***Writeup – CS5610 Project3***

***Collaborators: Lingyun Xiao & Xukun Zhang***

**Challenges Faced**

During the development of our project3, we faced several challenges that required careful planning and problem-solving.

Password security implementation was one of the primary challenges, as we needed to use bcrypt for encrypting passwords, ensuring accurate comparison during login, and maintaining security across user registration and updates.

Handling images in posts was another area that required effort; we initially used multer for file uploads but later switched to base64 encoding for easier integration, while also managing size limitations and optimizing UI display.

State management and user authentication were essential aspects, and we used context to manage component states like isLoggedIn to track user sessions, handle protected routes, and manage authentication flows effectively.

Designing and implementing APIs involved thorough discussions to ensure they met the application requirements; for instance, the API localhost:5001/api/posts/:username was designed to fetch posts by a specific user.

Additionally, deploying the application on Render posed its own set of challenges, particularly in configuring environments and ensuring smooth functionality.

**Potential Improvements**

There are several enhancements we would like to introduce in the future. Improving image handling features is a priority, including implementing compression before storage, supporting multiple images per post, and adding cropping or editing tools.

User experience can also be elevated by introducing features like user mentions, notifications, post sharing, real-time updates, and pagination or infinite scroll for better navigation.

We also plan to add new functionalities, such as a user following system, post likes and comments, user search capabilities, and categorization of posts with hashtags. These improvements aim to make the application more engaging and user-friendly.

**Assumptions Made**

We made several assumptions to guide our development process. We assumed that most users would access the application through desktop browsers, upload appropriately sized images, and prefer a straightforward, easy-to-use interface.

On the technical side, we assumed that MongoDB would be sufficient for storing user data and posts, and that base 64 encoding would adequately handle image storage for our use case.

Additionally, we did not prioritize advanced security measures beyond JWT for user sessions, given the scope of the project. From a performance perspective, we anticipated that the application would handle a moderate number of users and posts, with reasonable image sizes and efficient database queries without requiring extensive optimization.

**Time to Complete**

The entire project was completed within approximately one week. This timeline included planning, coding, testing, and deployment, requiring focused collaboration and efficient use of time to deliver a functional application.