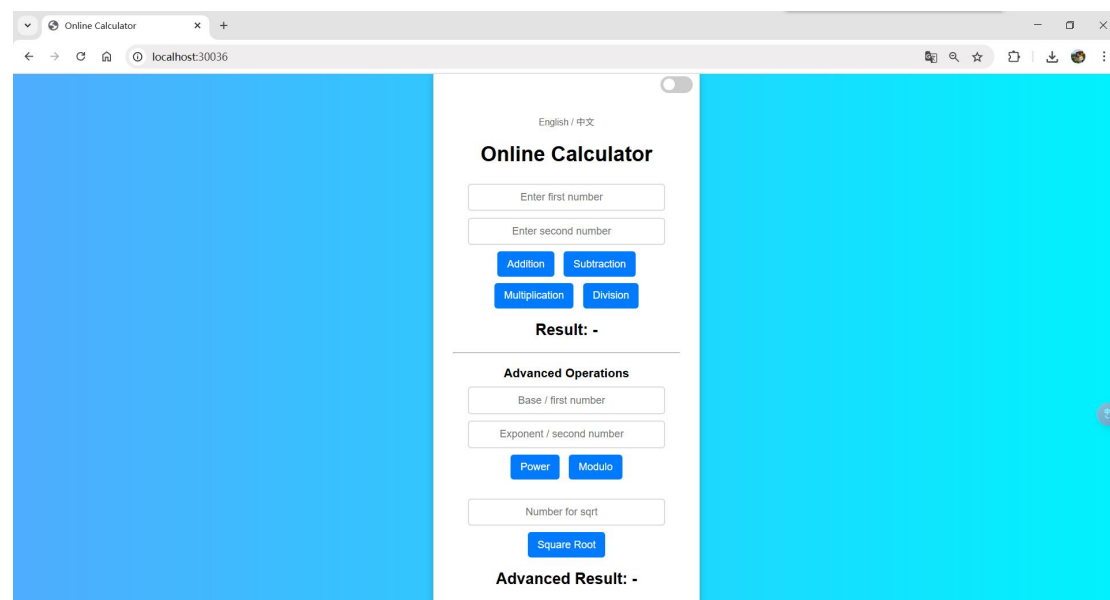


At the beginning of this phase, I deployed the previously containerised bilingual calculator application to a local Kubernetes cluster. The application was already packaged as a Docker image and capable of supporting core arithmetic operations including addition, subtraction, multiplication, division, exponentiation, square root, and modulo. It also featured a bilingual user interface implemented using HTML, CSS, and JavaScript, served by a Node.js and Express backend.

To begin the Kubernetes deployment, I created a deployment.yaml file that defined a Deployment resource. This specified one replica pod running the calculator container based on the image lonelydm/deakin-sit323-calculator:latest. The deployment configuration ensured the container exposed port 3000 and used an imagePullPolicy of Always to guarantee the latest version was used upon restart.

To allow external access to the service, I also created a service.yaml file defining a NodePort service. This mapped internal port 3000 to external port 30036, enabling the application to be accessed via <http://localhost:30036>.

Once both YAML files were applied using kubectl, I verified that the application was running correctly by checking the pod and service statuses. I was then able to access the user interface in both English and Chinese, and successfully tested all REST API endpoints through a browser and Postman. The deployed system behaved identically to the local Docker Compose version, but in a production-style Kubernetes-managed environment.



GitHub Link: <https://github.com/Lonely-DM/SIT323/tree/main/6.1P/sit323-2025-prac6p>