## **1. Project Overview 🚀**

The **User Management System** was a challenging yet rewarding experience where I worked on fixing bugs, enhancing features, and improving the system's overall performance. My contributions ranged from resolving critical issues like validation errors and build failures to implementing advanced features such as the **Profile Picture Upload** with MinIO integration.This project has given me hands-on exposure to debugging, testing, and ensuring high-quality deliverables that align with real-world development standards.

## **Project Links 🔗**

* **GitHub Repository**: [User Management](https://github.com/pendekantiakhil/user_management)- Final Project
* **Closed Issues**:
  + [#18](https://github.com/pendekantiakhil/user_management/issues/18): Error in Image Uploader
  + [#15](https://github.com/pendekantiakhil/user_management/issues/15): Duplicate Emails/Nicknames
  + [#12](https://github.com/pendekantiakhil/user_management/issues/12): Password Strength Not Validated
  + [#9](https://github.com/pendekantiakhil/user_management/issues/9): Profile Picture URL Validation
  + [#7](https://github.com/pendekantiakhil/user_management/issues/7): Prevent Role Downgrade
  + [#5](https://github.com/pendekantiakhil/user_management/issues/5): User-ID Missing in Verification
  + [#1](https://github.com/pendekantiakhil/user_management/issues/1): Docker Dependencies Issue
* **DockerHub**: [Link to Docker Image](https://hub.docker.com/repository/docker/vp679/user_management/general)

## **2. Bugs Fixed 🛠️**

| **Bug ID** | **Issue** | **Fix Implemented** |
| --- | --- | --- |
| **#18** | **Error in Image Uploader** | Fixed file handling logic to validate and process uploads seamlessly. |
| **#15** | **Duplicate Emails/Nicknames** | Added database constraints and application checks to prevent duplicates. |
| **#12** | **Password Strength Not Validated** | Enforced strong password rules: uppercase, lowercase, numbers, and special characters. |
| **#9** | **Profile Picture URL Validation** | Implemented regex-based URL validation for profile picture links. |
| **#7** | **Unauthorized Role Downgrade** | Preserved user roles during the email verification process. |
| **#5** | **Missing User-ID on Verification** | Ensured user-ID is correctly passed in email verification links. |
| **#1** | **Docker Image Dependencies Issue** | Fixed dependency conflicts in the Dockerfile for successful builds. |

### **Takeaways**

By fixing these bugs, I ensured smoother workflows, enhanced application reliability, and improved the user experience.

## **3. Feature Development: Enhanced Profile Picture Management 🖼️**

### **Description**

Implemented a feature to allow users to **upload profile pictures** seamlessly, storing them securely in **MinIO**.

### **Core Functionality**

* **File Validation**: Only .jpg, .jpeg, and .png file types are accepted.
* **Image Optimization**: Resized and optimized images for consistent performance.
* **Secure Storage**: Uploaded files are stored securely with retrieval links for user profiles.

### **Impact**

This feature improves user personalization and application usability, ensuring a robust and scalable solution for image uploads.

* **GitHub Issue**: [#17](https://github.com/pendekantiakhil/user_management/issues/17)

## **4. Testing and Quality Assurance 🔍**

To ensure stability and robustness of the application:

* Created **unit tests** for:
  + Password strength validation.
  + Duplicate email/nickname checks.
  + Profile picture upload scenarios.
* Enhanced test cases for **bulk user creation** to cover edge cases.

This resulted in improved **test coverage** and confidence in the application's functionality.

## **5. Key Learnings 🎓**

1. **Bug Resolution**: Gained expertise in analyzing and resolving system-critical issues with precision.
2. **Feature Integration**: Learned how to integrate third-party tools (like **MinIO**) effectively.
3. **Robust Testing**: Understood the significance of thorough test coverage to prevent regressions.
4. **Docker Proficiency**: Improved my skills in resolving Docker build issues and maintaining CI/CD pipelines.
5. **Code Optimization**: Followed clean coding practices to make the system scalable and maintainable.

### **6. Future Enhancements 🚀**

1. **Dynamic Image Editing**: Allow users to crop, rotate, and preview images before uploading.
2. **Automatic Compression**: Reduce file size while maintaining image quality for optimized performance.
3. **Version Control**: Enable rollback to previous profile pictures with version history.
4. **Security Scanning**: Integrate malware detection to ensure safe and clean uploads.
5. **CDN Integration**: Deliver images faster globally using Content Delivery Networks (CDN).

## **7. Challenges Faced & Overcome 🧩**

* **Handling Duplicate Issues**: Identifying root causes and resolving conflicts in bulk user creation.
* **Image Upload Integration**: Ensuring secure storage while optimizing images for better performance.
* **Docker Dependency Fixes**: Debugged and resolved version conflicts, ensuring smooth containerized builds.

Each challenge strengthened my problem-solving skills and ability to work on real-world software complexities.

## **8. Final Reflection 💭**

This project was an excellent opportunity to apply my technical knowledge to a **real-world application**. Resolving bugs, implementing a feature-rich profile management system, and ensuring application reliability have improved my technical capabilities and teamwork skills.