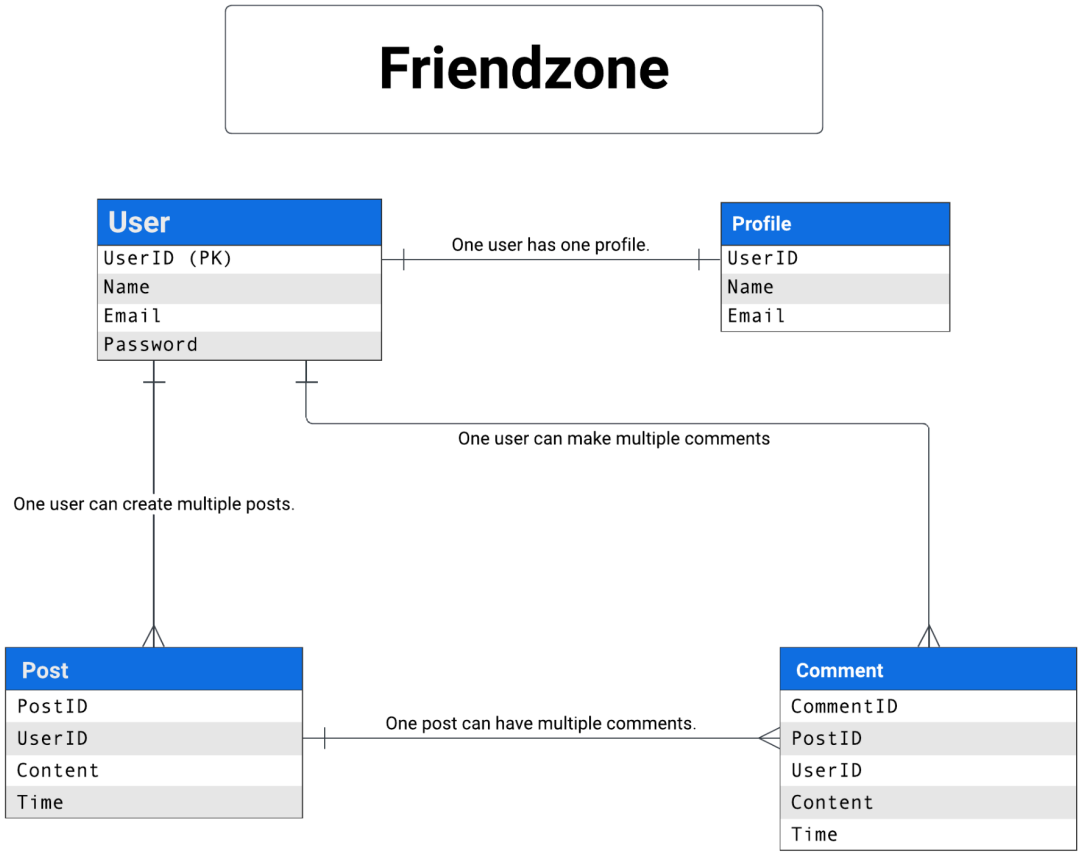
FriendZone Social Media Application

Task 1: Entity Relationship Diagram (ERD)



In this ERD:

- **User Entity:** Represents the basic information about a user, including UserID (Primary Key), Name, Email, Password.
- **Profile Entity:** Represents additional profile information for a user. It has a one-to-one relationship with the User entity, connected through the UserID.
- **Post Entity:** Represents a post made by a user. It includes PostID (Primary Key), UserID (Foreign Key referencing User), Content of the post.

- **Comment Entity**: Represents a comment made on a post. It includes CommentID (Primary Key), UserID (Foreign Key referencing User), PostID (Foreign Key referencing Post), Content of the comment.

The relationships between entities in the "**Friendzone**" social media platform:

1. User and Profile Relationship (One-to-One):
- One user can have only one profile.
- One profile is associated with only one user.

2. User and Post Relationship (One-to-Many):
- One user can create multiple posts.
- Each post is created by one user.

3. User and Comment Relationship (One-to-Many):
- One user can make multiple comments.
- Each comment is made by one user.

4. Post and Comment Relationship (One-to-Many):
- One post can have multiple comments.
- Each comment is associated with one post.

Task 2: Site Map

A site map helps visualize the structure of your website. For "Friendzone," it might look like this:

- Home
  - Sign Up
  - Log In
  - User Profile
    - Edit Profile
  - News Feed
    - Create Post
    - View Post
      - Comment on Post

Task 3: CREATE TABLE Statements

Let's create simplified CREATE TABLE statements for the mentioned entities:

```
CREATE TABLE User (
    UserID INT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL,
    Email VARCHAR(255) NOT NULL,
    Password VARCHAR(255) NOT NULL,
    -- Other fields if needed...
);

CREATE TABLE Post (
    PostID INT PRIMARY KEY,
    UserID INT,
    Content TEXT,
    Timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (UserID) REFERENCES User(UserID),
    -- Other fields if needed...
);

CREATE TABLE Comment (
    CommentID INT PRIMARY KEY,
    UserID INT,
    PostID INT,
    Content TEXT,
    Timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (UserID) REFERENCES User(UserID),
    FOREIGN KEY (PostID) REFERENCES Post(PostID),
    -- Other fields if needed...
);
```

```
-- Sample data for User entity
INSERT INTO User (UserID, Name, Email, Password) VALUES
(1, 'John Doe', 'john@example.com', 'hashed_password_1'),
(2, 'Jane Smith', 'jane@example.com', 'hashed_password_2');

-- Sample data for Post entity
INSERT INTO Post (PostID, UserID, Content) VALUES
(1, 1, 'Hello, Friendzone! This is my first post.'),
```

```
(2, 2, 'Just joined Friendzone. Excited to connect with friends!');

-- Sample data for Comment entity
INSERT INTO Comment (CommentID, UserID, PostID, Content) VALUES
(1, 2, 1, 'Welcome, John!'),
(2, 1, 2, 'Nice to have you, Jane!');

-- Note: Timestamp columns will be automatically populated with the current
timestamp.
```

Task 4: Security Reflection

Building a social media platform like "Friendzone" requires careful consideration of various security issues to ensure the protection of user data, maintain user trust, and prevent unauthorized access. In this reflection, we will delve into key security considerations and potential challenges associated with the development and maintenance of such a website.

User Authentication:
One of the primary concerns is user authentication. Ensuring secure password storage using hashing algorithms and salting is fundamental. Additionally, implementing multi-factor authentication (MFA) enhances the overall security posture, adding an extra layer of protection against unauthorized access.

Data Validation and SQL Injection:
In a social media platform where user-generated content is prevalent, preventing SQL injection attacks is critical. Robust data validation techniques, such as input validation and parameterized queries, need to be implemented to thwart malicious attempts to manipulate or extract sensitive information from the database.

Session Management:
Session management is a crucial aspect to prevent unauthorized access to user accounts. Implementing secure session handling mechanisms, including session timeouts and secure cookies, helps protect against session hijacking and unauthorized account access.

Secure Communication:
Encrypting data transmitted between users and the server using HTTPS is paramount to ensure the confidentiality and integrity of user data. Without proper

encryption, sensitive information, including login credentials and personal messages, could be susceptible to eavesdropping.

Data Privacy and Compliance:
Respecting user privacy and complying with data protection regulations, such as GDPR, is imperative. Implementing robust privacy settings that allow users to control the visibility of their posts and personal information is essential. Adequate disclosure and consent mechanisms should be in place, and users should be informed about how their data is collected, processed, and stored.

Cross-Site Scripting (XSS):
Given the interactive nature of social media platforms, preventing Cross-Site Scripting attacks is crucial. Sanitizing user inputs and encoding output help mitigate the risk of malicious scripts being injected into web pages, thereby protecting users from script-based attacks.

Security Updates:
Regularly updating and patching server software, frameworks, and libraries is a continuous effort to address security vulnerabilities. Failure to apply timely updates may expose the platform to known vulnerabilities, increasing the risk of exploitation by malicious actors.

User Authorization and Access Control:
Ensuring proper user authorization and access control mechanisms is vital to prevent unauthorized access to sensitive data. Role-based access control (RBAC) can be implemented to manage user permissions effectively, limiting access to specific functionalities based on user roles.

Backup and Recovery:
Establishing a robust backup and recovery strategy is essential for mitigating the impact of data loss or system compromise. Regularly backing up user data and having a well-defined disaster recovery plan ensures data integrity and minimizes downtime in the event of an unforeseen incident.

Monitoring and Logging:
Implementing comprehensive monitoring and logging mechanisms is crucial for detecting and responding to security incidents. Monitoring tools that analyze user activities and system behavior can help identify unusual patterns or suspicious activities, enabling a proactive response to potential threats.

In conclusion, the security considerations for a social media platform like "Friendzone" are multifaceted and require a holistic approach. Prioritizing user

authentication, data validation, secure communication, and compliance with privacy regulations is foundational. Ongoing efforts to stay informed about emerging threats, applying security updates promptly, and implementing robust monitoring mechanisms contribute to building a resilient and secure social media platform that users can trust with their personal information. Regular security audits, ethical hacking, and a culture of security awareness among development and operational teams are integral to maintaining a strong security posture throughout the lifecycle of the platform. As technology evolves, adapting security measures to address new challenges becomes an ongoing commitment to safeguarding the platform and its users.