Task Management Application - Svelte + Django REST

Author

- **Alesia Hajrulla**

All development was completed independently. The project required approximately 25–30 hours of work, covering backend, frontend, testing, and integration.

Project Overview

This full-stack application allows users to manage tasks with a simple and intuitive interface. It supports:

- Adding, editing, and deleting tasks
- Marking tasks as done or to do
- Searching and filtering tasks
- Paginated view for better UX

The backend is built using **Django REST Framework**, while the frontend uses **Svelte** with **Tailwind CSS**.

Functionalities

- ✅ Create a task
- ✅ Edit or delete an existing task
- ✅ Toggle task status: *To Do* ↔ *Done*
- ✅ Filter by status (All, To Do, Done)
- ✅ Search by task title
- ✅ View tasks across pages with pagination

Backend: Django REST

- API developed with Django 5.2 and Django REST Framework
- SQLite3 used as local database
- CORS enabled to allow frontend access
- All API endpoints are public (no login/auth)
- Pagination implemented using `PageNumberPagination`

How to Run (Backend)

```
cd taskmanager
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
python manage.py migrate
python manage.py runserver
```

API base URL: `http://localhost:8000/api/tasks/`
Custom search endpoint: `http://localhost:8000/api/tasks/search/` (POST)

Frontend: Svelte + Tailwind CSS

- User-friendly interface using Svelte
- Live search and dropdown filter
- Modal form for task creation/editing
- Pagination component fixed at the bottom

How to Run (Frontend)

```
cd task-frontend
npm install
npm run dev
```

App will run at: `http://localhost:5173/`

Make sure the backend is running at `localhost:8000`.

Code Structure

```
project-root/
│
├── taskmanager/           # Backend Django API
│   ├── tasks/             # Django app (models, views, serializers)
│   ├── db.sqlite3         # Local database
│   ├── manage.py
│   └── requirements.txt
│
└── task-frontend/         # Frontend Svelte app
    ├── src/components/    # TaskTable.svelte, TaskForm.svelte
    ├── src/stores.js      # Svelte store for tasks
    ├── package.json
    └── App.svelte         # Main application UI
```

Notes

- This project is meant for **local development** and **demonstration**.
- No authentication or user-specific filtering is implemented.
- You can reset the database anytime using:

```
rm db.sqlite3
python manage.py migrate
```

License

This project is provided for academic and training purposes. Feel free to reuse and adapt with credit to the original author.