### **Project Overview Document**

**Project Title**: **FriendsNest**

**Project Description**:  
This project aims to develop a social media application where users can register, create posts, view other users' posts,Like and Comment posts, and engage in real-time chat if friend requests are accepted. The application will include features such as refreshing the feed using API polling, real-time chat using WebSockets, video and photo streaming using WebRTC, and robust authentication using OAuth. The application will be built with a focus on scalability, performance, and security.

**Project Objectives**:

1. **User Registration & Authentication**: Implement user registration, login, and authentication using OAuth to ensure secure access.
2. **Post Creation and Interaction**: Allow users to create posts, view others' posts, and interact through likes and comments via a REST API.
3. **Real-Time Chat**: Enable users to chat in real-time with other users if friend requests are accepted, using WebSockets for seamless communication.
4. **Media Streaming**: Implement video and photo streaming in posts using WebRTC for a rich media experience.
5. **Feed Refreshing**: Use API polling to refresh the feed periodically, ensuring users always see the latest posts.
6. **Database Management**: Leverage PostgreSQL for storing and managing data, ensuring efficient querying and data integrity.
7. **Scalability & Security**: Design the system to be scalable and secure, handling user growth and protecting user data.

**Scope**: The application will include the following features:

* **User Management**: Registration, login, and OAuth-based authentication.
* **Post Management**: Creating, viewing, liking, and commenting on posts.
* **Friendship Management**: Sending, receiving, and accepting friend requests.
* **Chat Functionality**: Real-time messaging between users with accepted friend requests.
* **Media Handling**: Streaming of video and photos in posts.
* **Feed Management**: Real-time feed updates using API polling.
* **Notifications**: Notify users of friend requests, chat messages, and interactions with their posts.

**Technology Stack**:

* **Backend**:
  + **Framework**: NestJS
  + **Database**: PostgreSQL
  + **Real-Time Communication**: WebSockets (via NestJS)
  + **Media Streaming**: WebRTC
  + **Authentication**: OAuth 2.0
  + **API**: RESTful API for CRUD operations and interactions
* **Frontend**:
  + **Framework**: React.js
  + **State Management**: Redux
  + **UI Library**: Tailwind CSS or Material-UI (optional, depending on design)
  + **Communication**: Axios or Fetch API for REST API calls, WebSocket for real-time chat
  + **Video Streaming**: WebRTC integration for video posts

**Key Components**:

1. **User Authentication**: Implement OAuth 2.0 for secure user authentication and authorization.
2. **Post Feed**: A dynamic feed that displays posts from users, updated in real-time or through periodic polling.
3. **Chat System**: Real-time messaging between users with WebSocket integration.
4. **Friendship Management**: Managing friend requests and relationships between users.
5. **Media Streaming**: Implementing WebRTC for video and photo streaming within posts.

**Success Criteria**:

* Successful user registration, authentication, and authorization using OAuth.
* Seamless post creation, viewing, liking, and commenting.
* Efficient and scalable real-time chat functionality.
* Smooth streaming of videos and photos in posts.
* Real-time feed updates with minimal latency.
* Secure handling of user data and interactions.

**Project Timeline**:

1. **Phase 1**: Project Setup & Authentication (2-3 weeks)
   * Set up NestJS, PostgreSQL, and React.
   * Implement OAuth-based user registration and login.
2. **Phase 2**: Post & Feed Management (3-4 weeks)
   * Develop RESTful APIs for post creation, viewing, liking, and commenting.
   * Implement feed refreshing using API polling.
3. **Phase 3**: Real-Time Chat (3 weeks)
   * Set up WebSocket communication for real-time messaging.
   * Implement a chat UI in React.
4. **Phase 4**: Media Streaming (3-4 weeks)
   * Integrate WebRTC for video and photo streaming in posts.
5. **Phase 5**: Testing, Optimization & Deployment (3-4 weeks)
   * Perform extensive testing.
   * Optimise for performance and security.
   * Deploy the application.