Lost And Found App 13P documentation

Zhangeldin Nurdaulet 220103261

Project Title: Lost and Found
Description: A web-based application for posting, managing, and searching lost or found items within a community or organization.

Technology Stack:

Programming Language: Python 3
Framework: Django
Database: PostgreSQL
Frontend: Basic HTML/CSS (rendered via Django templates) and Bootstrap
Containerization: Docker (with Dockerfile and docker-compose.yml)
Web Server: Nginx (configured in the nginx/ folder)
Environment Management: .env.prod for production secrets and environment variables
Dependency Management: requirements.txt

Project Structure:

```
LostAndFoundApp_13P/

├── items/               # Django main app

├── lost_and_found/      # Core Django project settings

├── nginx/               # Nginx configuration files

├── .env.prod            # Environment variables for production

├── Dockerfile           # Docker image build instructions

├── docker-compose.yml   # Multi-container orchestration (web, db, nginx)

├── Makefile             # CLI automation (build/run commands)

├── manage.py            # Django utility script

├── requirements.txt     # Python dependencies

└── README.md            # Project overview (already included)
```

Project models:

Item Model:

Field Name	Type	Description
title	CharField	Short title of the item (max 100 chars)
description	TextField	Detailed description of the item
contact_telegram	CharField	Optional Telegram contact for reaching the user (max 255 chars)
image	ImageField	Optional image of the item (uploaded via item_image_path utility)
created_at	DateTimeField	Timestamp when the item was created (auto-generated)
user	ForeignKey → User	Reference to the reporting user (nullable, on delete: set to NULL)
status	CharField	Status of the item — LOST or FOUND (Enum-based choices)

Comment Model:

Field Name	Type	Description
item	ForeignKey → Item	Reference to the associated item (cascade delete)
text	TextField	The content of the comment
created_at	DateTimeField	Timestamp when the comment was created (auto-generated)
user	ForeignKey → User	The user who left the comment (nullable, on delete: set to NULL)

Project Endpoints:

Core item:

URL Path	HTTP Method(s)	View	Description
`/`	`GET`	`ItemListView`	Displays a paginated list of all items. Supports search by keyword and filtering by status (`LOST` or `FOUND`).
`/<int:pk>/`	`GET`, `POST`	`ItemDetailView`	Shows item details. Also supports direct comment submission via `POST`.

`/create/`	`GET`, `POST`	`ItemCreateView`	Presents a form to create a new item. Requires user to be logged in.
`/item/<int:pk>/delete/`	`POST`	`ItemDeleteView`	Deletes an item. Only available to the item's owner or admin users.
`/change_status/<int:pk>/`	`POST`	`ChangeStatusView`	Allows item owner or admin to change item status (`LOST` or `FOUND`).

User functionality:

URL Path	HTTP Method(s)	View	Description
`/login/`	`GET`, `POST`	`LoginView`	Standard login page using Django's built-in authentication (template: `login.html`).
`/logout/`	`GET`	`LogoutView`	Logs the user out and redirects appropriately.
`/register/`	`GET`, `POST`	`RegisterView`	User registration form. Automatically logs in the user upon successful registration.
`/profile/`	`GET`	`ProfileView`	Displays the profile of the currently logged-in user.
`/profile/<int:user_id>/`	`GET`	`ProfileView`	Displays another user's profile and their items.

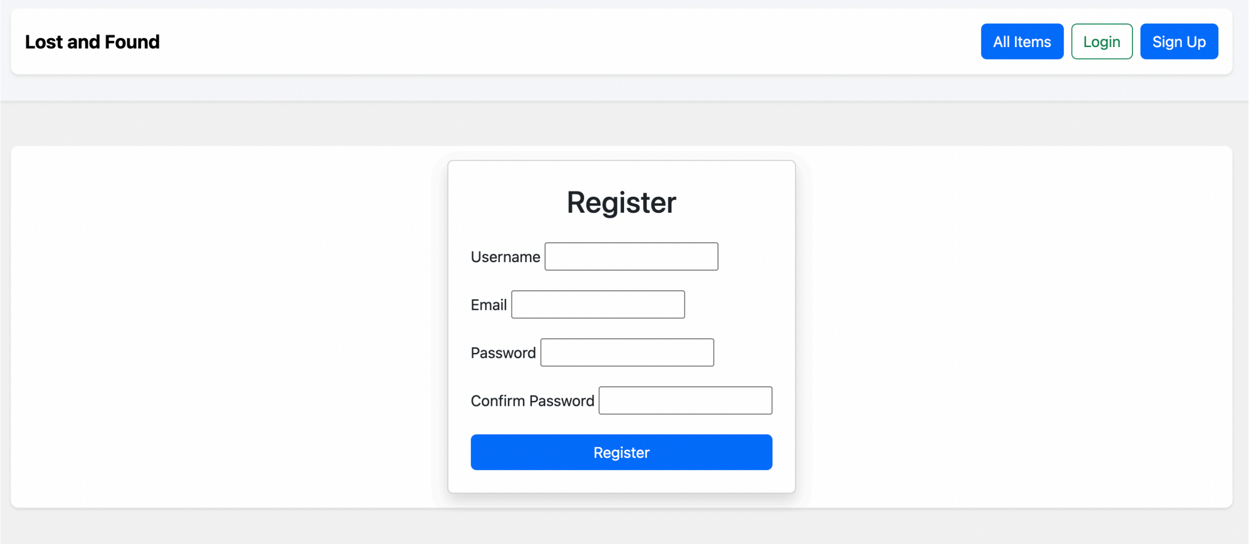
Comments:

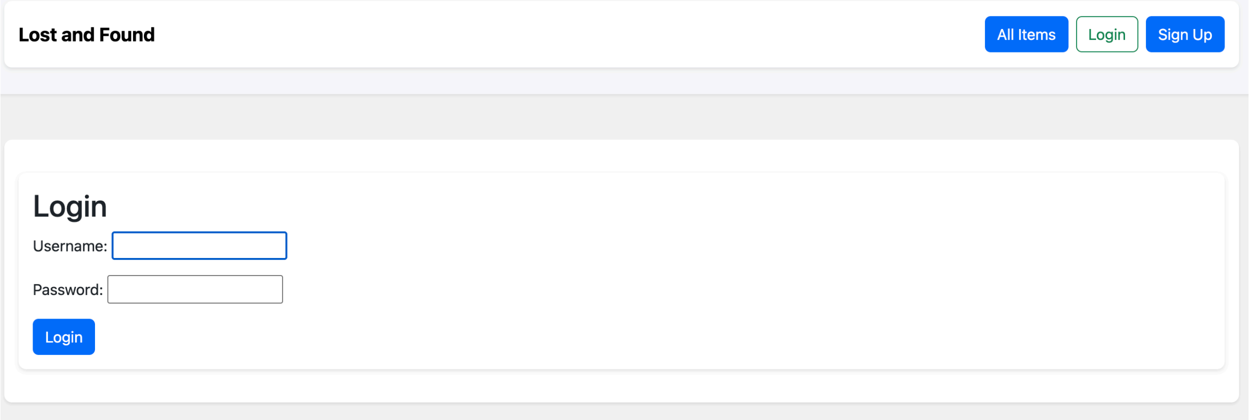
URL Path	HTTP Method(s)	View	Description
`/items/<int:item_id>/comment/`	`POST`	`CommentCreateView`	Handles comment form submission for a specific item. Authenticated users are linked to the comment; guests are also allowed.

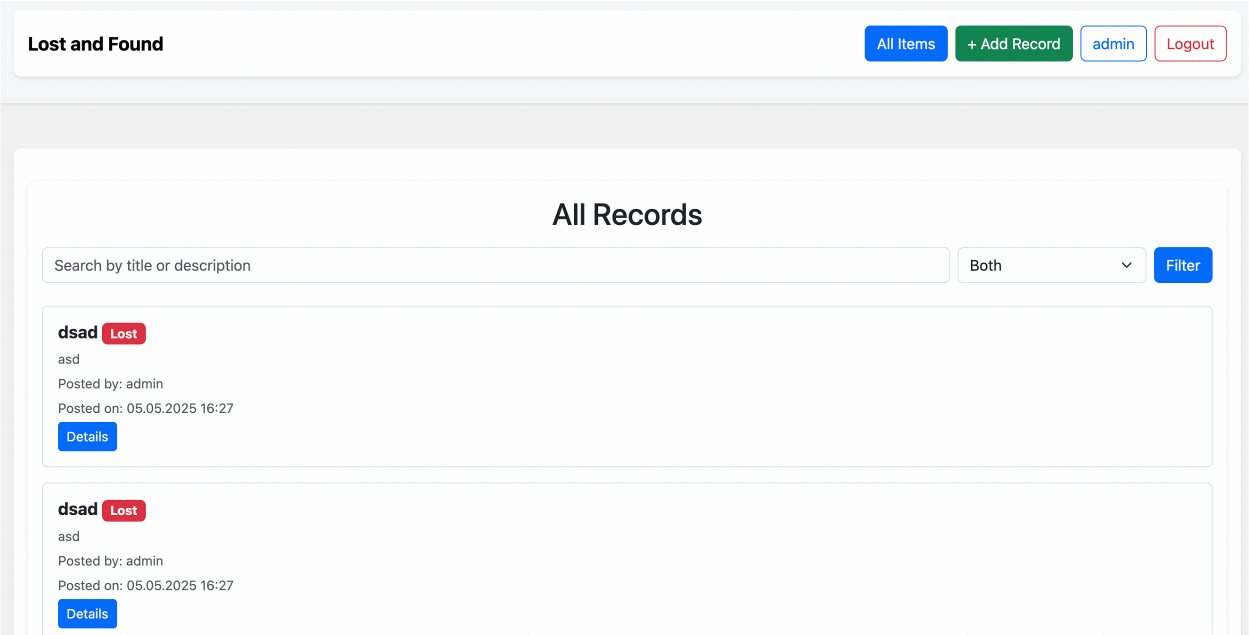
Additional

URL Path	Description
/api/	Includes API routes from the items app. Likely for RESTful access
/silk/	Silk profiling tool for performance monitoring and debugging. Useful during development.
/admin/	Django admin panel. Used for managing users, items, and comments through the admin interface.

Worflow:







dsad Lost

asd

Posted by: admin

Posted on: 05.05.2025 16:27

Details

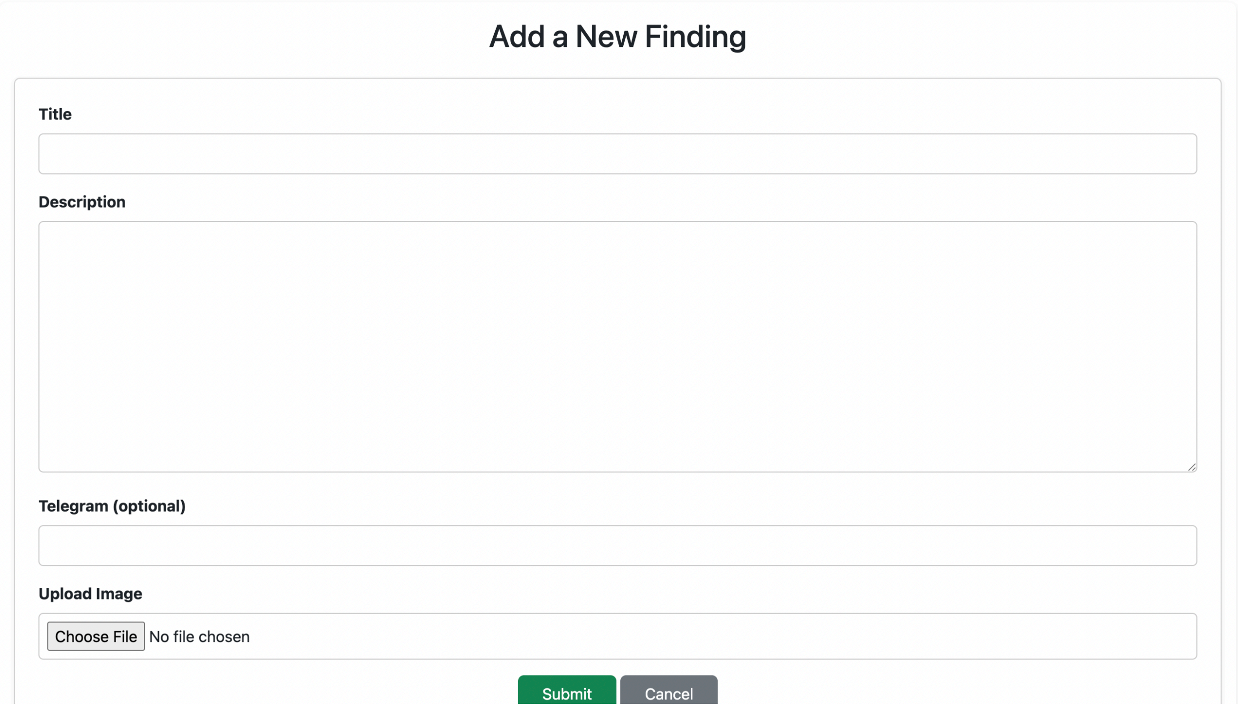
dsad Lost

asd

Posted by: admin

Posted on: 05.05.2025 16:27

Details



Add a New Finding

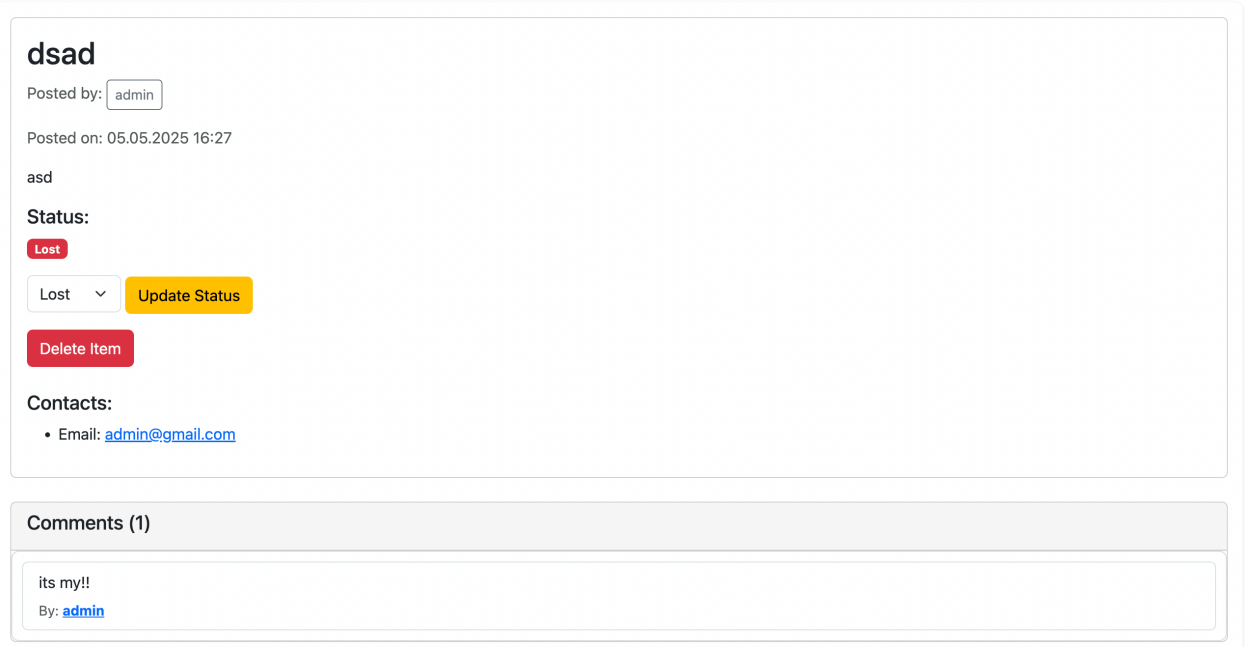
Title

Description

Telegram (optional)

Upload Image

Choose File No file chosen



dsad

Posted by: admin

Posted on: 05.05.2025 16:27

asd

Status:

Lost

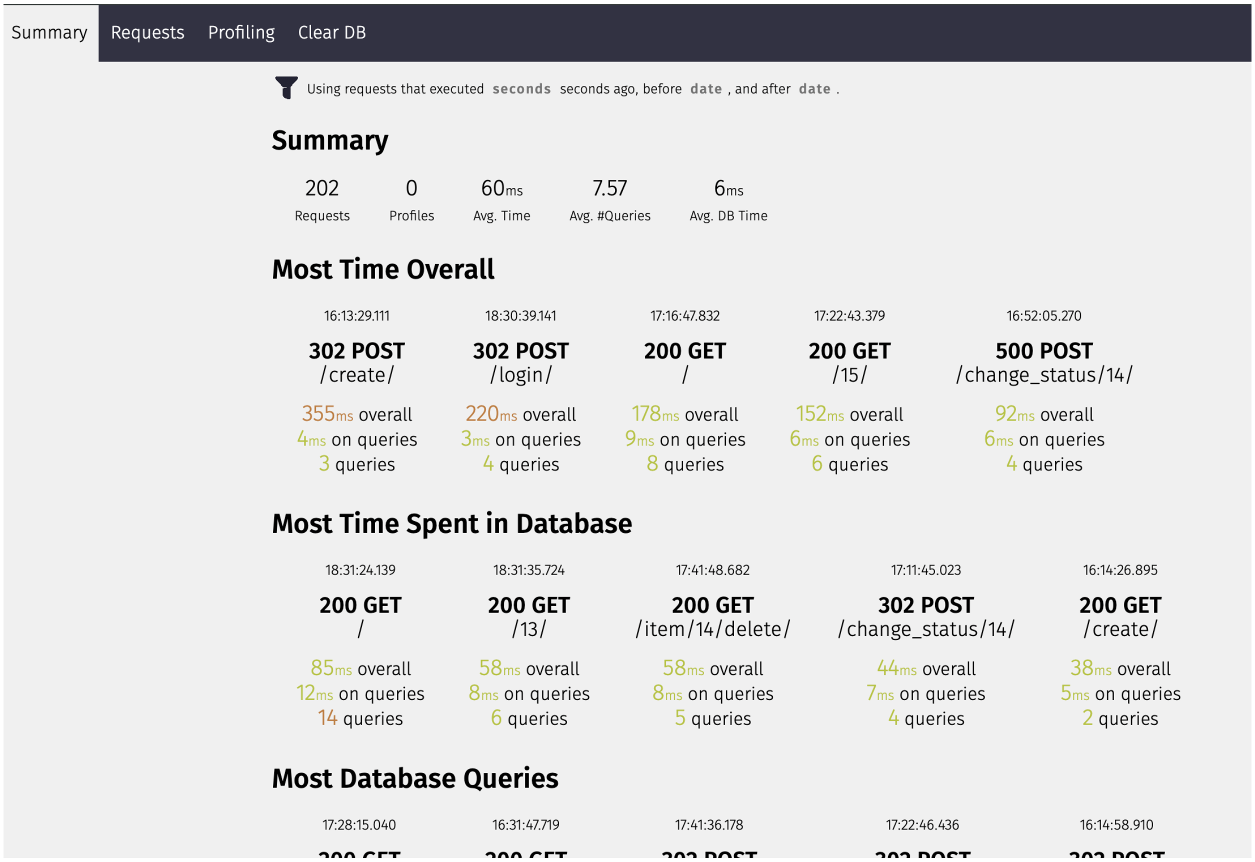
Contacts:

- Email: admin@gmail.com

Comments (1)

its my!!

By: **admin**



Using requests that executed seconds seconds ago, before date , and after date .

Summary

202	0	60ms	7.57	6ms
Requests	Profiles	Avg. Time	Avg. #Queries	Avg. DB Time

Most Time Overall

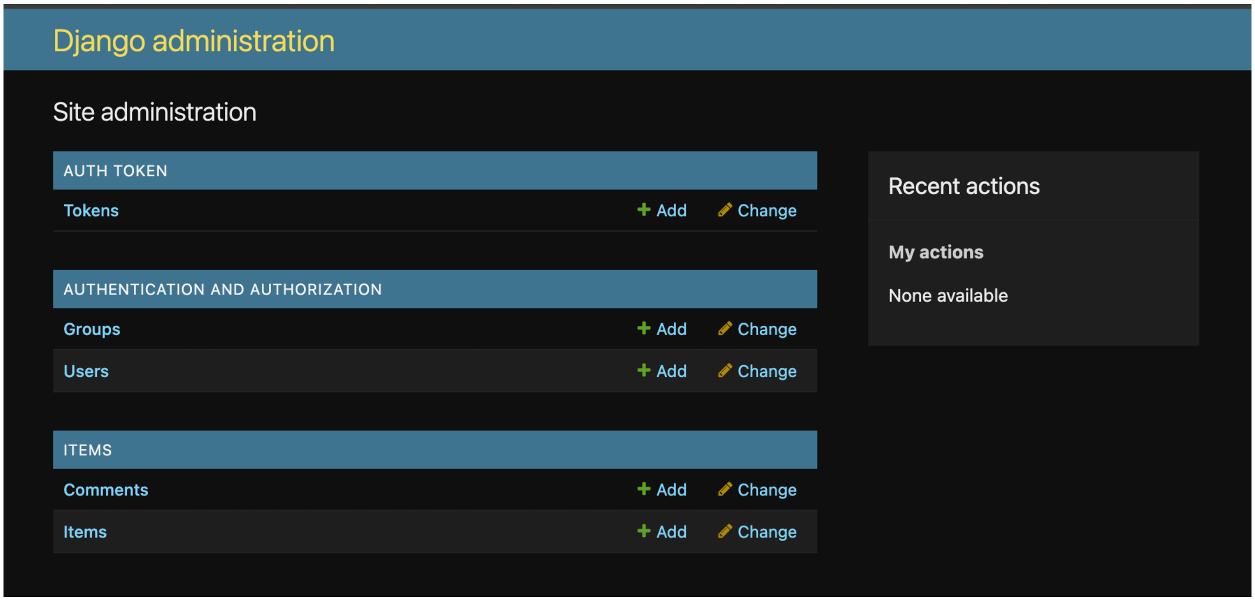
16:13:29.111	18:30:39.141	17:16:47.832	17:22:43.379	16:52:05.270
302 POST /create/	**302 POST** /login/	**200 GET** /	**200 GET** /15/	**500 POST** /change_status/14/
355ms overall 4ms on queries 3 queries	220ms overall 3ms on queries 4 queries	178ms overall 9ms on queries 8 queries	152ms overall 6ms on queries 6 queries	92ms overall 6ms on queries 4 queries

Most Time Spent in Database

18:31:24.139	18:31:35.724	17:41:48.682	17:11:45.023	16:14:26.895
200 GET /	**200 GET** /13/	**200 GET** /item/14/delete/	**302 POST** /change_status/14/	**200 GET** /create/
85ms overall 12ms on queries 14 queries	58ms overall 8ms on queries 6 queries	58ms overall 8ms on queries 5 queries	44ms overall 7ms on queries 4 queries	38ms overall 5ms on queries 2 queries

Most Database Queries

17:28:15.040	16:31:47.719	17:41:36.178	17:22:46.436	16:14:58.910
200 GET	**200 GET**	**302 POST**	**302 POST**	**302 POST**



Django administration

Site administration

AUTH TOKEN

Tokens + Add ✎ Change

AUTHENTICATION AND AUTHORIZATION

Groups + Add ✎ Change

Users + Add ✎ Change

ITEMS

Comments + Add ✎ Change

Items + Add ✎ Change

Recent actions

My actions

None available