### **Reflection Document 📝**

### **Project Summary**

The **User Management System** project was an enriching opportunity to develop, test, and resolve critical functionalities for a robust user management application. My role involved identifying and fixing bugs, implementing new features, and ensuring the stability of the system.

### **Bugs and Fixes**

1. **Duplicate Email/Nickname Validation During User Update**
   * **Issue**: Users were able to update their profiles with duplicate emails or nicknames, leading to conflicts.
   * **Fix**: Implemented database-level constraints and application-level validation checks to prevent duplicates.
   * **GitHub Issue**: [#10](https://github.com/Yaminichv/user_management/issues/10)
2. **Email Verification Fails Due to None User-ID**
   * **Issue**: During email verification, None values for user-ID caused failures.
   * **Fix**: Added a check to validate user-IDs before sending email verification.
   * **GitHub Issue**: [#8](https://github.com/Yaminichv/user_management/issues/8)
3. **Password Strength Validation**
   * **Issue**: Weak passwords were allowed during user creation.
   * **Fix**: Added validation rules to enforce strong passwords with uppercase, lowercase, numbers, and special characters.
   * **GitHub Issue**: [#6](https://github.com/Yaminichv/user_management/issues/6)
4. **Role Downgrade During Email Verification for Non-Anonymous Users**
   * **Issue**: Users' roles were being unintentionally downgraded during the email verification process.
   * **Fix**: Implemented checks to preserve user roles while performing email verification.
   * **GitHub Issue**: [#12](https://github.com/Yaminichv/user_management/issues/12)
5. **Docker Build Issue**
   * **Issue**: Docker builds were failing due to configuration issues in the docker-compose.yml file.
   * **Fix**: Corrected the docker-compose configurations and ensured proper service linking.
   * **GitHub Issue**: [#1](https://github.com/Yaminichv/user_management/issues/1)

### **Feature Implementation**

**User Search and Filtering**I implemented a **User Search and Filtering** feature that allows admins to search users based on attributes like name and email. Additionally, filtering options were added to streamline user management.

* This feature enhances the system's usability and efficiency for administrators.
* **GitHub Issue**: [#14](https://github.com/Yaminichv/user_management/issues/14)

### **Testing and QA**

To improve test coverage, I created unit tests to validate key functionalities:

* Duplicate email detection
* Password validation checks
* User search and filtering

These tests ensure the application's reliability and robustness by covering edge cases and key scenarios.

### **Key Learnings**

1. **Bug Resolution**: I gained experience identifying and resolving critical bugs, improving the system's reliability.
2. **Feature Development**: Implementing a new feature required following best practices for clean coding, testing, and documentation.
3. **Testing**: Writing unit tests helped me understand the importance of test coverage and handling edge cases.
4. **CI/CD with Docker**: I enhanced my skills in deploying projects using Docker and ensuring smooth CI/CD pipelines.

### **Final Thoughts**

This project provided me with real-world experience in developing and maintaining a complex system. Working on bug fixes, implementing features, and writing tests has honed my software engineering skills, preparing me for industry-level challenges.

### **Project Links**

* **GitHub Repository**: [User Management System](https://github.com/Yaminichv/user_management)
* **Closed Issues**:
  + #14: User Search and Filtering
  + #10: Duplicate email/nickname validation during user update
  + #8: Email verification fails due to user-id being None
  + #6: Validate password strength while user creation
* **DockerHub**: [User Management Docker Image](https://hub.docker.com/r/yaminichowdary996/user_management)