Jonathan Payes

Professor William Keith

IS218-450

August 11, 2025

**Reflection on Learning and Project Experience**

Throughout this project, I had a tremendous amount of on-the-job experience in all these areas, including software quality assurance, software testing, and collaborative software development process, which are directly relevant to those of the real-world production quality standards. The use and introduction of modern tools like PostgreSQL, GitHub Actions, and containerized deployment through DockerHub, in a Dockerized FastAPI application, was one of the most worthwhile outputs that could come out of this journey. The project provided me with the experience of being a quality assurance analyst and a developer in the same capacity, validating some functionality on the project, finding gaps, writing tests, and finally, delivering rock-steady code reached via systematic workflows.

With the aim of improving test coverage, I prioritized the task of crafting and adding up a total of 10-unit + tests that were well comprehensive and significant. These tests included edge scenarios like incorrect token formats, expired tokens, unprivileged privilege escalation, data persistence, and any violation of the boundaries like long authentication. Every test was carefully made to ensure that the application acts in a proper way in expected and unexpected situations. An example would be that I wrote a test to make sure a user could not promote them self to admin- a very important test to make sure there is no role breaking. The other test was that once a user changed their email it would be the same across queries in the database mimicking the conditions in the real world demanded by data consistency. They created these tests in the form of pytest, which in turn integrated into the CI/CD pipeline, thus giving them instant feedback on regressions.

Regarding the features that were being implemented, I created and developed a fully operational profile picture upload endpoint. This entailed the incorporation of object storage system MinIO, a secure file uploading structure with the construction of the UploadFile, content type verification, and storage of user metadata in the right place. The REST-based feature was well-validated and error-proofed, was fully documented together with test coverage. This not only brought real features to the app but also proved the capability to develop secure, scalable, and user-facing elements. A major emphasis on collaboration was present in the project. Every major activity such as test, bug fix, or feature development was managed as GitHub Issue. I relied on the best habits by forking original and committing with descriptive messages and opening pull requests in association with relevant issues. I also engaged in code reviews and followed the guidelines of contribution such as the use of linters, making commits atomic, ensuring testability of the code and readability.

This created a lifelike development setting, and it gave me a lesson on the importance of effective communication and responsibility in software development when it comes to utilizing groups. In addition to having a technical implementation experience, the project has got me ready in terms of the industry. I now am familiar not only with code writing but software delivery through maintainable tested, and documented code. I used such tools as Docker, GitHub Actions, and dependency injection frameworks, which are essential in DevOps and full-stack jobs nowadays. This process was undertaken by deploying the project on the DockerHub solidifying the need to work within reproducible build environments and the container.

In general, this experience boosted my professional growth practices and confidence in coding. My technical and soft skills have been enhanced because I was able to solve bugs, afford greater test coverage, install a new feature and act in an engineering manner in a production environment. This journey passes in my GitHub activity now because it has a clean commit history, related issues, working CI/CD, and a live docker image. I have achieved something of which I am proud but more to the point I am ready to do something new.

Docker Hub:

<https://hub.docker.com/repository/docker/jap323/user_management/general>