Theft Alert System for Supermarkets Installation Manual­­

**Secure View Theft Alert System**

# Introduction

## Purpose of this Manual

This manual provides a comprehensive, step-by-step guide to installing the CCTV Security Platform. It is designed for system administrators and IT professionals who are responsible for deploying and maintaining the platform. The manual ensures a seamless setup process, from preparing the system environment to configuring the application and verifying its functionality.

## System Requirements

To deploy the CCTV Security Platform effectively, ensure that your system meets the following requirements:

**Hardware:**

**Operating System:** Ubuntu 20.04 LTS or later **CPU:** 2 cores or more

**RAM:** 4GB minimum, 8GB recommended **Storage:** 50GB minimum

**Network:**

**Internet Connection:** Stable connection with at least 5Mbps upload and download speed

**Software:**

**Python 3.8** or later

**pip** (Python package manager)

**Git**

# Prerequisites

## Required Software

Ensure the following software is installed on your Windows system:

**Python 3.8** or later

Download from the of cial [Python website](https://www.python.org/).

During installation, check the box for **"Add Python to PATH"**.

**pip** (Python package manager)

Included with Python installation. Verify with:

python --version pip --version

**Git**

Download and install from [Git](https://git-scm.com/).

### Text Editor or IDE

Recommended: Visual Studio Code or PyCharm.

### Virtualenv

Install using pip after Python installation:

pip install virtualenv

## System Preparation

### Check Python Installation

Open the Command Prompt and con rm Python is installed correctly:

python --version pip --version

### Create a Project Directory

Set up a dedicated directory for the CCTV Security Platform:

mkdir C:\CCTV\_Security\_Platform cd C:\CCTV\_Security\_Platform

### Set Up a Virtual Environment

Create and activate a virtual environment to isolate project dependencies:

python -m venv venv

.\venv\Scripts\activate

If successful, your Command Prompt should display (venv) before the prompt.

### Install Required Python Libraries

Once the virtual environment is active, install Django and other dependencies:

pip install django

# Python Environment Setup

## Verifying Python Installation

Ensure Python and pip are correctly installed by running the following commands in Command Prompt:

python --version pip --version

## Setting Up a Virtual Environment

### Navigate to Your Project Directory

Create and switch to the directory where the project will reside:

mkdir C:\CCTV\_Security\_Platform cd C:\CCTV\_Security\_Platform

### Create a Virtual Environment

Use virtualenv to create an isolated Python environment:

python -m venv venv

### Activate the Virtual Environment

Activate the environment to ensure dependencies are installed locally:

.\venv\Scripts\activate

When activated, (venv) will appear in the command prompt.

### Install Project Dependencies

With the virtual environment active, install required dependencies:

pip install -r requirements.txt

### Deactivate the Virtual Environment

When finished, deactivate the virtual environment:

deactivate

# Application Installation

## Cloning the Repository

### Install Git

Ensure Git is installed on your system (refer to the prerequisites section).

### Clone the Repository

Use the following command to clone the CCTV Security Platform repository:

git clone https://github.com/KipkorirVictor/Supermaket-Survaillance-System

cd Supermaket-Survaillance-System

## Installing Dependencies

### Activate the Virtual Environment

Ensure you are inside your project directory and activate the virtual environment:

.\venv\Scripts\activate

### Install Python Dependencies

Install all required dependencies listed in the requirements.txt le:

pip install -r requirements.txt

## Con guring Environment Variables

### Create an Environment File

Create a .env le in the root directory of your project:

echo. > .env

### Set Environment Variables

Open the .env le using a text editor (e.g., Notepad):

notepad .env

Add the following conguration, replacing placeholders with your actual values:

DEBUG=False SECRET\_KEY=your\_secret\_key\_here ALLOWED\_HOSTS=127.0.0.1,localhost

### Save and Close

Ensure the .env le is saved in the project directory.

# Django Configuration

## Setting Up the settings.py File

### Open the settings.py File

Navigate to the cctv\_security\_platform directory and open the settings.py le for editing:

notepad cctv\_security\_platform\settings.py

### Configure the Database

By default, Django uses SQLite. Ensure the following configuration is set in settings.py to use SQLite:

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3', 'NAME': BASE\_DIR / 'db.sqlite3',

}

}

### Configure Static and Media Files

Update the static and media les settings to ensure proper le handling:

STATIC\_URL = '/static/'

STATIC\_ROOT = BASE\_DIR / 'static les'

MEDIA\_URL = '/media/'

MEDIA\_ROOT = BASE\_DIR / 'media'

### Set Allowed Hosts

Configure the allowed hosts to allow access from your local server or domain. Add your domain or localhost:

ALLOWED\_HOSTS = ['127.0.0.1', 'localhost', 'your\_domain.com']

### Save and Close

Save the changes and close the editor.

## Additional Configuration (Optional)

### Email Configuration

If you need to configure email for Django, and the following settings and all in with your SMTP server details (optional for testing purposes):

EMAIL\_BACKEND = 'django.core.mail.backends.smtp.EmailBackend' EMAIL\_HOST = 'smtp.gmail.com' *# or your SMTP server* EMAIL\_PORT = 587

EMAIL\_USE\_TLS = True

EMAIL\_HOST\_USER = ['your\_email@example.com'](mailto:%27your_email@example.com) EMAIL\_HOST\_PASSWORD = 'your\_email\_password'

### Security Settings

For a production environment, consider setting DEBUG=False.

You can also configure other security-related settings like CSRF\_COOKIE\_SECURE, SESSION\_COOKIE\_SECURE, and X\_FRAME\_OPTIONS.

# Database Migration

## Running Initial Migrations

Django uses migrations to set up the database schema. Run the following command to apply initial migrations and create the necessary database tables:

### Apply Migrations

In the project directory, run the following command to apply migrations:

python manage.py migrate

This command will create the default SQLite database le (db.sqlite3) in the project directory and apply any required migrations for the app.

### Verify Migration Success

If migrations are successful, you should see output indicating that tables have been created in the database. If any issues occur, the error messages will guide you toward the solution.

## Creating a Superuser

A superuser is required to access the Django admin interface. Follow these steps to create a superuser:

### Create Superuser

Run the following command to start the process of creating a superuser:

python manage.py createsuperuser

### Enter Superuser Details

You will be prompted to provide the following details for the superuser account:  **Username**: Choose an admin username.

**Email**: Enter the admin email address.

**Password**: Set a secure password for the superuser.

After entering the required details, you should see a success message indicating that the superuser has been created.

### Verify Superuser Creation

After successful creation, you can access the Django admin interface by visiting [http://127.0.0.1](http://127.0.0.1/) 8000/admin in your browser and logging in with the superuser credentials.

# Testing the Installation

## Running Django Development Server

To verify that the application is set up correctly, you can run the Django development server. This will allow you to access the site locally on your machine.

### Activate the Virtual Environment

Ensure that your virtual environment is active:

.\venv\Scripts\activate

### Run the Development Server

Start the Django development server using the following command:

python manage.py runserver

The server will start, and you should see output indicating that the server is running on [http://127.0.0.1](http://127.0.0.1/) 8000.

### Access the Application

Open a web browser and visit:

[http://127.0.0.1](http://127.0.0.1/) 8000

You should see the home page of the CCTV Security Platform application.

## Accessing the Admin Interface

The Django admin interface provides an easy way to manage the application’s content and users.

**Access the Admin Panel** In a web browser, visit:

[http://127.0.0.1](http://127.0.0.1/) 8000/admin

### Log in with Superuser Credentials

Use the superuser username and password you created earlier to log in.

### Verify Admin Access

After logging in, you should be able to see the admin interface where you can manage users, videos, and other aspects of the system.

## Testing Video Streams

If your installation includes video streaming, you will want to test whether the video feed is working correctly.

### Start Video Streams (if applicable)

Depending on your con guration, start the video streams or ensure that the video sources are properly connected.

### Access Stream URLs

If the application includes video streaming, you can access the streams either through a URL provided in

the admin panel or by visiting speci c endpoints in your browser (e.g., [http://127.0.0.1](http://127.0.0.1/) 8000/video\_stream).

### Verify Video Feed

Ensure that the video stream loads and is displayed properly in the application. If there are issues, check the con guration and logs for troubleshooting.