School Management System API

A Django REST Framework (DRF) based API for managing schools, students, and user authentication using JSON Web Tokens (JWT). This project also incorporates tools for development, testing, and production environments.

Features

Core Functionality

- Manage **Schools** and **Students** resources with the following:
  - CRUD operations.
  - Filtering, searching, and sorting.
  - Pagination support.
- **User Authentication**:
  - Login/Logout endpoints using JWT.
  - Custom user roles (`Admin`, `Staff`) with role-based permissions.
- Assign **Custom Permissions** for Admins and Staff:
  - `IsAdmin` and `IsStaffOrAdmin`.
- Middleware for debugging and performance profiling:
  - `debug_toolbar` for development.
  - `silk` for profiling.

Tech Stack

- **Backend Framework**: Django 5.1.5
- **API Framework**: Django REST Framework (DRF)
- **Authentication**: JWT (via `rest_framework_simplejwt`)
- **Database**: SQLite (default, switchable to PostgreSQL for production)
- **Development Tools**:
  - `django-debug-toolbar` (for debugging)
  - `silk` (for performance profiling)
  - `pytest` (for testing)
  - `drf-spectacular` (for API documentation)

Installation

Prerequisites

- Python 3.10+
- Pip (Python package installer)

- Virtual Environment (recommended: venv)

1. Clone the Repository

```
git clone https://github.com/yourusername/school-management-api.git
cd school-management-api
```

Note: You have to do some extra setup for the production environment according to your specific deployment needs and infrastructure requirements.

2. Create and Activate a Virtual Environment

- *For Linux/Mac:*

```
python -m venv venv
source venv/bin/activate
```

- *For Windows:*

```
python -m venv venv
venv\Scripts\activate
```

3. Install Dependencies

- Base Dependencies:

```
pip install -r requirements/base.txt
```

- Development Dependencies:

```
pip install -r requirements/dev.txt
```

- Production Dependencies:

```
pip install -r requirements/prod.txt
```

Environment Setup

1. Set the Django Settings Module

For development:

- On Linux/Mac

```
export DJANGO_SETTINGS_MODULE=config.settings.dev
```

- *On Windows*

```
set DJANGO_SETTINGS_MODULE=config.settings.dev
```

For production:

- On Linux/Mac

```
export DJANGO_SETTINGS_MODULE=config.settings.prod
```

- *On Windows*

```
set DJANGO_SETTINGS_MODULE=config.settings.prod
```

2. Apply Migrations

```
python manage.py migrate
```

3. Create a Superuser

```
python manage.py createsuperuser
```

Run the Server

- Development:

```
python manage.py runserver
```

To Test API on Browser

Open the following URL in your browser (running locally):

http://localhost:8000/api/schema/docs/

OR

http://localhost:8000/api/schema/redoc/

API Endpoints

Authentication

Method	Endpoint	Description
POST	`/api/auth/login/`	Obtain JWT tokens.
POST	`/api/auth/logout/`	Blacklist refresh tokens.

Schools

Method	Endpoint	Description
GET	`/api/schools/`	List all schools (with filters).
POST	`/api/schools/`	Create a new school.
GET	`/api/schools/{id}/`	Retrieve a school by ID.
PUT	`/api/schools/{id}/`	Update a school.
DELETE	`/api/schools/{id}/`	Delete a school (if no students).

Students

Method	Endpoint	Description
GET	`/api/schools/{school_id}/students/`	List students in a school (filterable).
POST	`/api/schools/{school_id}/students/`	Add a new student to a school.
GET	`/api/students/{id}/`	Retrieve a student by ID.
PUT	`/api/students/{id}/`	Update student information.
PATCH	`/api/students/{id}/`	Partially update student information.
DELETE	`/api/students/{id}/`	Delete a student.

Testing

- Run tests using pytest:

```
pytest
```