IP : http://18.220.47.199/

The application is now publicly accessible at http://18.220.47.199/, where the functionality mirrors that of the local environment. The deployment also enables me to revisit past projects, such as my profile and the inventory app, maintaining a continuity of work and an expanding portfolio.

Github : https://github.com/ICSI-518-Software-Engineering/web-development-and-deployment-sai2615.git

Throughout the development of my MERN stack application, ChatGPT served as a competent assistant during the development process. Meanwhile, it was my prior experience that ultimately ensured the project's functionality on the EC2 instance.While I possess a solid foundation in full-stack development, Docker's usage occasionally required a deeper understanding. ChatGPT stepped in to bridge this knowledge gap, providing guidance and clarifications whenever I encountered ambiguous documentation or complex Docker behaviors. The Nginx configuration was a particular area where ChatGPT's insights were beneficial. It helped me grasp the concept of a reverse proxy setup, which is critical in directing traffic efficiently to the web service. With its aid, I configured Nginx to listen on port 80 and successfully proxy requests to the backend service.

Using the command prompt, Docker Compose was instrumental in managing the application's lifecycle. ChatGPT helped me comprehend the importance of taking containers down to maintain a clean slate, which is especially vital during the development phase to avoid lingering states. Building the images from the Dockerfiles was seamless, with ChatGPT aiding in understanding the build process and the significance of each step. The images for the frontend, backend, and MongoDB were built and run locally, allowing for immediate interaction with the application—registering, logging in, and accessing details were all accomplished on the localhost.

When it came time to deploy the application, I took the initiative, creating an EC2 instance and using PowerShell for remote access. With past experiences guiding me, I built the Docker images from scratch, deliberately avoiding the cache to ensure all layers were updated. This approach was crucial to integrate the latest changes and to avoid the pitfalls of stale configurations.

ChatGPT, while not directly assisting with the EC2 deployment, had laid a strong foundation by clarifying the Docker concepts necessary for this step. Consequently, I deployed the services in detached mode using Docker Compose, which included the Nginx-served frontend, the Node.js backend, and the MongoDB database. Despite its strengths, there were areas where ChatGPT fell short, such as adapting the Docker setup for the EC2 instance environment. Nevertheless, its assistance was invaluable in reading and interpreting documentation, understanding Docker configurations, and explaining the purpose behind each Dockerfile directive and Compose specification.

**Chatgpt Rating: 3/4**