**: Use monitoring tools to track system performance, detect issues, and ensure uptime.
- **Logging**: Implement logging to record system activities, errors, and user actions for troubleshooting and auditing purposes.

Disaster Recovery

- **Backup and Restore Procedures**: Have a documented procedure for restoring backups in case of data loss or system failure.
- **Redundancy**: Implement redundant systems (e.g., failover servers) to minimize downtime in case of a hardware failure.

5. Conclusion

This system documentation provides an overview of the architecture, configuration, and maintenance procedures for the SearchForOrganics.com platform. By following these guidelines, the system can be maintained efficiently and securely to provide users with a reliable and user-friendly organic product search engine.
