

# Django Notes App

## User Guide & Documentation

---

### Introduction

The Django Notes App is a secure web application designed for efficient note-taking and management. Built with Django, this application enables users to create personal accounts and manage their notes through an intuitive interface.

This document provides comprehensive instructions for installing, configuring, and using the application.

---

### Features

#### User Authentication

- User registration with secure credential management
- Login/logout functionality
- Password protection and security

#### Notes Management

- Create new notes with titles and content
- View organized list of all personal notes
- Access detailed view of individual notes
- Edit existing notes with full version control
- Delete unwanted notes

#### User Interface

- Clean, intuitive design
  - Responsive layout that works across devices
  - Mobile-friendly interface
- 

### Technical Architecture

#### Technology Stack

- **Backend Framework:** Django
- **Database:** MySQL
- **Containerization:** Docker and Docker Compose
- **Frontend:** Django templates with responsive design

## Application Structure

notes\_app/

```
├── notes/           # Notes management app
├── notes_app/       # Main project directory
├── static/          # Static files
├── staticfiles/     # Static files
├── templates/       # HTML templates
├── users/           # User authentication app
├── .dockerignore    # Docker ignore file
├── .env             # Environment variables
├── docker-compose.yml # Docker Compose configuration
├── Dockerfile        # Docker configuration
├── manage.py        # Command line
└── Requirements.txt  # Required Packages
```

---

## Installation & Setup

### Prerequisites

Before installing the application, ensure you have the following software installed:

- Git
- Docker

### Installation Steps

1. **Clone the Repository**
2. `git clone https://github.com/Sidharth-Chirathazha/Notes-App.git`

3. `cd notes_app`

#### 4. **Configure Environment Variables**

Create a `.env` file in the root directory with the following content:

`DJANGO_SECRET_KEY=Your Django Secret Key`

`DEBUG=True(For development)/ False(For production)`

`MYSQL_ENGINE=django.db.backends.mysql`

`MYSQL_HOST=Your Db Host`

`MYSQL_PORT=Your Db Port`

`MYSQL_DATABASE=Your Db Name`

`MYSQL_USER=Your Db User Name`

`MYSQL_PASSWORD=Db Password`

`MYSQL_ROOT_PASSWORD=Db Root Password`

#### 5. **Launch with Docker**

Build and start the containers:

`docker-compose up -d`

The application will be available at: `http://localhost:8000`

#### 6. **Stopping the Application**

Standard shutdown:

`docker-compose down`

Complete removal (including data):

`docker-compose down -v`

---

## **Usage Guide**

### **User Registration and Authentication**

#### 1. **Create an Account**

- Navigate to `/register/`
- Fill in the required information
- Submit the registration form

## 2. Log In

- Visit /login/
- Enter your credentials
- Click "Login"

## 3. Log Out

- Click the "Logout" option when finished

## Notes Management

### 1. View All Notes

- After logging in, you'll be directed to the notes dashboard
- All your existing notes will be displayed in a list

### 2. Create a New Note

- Click the "Add Note" button
- Enter a title and content for your note
- Click "Save" to store the note

### 3. View Note Details

- Click on any note title from the list
- The full note details will be displayed

### 4. Edit a Note

- From the note detail view, click "Edit"
- Make your changes to the title or content
- Click "Save" to update the note

### 5. Delete a Note

- From the note detail view, click "Delete"
- Confirm the deletion when prompted

---

## API Documentation

### Authentication Endpoints

URL	Method	Description
/	GET	Home page
/register/	GET, POST	User registration
/login/	GET, POST	User login
/logout/	GET	User logout

## Notes Endpoints

URL	Method	Description
/notes/	GET	List all notes for the logged-in user
/note/<uuid:note_id>/	GET	View a specific note's details
/note/add/	GET, POST	Add a new note
/note/<uuid:note_id>/edit/	GET, POST	Edit an existing note
/note/<uuid:note_id>/delete/	GET, POST	Delete a note

---

## Development Guide

### Database Management

1. **Running Migrations**
2. `docker-compose exec web python manage.py makemigrations`
3. `docker-compose exec web python manage.py migrate`
4. **Creating an Administrator Account**
5. `docker-compose exec web python manage.py createsuperuser`
6. **Accessing the Admin Interface**

Navigate to `http://localhost:8000/admin` and log in with your superuser credentials.

### Database Configuration

The application uses MySQL as its database system. Connection parameters are managed through environment variables defined in the `.env` file and Docker Compose configuration.

---

## Troubleshooting

## Common Issues and Solutions

### Database Connection Problems

- Verify the MySQL container is running: `docker-compose ps`
- Check database credentials in the `.env` file
- Ensure MySQL service has fully initialized before web service connection

### Application Not Loading

- Review error logs: `docker-compose logs web`
- Confirm all migrations have been applied
- Check Docker container status and health

### Authentication Issues

- Clear browser cookies and cache
  - Verify user credentials
  - Reset password if necessary
- 

## Support & Contact

### Project Information

- **Developer:** Sidharth Chirathazha
- **Email:** [sidharthchirathazha@gmail.com](mailto:sidharthchirathazha@gmail.com)
- **GitHub Repository:** <https://github.com/Sidharth-Chirathazha/Notes-App>

### Contributing

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

### License

This project is licensed under the MIT License.