### **Reflection Document 📝**

**Project Summary**The **User Management System** project was a highly rewarding experience, allowing me to identify and resolve critical bugs, implement new functionalities, and improve system reliability. My work spanned fixing multiple issues, ensuring comprehensive testing, and enhancing the user experience with new features.

### **Project Links**

● **GitHub Repository**: [Github Repo](https://github.com/Roman-sv966/user_management): https://github.com/Roman-sv966/user\_management

● **Closed Issues**:

○ [#11](https://github.com/Roman-sv966/user_management/issues/11): User Profile Management with Enhanced Update

○ [#9](https://github.com/Roman-sv966/user_management/issues/9): Missing is\_professional Field Fix

○ [#7](https://github.com/Roman-sv966/user_management/issues/7): User-ID None Fix During Email Verification

○ [#5](https://github.com/Roman-sv966/user_management/issues/5): Role Assignment Issue Fix

○ [#3](https://github.com/Roman-sv966/user_management/issues/3): FastAPI-Starlette Dependency Conflict

○ [#2](https://github.com/Roman-sv966/user_management/issues/2): Critical Vulnerability Fix in Dependencies

○ [#1](https://github.com/Roman-sv966/user_management/issues/1): Docker Build Failure Fix

● **DockerHub**: <https://hub.docker.com/repository/docker/sv966/user_management/general>

### **Bugs and Fixes**

1. **User Profile Management with Enhanced Profile Update**

○ **Issue**: Users could not update their profile information comprehensively, including fields like is\_professional and profile pictures.

○ **Resolution**: Implemented a robust update functionality that supports enhanced user profile updates.

○ **GitHub Issue**: [#11](https://github.com/Roman-sv966/user_management/issues/11)

1. **is\_professional Field Not Included in Schema**

○ **Issue**: The is\_professional field was missing from the user schema and the response object, causing incomplete data retrieval.

○ **Resolution**: Updated the schema to include the is\_professional field, ensuring it is validated and returned in API responses.

○ **GitHub Issue**: [#9](https://github.com/Roman-sv966/user_management/issues/9)

1. **User-ID is None During Email Verification**

○ **Issue**: Email verification failed when the user-id was None.

○ **Resolution**: Added checks to validate the user ID before proceeding with email verification to prevent failures.

○ **GitHub Issue**: [#7](https://github.com/Roman-sv966/user_management/issues/7)

1. **Incorrect Role Assignment During Admin Email Verification**

○ **Issue**: Roles were being unintentionally downgraded during email verification, causing privilege loss for certain users.

○ **Resolution**: Ensured roles remain preserved during verification by adding validation logic.

○ **GitHub Issue**: [#5](https://github.com/Roman-sv966/user_management/issues/5)

1. **FastAPI-Starlette Dependency Conflict**

○ **Issue**: FastAPI and Starlette had a version conflict, leading to application crashes during runtime.

○ **Resolution**: Upgraded both dependencies to compatible versions to resolve conflicts and ensure stability.

○ **GitHub Issue**: [#3](https://github.com/Roman-sv966/user_management/issues/3)

1. **High and Critical Vulnerabilities in Python Dependencies**

○ **Issue**: Several critical vulnerabilities were identified in the application's Python dependencies, posing security risks.

○ **Resolution**: Upgraded dependencies to their latest, secure versions to eliminate vulnerabilities and ensure system security.

○ **GitHub Issue**: [#2](https://github.com/Roman-sv966/user_management/issues/2)

1. **Docker Build Failure for Specific libc-bin Version**

○ **Issue**: Docker builds were failing due to missing or incompatible libc-bin package versions.

○ **Resolution**: Corrected the Docker configuration to explicitly install the required version of libc-bin, ensuring successful builds.

○ **GitHub Issue**: [#1](https://github.com/Roman-sv966/user_management/issues/1)

### **Feature Implementation**

**User Profile Management with Enhanced Update Functionality**I implemented a **User Profile Management** feature that enhances profile updates. Admins and users can now update additional fields such as profile pictures, professional status (is\_professional), and other attributes seamlessly.

● This feature improves user experience and makes profile management more comprehensive.

● **GitHub Issue**: [#11](https://github.com/Roman-sv966/user_management/issues/11)

### **Testing and QA**

To ensure the reliability and robustness of the system, I implemented unit tests for the following functionalities:

● Validation of is\_professional schema field.

● Role preservation during email verification.

● Enhanced profile updates.

● Dependency conflict resolution.

● Email verification error handling.

These tests significantly improved test coverage and ensured that edge cases and failure scenarios were addressed effectively.

### **Key Learnings**

1. **Bug Resolution**: I gained experience in identifying, debugging, and fixing critical issues, ensuring system stability and reliability.
2. **Feature Development**: Implementing new functionality allowed me to practice clean code principles, schema validation, and API enhancements.
3. **Dependency Management**: Resolving conflicts and upgrading dependencies taught me the importance of keeping libraries up to date for security and stability.
4. **Docker and CI/CD**: Debugging Docker build failures helped me understand containerization and deploying applications in production-like environments.
5. **Testing**: Writing unit tests reinforced my understanding of testing frameworks and the importance of validating both success and failure scenarios.

### **Future Enhancements 🚀**

1. **Dynamic Profile Picture Editing**: Enable users to crop, rotate, and adjust images before uploading.
2. **Audit Logs for Profile Updates**: Implement logging to track changes made to user profiles for accountability.
3. **Advanced Role-Based Permissions**: Allow granular control over which user roles can modify specific profile fields.
4. **Profile Picture Versioning**: Maintain a version history of uploaded profile pictures, allowing rollback to previous versions.
5. **Real-Time Email Verification Status**: Display live updates on email verification status for improved user feedback.

### **Final Thoughts**

The project provided me with hands-on experience in resolving real-world issues, implementing robust features, and ensuring quality through thorough testing. By addressing critical bugs and adding enhancements, I contributed to building a more reliable and user-friendly system. This experience has strengthened my problem-solving, coding, and testing skills, preparing me to tackle complex challenges in the software development industry.