1• **Project overview**

The project aims to develop a web platform that enables users to connect, interact, and share content with others who have similar interests or existing relationships. Create an engaging and user-friendly platform that fosters online communities and facilitates meaningful connections between people.

2• **Functional requirements**

1. Post service

1. Post creation:

* Users can create video,image posts and text-based posts.

2. Edit and delete posts:

* Users can delete or edit their own existing posts.

3. Post privacy:

* Users can set post privacy whether it is public, private or to friends only.

4. Comment, like and share:

* Users can comment,like and share posts.
* Comment may include text, images and emojis.

2. Real-Time Notifications

Notification Types:

* Notifications for likes, comments, shares, new followers, friend requests and messages.

Real-Time Delivery:

* Notifications should be delivered in real-time using WebSocket

3. Friends and Followers

1. Friend Requests:

* Users can send and accept friend requests.

1. Followers:

* Users can follow other users without requiring mutual approval.

1. Friend List:

* Users can view their list of friends and followers.

2. User Profiles

1. User Registration:

* Users can register using an email.

1. User Login:

* Users can log in using their email and password.

1. Profile Management:

* Users can create, update, and delete their profiles.
* Profiles include some information such as name, bio, profile picture, and contact info.

1. Privacy Settings:

* Users can set their profile visibility (public, private, or friends-only).
* Users can block or report other users.

5.Search:

* Users can search for other users by name or username.

3• **Non functional requirements**

1. Performance

Response Time:

* The system should respond to user requests within 300ms for 97% of requests.

Scalability:

* The platform should support up to 30k concurrent users.
* The system should scale horizontally to handle increased load.

Throughput:

* The system should handle at least 10,000 requests per second.

2. Availability

1. Uptime:

* The platform should have 98% uptime.

1. Redundancy:

* Critical services should have redundancy and failover mechanisms.

3. Security

* Authentication and Authorization: Use OAuth 2.0 and JWT web tokens for secure authentication and authorization.

4. Reliability

1. Error Handling:

* The system should smoothly handle errors and provide meaningful error messages to users.

1. Data Consistency:

* Ensure eventual consistency across microservices using event-based communication.

5. Maintainability

1. Modular Design:

* The platform should be built using a microservices architecture for easy maintenance and updates.

6. Usability

1. User Interface:

* The platform should have a responsive user interface.

1. Accessibility:

* The platform should comply with accessibility standards like WCAG.

7. Compliance

Data Privacy:

* Comply with data privacy regulations such as GDPR and CCPA.