

# KilimoForum Documentation

## Introduction

**KilimoForum** is an online platform designed to connect customers, farmers, and sellers of agricultural tools. It facilitates discussions, provides information, and allows users to interact with each other regarding agricultural needs and tools.

## Table of Contents

1. [Features](#)
2. [Requirements](#)
3. [Installation](#)
4. [Usage](#)
5. [Project Structure](#)
6. [API Endpoints](#)
7. [Contributing](#)
8. [License](#)

## Features

- User registration and authentication
- Job postings for agricultural tools and services
- Search functionality for jobs and tools
- User profiles with role-based access
- Admin dashboard for managing content

## Requirements

- PHP 7.4 or higher
- MySQL or SQLite
- Laravel 9.x
- Composer
- XAMPP for local development (optional)

## Installation

### Clone the Repository

```
bash
Copy code
git clone https://github.com/yourusername/KilimoForum.git
cd KilimoForum
```

## Install Dependencies

```
bash
Copy code
composer install
```

## Configure Environment

1. Copy the `.env.example` file to `.env`:

```
bash
Copy code
cp .env.example .env
```

2. Update the `.env` file with your database and application settings:

```
env
Copy code
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=kilimoforum
DB_USERNAME=root
DB_PASSWORD=
```

## Generate Application Key

```
bash
Copy code
php artisan key:generate
```

## Run Migrations

```
bash
Copy code
php artisan migrate
```

## Start the Development Server

```
bash
Copy code
php artisan serve
```

Your application will be accessible at <http://localhost:8000>.

## Usage

- **Register:** Create an account to start interacting with the platform.
- **Post Jobs:** Users can post job opportunities related to agricultural tools.

- **Search:** Use the search functionality to find relevant jobs or tools.
- **Admin Dashboard:** Access the admin dashboard to manage users, jobs, and other content.

## Project Structure

- **app/:** Contains the application's core code, including models, controllers, and middleware.
- **config/:** Configuration files for various aspects of the application.
- **database/:** Database migrations and seeders.
- **public/:** Publicly accessible files such as assets and the entry point for the application.
- **resources/:** Views, language files, and raw assets.
- **routes/:** Route definitions for the application.
- **storage/:** Storage for logs, cached files, and file uploads.
- **tests/:** Automated tests for the application.

## API Endpoints

- **POST /api/register:** Register a new user.
- **POST /api/login:** Authenticate a user and return a token.
- **GET /api/jobs:** Retrieve a list of job postings.
- **POST /api/jobs:** Create a new job posting (admin only).
- **GET /api/jobs/{id}:** Retrieve details of a specific job posting.
- **PUT /api/jobs/{id}:** Update a specific job posting (admin only).
- **DELETE /api/jobs/{id}:** Delete a specific job posting (admin only).

## Contributing

1. Fork the repository.
2. Create a new branch (`git checkout -b feature/YourFeature`).
3. Make your changes.
4. Commit your changes (`git commit -am 'Add some feature'`).
5. Push to the branch (`git push origin feature/YourFeature`).
6. Create a new Pull Request.

## License

This project is licensed under the MIT License.