

Final Project

Student Name : Kiran Ramiseti.

Student UCID : kr483.

Project Name : USER MANAGEMENT.

Course: IS601002-WEB SYSTEMS DEVELOPMENT.

Instructor Name: Keith Williams.

1. What I learnt in this course.

In this course, I've acquired a wealth of knowledge and practical skills that are crucial for modern software development. Here's a brief overview of what I learned:

a. Docker.

I learned how to use Docker, which has revolutionized deployment by allowing me to package my application and its environment into a container that can run anywhere, ensuring consistency across different development and production environments.

b. Nginx.

I learned how to use Docker, which has evolutionized deployment by allowing me to package my application and its environment into a container that can run anywhere, ensuring consistency across different development and production environments.

c. Git.

I deepened my understanding of Git for version control, which is essential for managing changes to project codebases, collaborating with others, and maintaining a robust workflow in software projects.

d. Python.

I enhanced my Python skills, focusing on writing clean, well-documented code that follows industry standards. Python's versatility was demonstrated through various applications, from web development to data analysis.

e. Testing Using Python.

I learned how to write reliable tests using Python frameworks like PyTest, which help ensure my code works as intended before deployment, reducing bugs, and improving code quality.

f. Github Actions.

I explored how to automate workflows with GitHub Actions to run tests, lint code, and manage deployments directly from GitHub repositories, which streamlined my development process and enhanced collaboration.

g. Dockerhub.

I learned to use DockerHub for storing and managing Docker images. It enabled me to share images with my team, access pre-built images from others, and manage version control of images, which simplified deployments.

h. APIs.

delved into API development, which is fundamental for building modern software applications. I learned how APIs facilitate communication between different software components and services, allowing systems to exchange data seamlessly. This knowledge proved invaluable for integrating third-party services and for creating robust, scalable back-end systems that can communicate effectively with front-end interfaces.

i. Minio.

I learned to use Minio, an open-source object storage server compatible with Amazon S3 APIs. Minio has enabled me to efficiently handle storage and retrieval of large amounts of unstructured data, such as photos, videos, and log files. I gained practical experience in setting up Minio, securing access to data, and using it in applications for reliable, scalable storage solutions.

j. REPL

Great! Learning about REPL (Read-Eval-Print Loop) is another valuable addition. REPL is an interactive programming environment that allows you to write code and execute it to see results immediately, which is incredibly useful for learning, debugging, and experimenting with new ideas. Here's how this knowledge has enhanced my skill set:

➤ Immediate Feedback.

Using REPL environments, such as Python's IDLE or the interactive shells for languages like Node.js, has allowed me to get immediate feedback on code snippets, making it easier to understand how different functions and operations work.

➤ **Experimentation.**

REPL is perfect for experimenting with new libraries or language features, testing small pieces of code, or troubleshooting specific functionalities before integrating them into a larger project.

➤ **Enhanced Debugging.**

It has improved my debugging skills by allowing me to isolate problematic sections of code and test solutions interactively, reducing the time it takes to find and fix bugs.

➤ **Learning and Teaching Aid.**

REPL has been an excellent tool for reinforcing concepts learned in class by allowing me to practice code in real-time and see the effects of changes immediately, which has also been beneficial when helping peers understand coding concepts.

2. Problems faced while doing this project.

a. Editing produciton.yml file.

Production.yml file had issues that were causing the github actions to fail. I changed github username and docker image name. I added GITHUB_USERNAME and GITHUB_TOKEN that helped in resolving the issues successfully.

link : https://github.com/Kiran-Ramisetti-kr483/user_management/pull/2/files

b. Email verification issue

The user verification structure was not in order. So, I had to change order of few lines of code and added few test cases to check if the email verification was working well or not.

link : https://github.com/Kiran-Ramisetti-kr483/user_management/pull/5/files

c. Creating the auth_routes.py

created a new file auth_routes.py to keep codebase clean and maintainable. In user_routes.py we handle routes related to user profile management like creating, updating, retrieving and deleting user profiles. In auth_routes.py we manage routes related to authentication such as login, logout, and password management.

link : https://github.com/Kiran-Ramisetti-kr483/user_management/pull/7/files

d. duplicate email and nickname issue.

In this issue I notice that few test cases were passing even when the test case was failing related to error code 400 and saw few issues with duplicate emails and nickname. I had to update new routes in `user_routes.py` and `user_service.py`, `common.py`, `test_users_api.py` and `test_user_service.py`.

link : https://github.com/Kiran-Ramisetti-kr483/user_management/pull/9/files

e. Issue with unverified users and locked users.

There were few issues with verification process for a new user and locked user. I updated code in `user_routes.py` and `user_service.py`. I added new test cases in `test_users_api.py` to ensure that the verification process for new user is working properly.

link : https://github.com/Kiran-Ramisetti-kr483/user_management/pull/11/files

f. Password validation.

I updated the code in `user_schemas.py` to strictly validate the password and added few lines of code in `user_routes.py`. I even added few test cases to in `test_user_api.py` and `test_user_schemas.py` to verify validation of password.

link : https://github.com/Kiran-Ramisetti-kr483/user_management/pull/16/files

3. New Feature - upload pictures using minio.

To be able to upload and update profile picture to personalize the user account. For this I implemented an API endpoint for users to upload their profile picture, store the uploaded profile pictures securely in Minio, updating the user profile API endpoints to include the profile picture URL and retrieving the profile picture URL from Minio when displaying user profiles.

link : https://github.com/Kiran-Ramisetti-kr483/user_management/pull/20/files

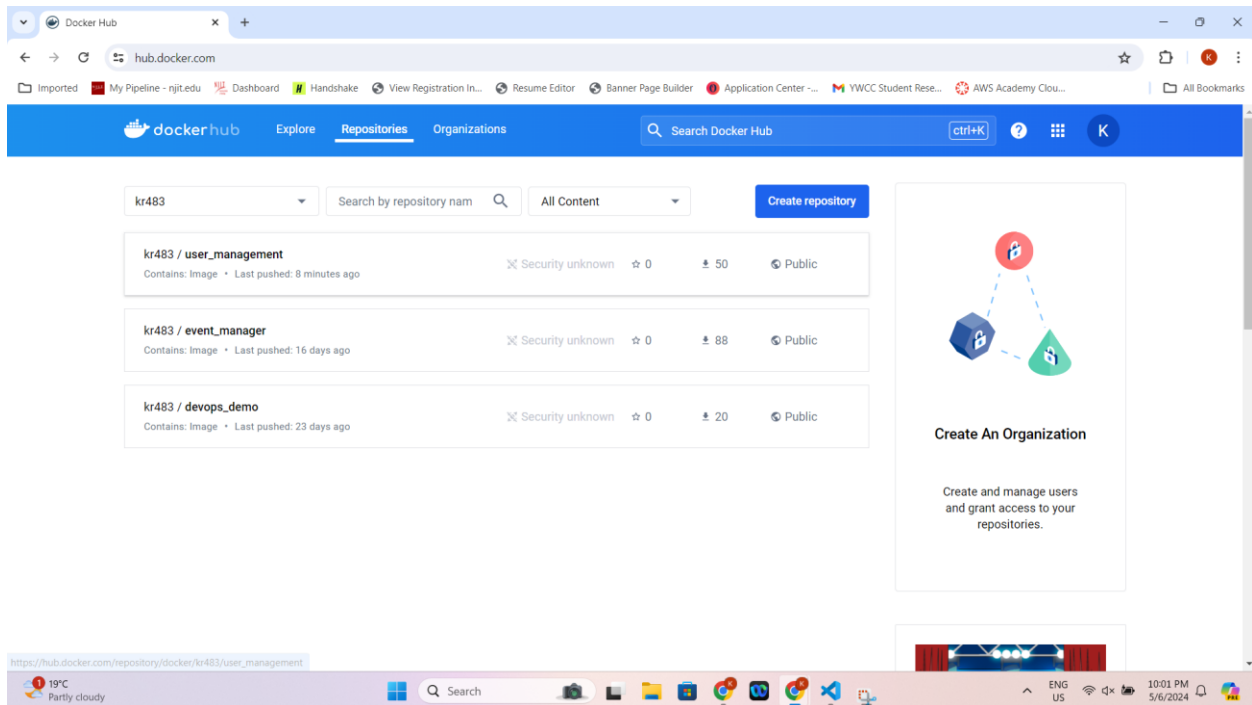
4. Test cases.

In total I added 19 test cases to resolve few issues and test if those issues are resolved or not. The above links in Problems faced while doing this project have the new test cases I added into the project.

5. Dockerhub link and Image.

The project is working well and the docker image is being saved in the dockerhub. The link and image of my dockerhub repositories are give below:

link : <https://hub.docker.com/repositories/kr483>



These skills ensure that I am well-equipped to handle advanced software projects, contribute to high-performance teams, and adapt to emerging technologies. They make me a strong candidate for numerous roles in the tech sector, from software development to systems engineering and beyond. Each skill not only enhances my technical proficiency but also demonstrates a capacity for continual learning and improvement—key traits for career advancement in technology.