**Detailed Report for Social Media Website Developed**

**Introduction**

This report provides an overview of a social media website developed using Django. The website allows users to register, log in, post photos, and interact with others through likes and comments. Tailwind CSS has been used for styling, and the application currently has a nonfunctional share button.

**Key Features**

**User Authentication**

Registration: Users can create new accounts.

Login: Users can log into their accounts.

Logout: Users can log out of their accounts.

**Content Interaction**

Photo Uploads: Users can upload photos to their profile.

Feed: Uploaded photos are displayed in a feed visible to all users.

Likes: Users can like posts in the feed.

Comments: Users can comment on posts.

**Frontend Design**

Tailwind CSS: The website utilizes Tailwind CSS for styling, providing a modern and responsive design.

**Limitations**

NonFunctional Share Feature: The share button on posts is currently not operational.

Limited Interaction: The current features are limited to likes, comments, and photo uploads.

**Technologies Used**

**Backend**

Django: Serves as the core framework for building the website.

SQLite: Default database for storing user data and posts.

**Frontend**

Tailwind CSS: Used for styling the website’s interface.

HTML/CSS/JavaScript: Basic technologies for structuring and designing the frontend.

**System Requirements**

Python: The primary programming language used.

Django: Needs to be installed along with its dependencies.

Tailwind CSS: Requires Node.js for installation and setup.

**Installation and Configuration**

**Django Setup**

pip install django

After installation, Django is used to set up the project structure and applications.

Tailwind CSS Integration

Installation: Tailwind CSS is integrated into the project using Node.js and configured according to the project's requirements.

Database Configuration

SQLite: Used for development. Configured in Django’s settings to manage user data, posts, likes, and comments.

**Application Usage**

Running the Server: The Django server is started to host the application.

User Interaction: Users register, log in, post photos, and engage with others' posts.

**Conclusion**

The developed social media website demonstrates the essential functionalities expected from a basic social networking platform, leveraging Django's robust backend capabilities and Tailwind CSS for an appealing frontend. The current limitation, a nonfunctional share button, represents an area for future development and enhancement. This platform serves as a foundational base for a more feature-rich social media application.

OUTPUT:

