Feedback System Project Documentation

1. Introduction

This document provides an overview and detailed explanation of the Feedback System project.

The project is implemented using Django (Python web framework) with a focus on user feedback storage and management. It incorporates both frontend and backend development aspects.

2. Project Setup and Structure

The project consists of the following key components:

- Backend: Django REST Framework with an SQLite3 database.
- Frontend: Static HTML, CSS, and Bootstrap 5 for UI styling.
- Media Management: Uploaded files are stored in a directory named 'Media'.
- App: A Django app named 'fbsystem' handles the core logic.

3. Backend Explanation

The backend is implemented using Django. Key components include:

Models (models.py):

- The Feedback model defines the structure of the database table. It includes fields for:
  - name (CharField, max_length=100)
  - faculty (CharField, max_length=100)

- review (TextField)
- image (ImageField, optional, stored in 'feedback_images/').

Views (views.py):

- feedback_view: Handles form submission and saves valid feedback to the database.
- feedback_success: Renders a success page upon successful feedback submission.

Forms (forms.py):

- FeedbackForm: A Django ModelForm tied to the Feedback model. It defines fields and widgets for user inputs.

Admin (admin.py):

- Customizes the Django admin interface to display, filter, and search feedback records.

4. Frontend Explanation

The frontend includes a feedback form created using static HTML, CSS, and Bootstrap 5.

Key Fields in feedback_form.html:

- Name: A text input for the user's name.
- Faculty: A text input for the faculty name.
- Review: A textarea for the user's feedback.
- Image: A file upload field for an optional image.

Styling:

- Custom styles are applied for the form, navbar, and buttons.

- Bootstrap is used for responsiveness and additional styling.

5. Static Files and Media

Static Files:

- Stored in a 'static' directory within the project.

- Includes CSS, images, and JavaScript files.

Media Files:

- User-uploaded files are stored in the 'Media' directory, configured in settings.py.

- MEDIA_URL: '/media/'

- MEDIA_ROOT: Path to the 'Media' directory in the project root.

6. Database Explanation

Database: SQLite3

- The default database for Django projects.

- Tables are created and managed automatically based on Django models.

How It Works:

- The Feedback model is translated into a database table during migrations.

- User submissions are stored as records in this table.

7. Connecting Frontend and Backend

The frontend interacts with the backend using Django templates.

- The feedback_form.html file includes a form that submits data to the backend via a POST request.
- The Django view (feedback_view) processes the data, validates it using the FeedbackForm, and saves it to the database upon successful validation.

8. Conclusion

This documentation outlines the Feedback System project, emphasizing the integration of Django for backend development and static files for the frontend. The project serves as an efficient solution for collecting and managing feedback within a university environment.