Project Manual for FAQ Backend System

# File Structure

The project file structure is organized as follows:  
  
faq\_backend/  
 ├── faq/  
 │ ├── \_\_init\_\_.py  
 │ ├── admin.py  
 │ ├── api/  
 │ │ ├── \_\_init\_\_.py  
 │ │ ├── urls.py  
 │ │ ├── views.py  
 │ ├── migrations/  
 │ │ ├── 0001\_initial.py  
 │ │ ├── \_\_init\_\_.py  
 │ ├── models.py  
 │ ├── serializers.py  
 │ ├── tasks.py  
 │ ├── tests.py  
 │ ├── urls.py  
 │ ├── views.py  
 ├── faq\_backend/  
 │ ├── \_\_init\_\_.py  
 │ ├── asgi.py  
 │ ├── settings.py  
 │ ├── urls.py  
 │ ├── wsgi.py  
 ├── db.sqlite3  
 ├── manage.py  
 ├── venv/  
 ├── .flake8  
 ├── README.md  
 ├── Startserver.md  
 ├── assumptions\_issues\_fixes.docx  
 ├── pytest.ini  
 ├── requirements.txt

# Project Setup Steps

Follow the steps below to set up the project and start the server:  
  
1. Clone the repository.  
  
2. Navigate to the project directory.  
  
3. Install the required dependencies.  
  
```bash  
pip install -r requirements.txt  
```  
  
4. Apply migrations.  
  
```bash  
python manage.py migrate  
```  
  
5. Create a superuser (optional).  
  
```bash  
python manage.py createsuperuser  
```  
  
6. Start the server.  
  
```bash  
python manage.py runserver  
```  
  
7. Access the API using the provided URLs for FAQs in different languages.

# Files Description

## 1. faq\_backend/faq/models.py

This file contains the Django models for the FAQ system, including fields for storing   
questions, answers, and translations for multiple languages.   
It defines the FAQ model and its methods.

## 2. faq\_backend/faq/views.py

This file contains the views for the FAQ API, including handling API requests and   
returning translated FAQ data based on the query parameter `lang`.

## 3. faq\_backend/faq/serializers.py

This file contains serializers that convert FAQ model instances into JSON format for API   
responses and handle deserialization for API requests.

## 4. faq\_backend/faq/tests.py

This file contains the unit tests for the FAQ models, views, and API responses.  
It includes tests to check proper API functionality and correct translations.

## 5. faq\_backend/faq/migrations/

This folder contains the database migration files. These files are automatically generated   
whenever you make changes to your models and run the `python manage.py makemigrations` command.

## 6. faq\_backend/settings.py

This file contains the settings and configurations for the Django project, including   
database setup, caching, language support, and third-party library integrations.

## 7. db.sqlite3

This is the SQLite database file that stores all the data for the project, including   
FAQs, translations, and other necessary application data.

## 8. manage.py

This is the command-line utility that allows you to interact with the Django project,   
including running the server, applying migrations, creating superusers, and more.

## 9. requirements.txt

This file contains a list of Python dependencies required to run the project,   
which can be installed using `pip install -r requirements.txt`.

## 10. .flake8

This is the configuration file for flake8, a linting tool for Python code that checks   
code quality according to the PEP8 guidelines.

## 11. README.md

This file provides an overview of the project, installation instructions, API usage,   
and contribution guidelines.

## 12. Startserver.md

This file contains instructions on how to start the development server for the project.  
It includes setup steps, troubleshooting, and commands for running the server.

## 13. assumptions\_issues\_fixes.docx

This document contains the assumptions made during the project implementation,   
issues encountered, and solutions applied to resolve those issues.

## 14. pytest.ini

This file contains the configuration for pytest, which is used to run the tests for   
the project.