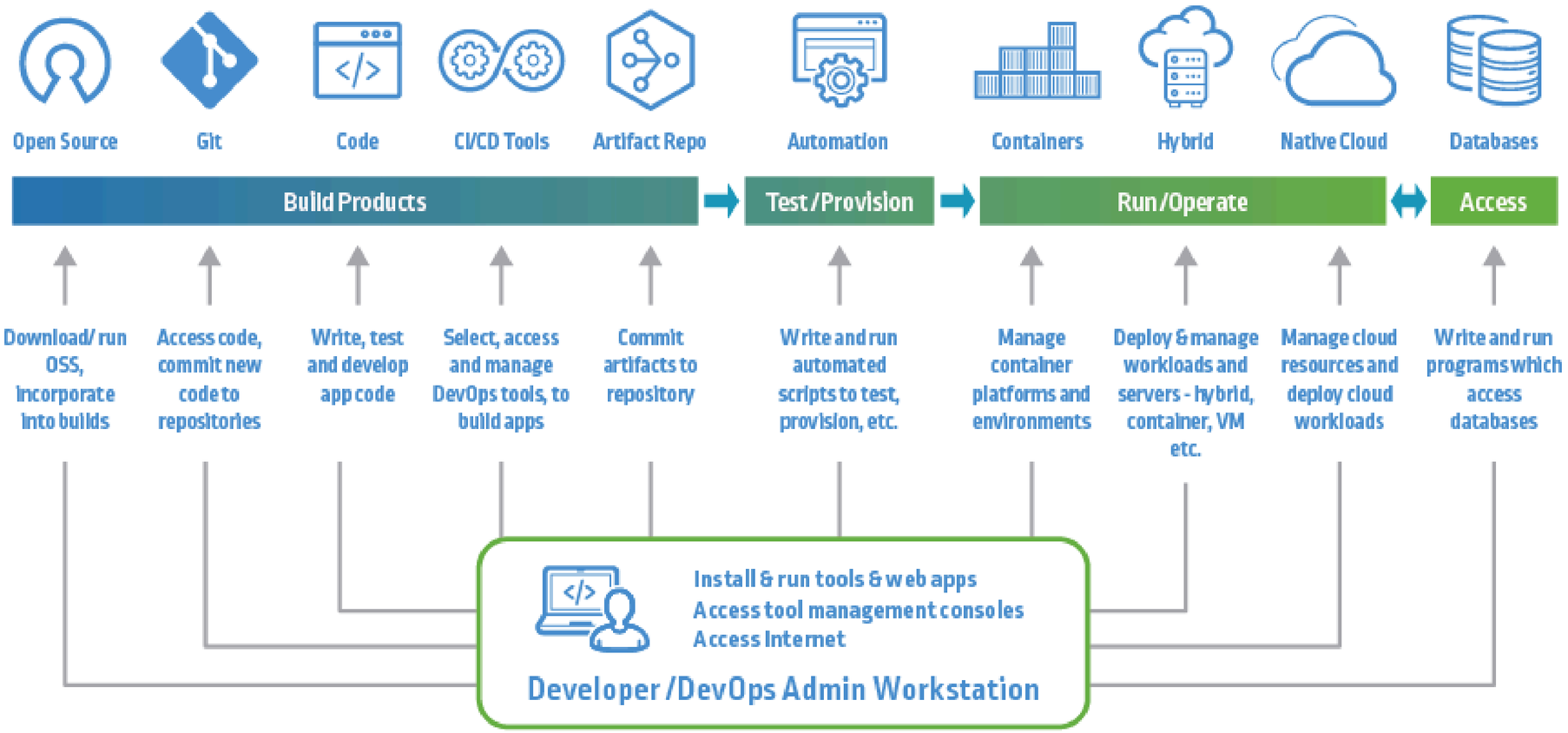
# How to Elevate Development

## Using an Impactful CI/CD Pipeline for Flutter Web Applications

Presented by Sydney Godwin for CSC 325

# What is a CI/CD Pipeline?

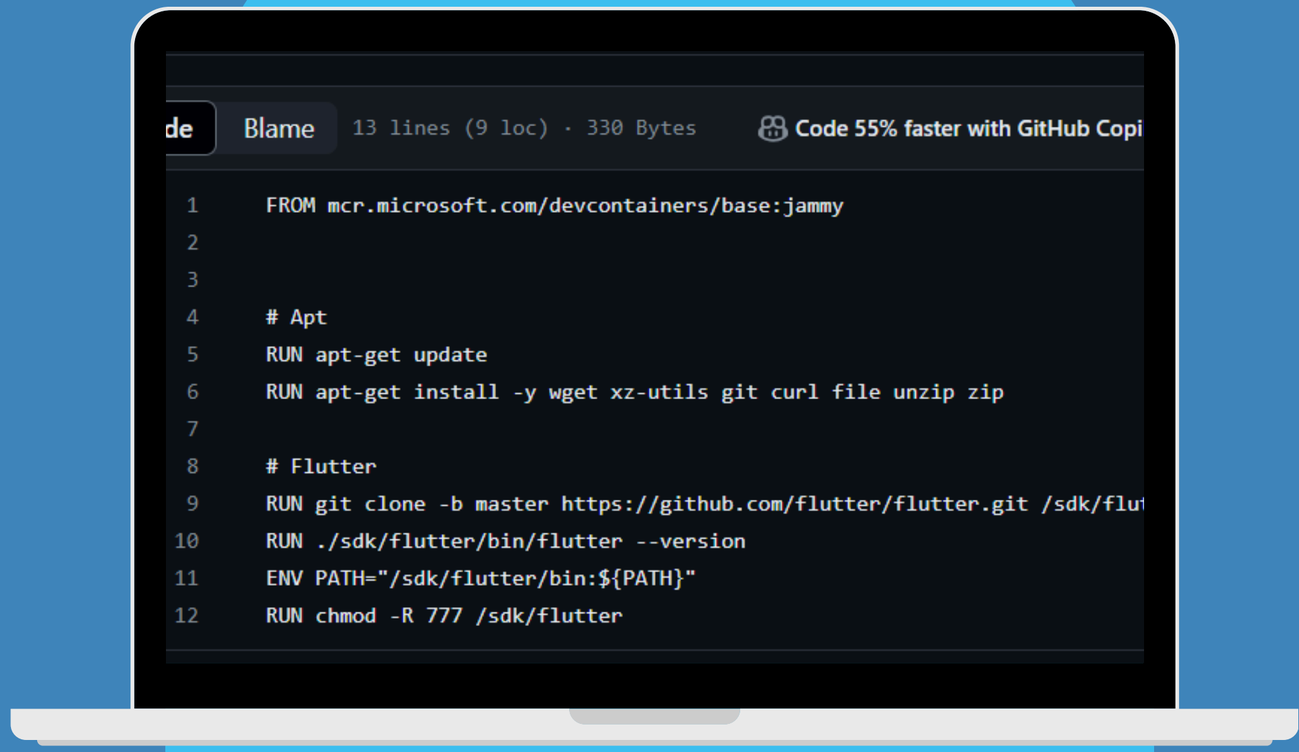



("CI/CD Pipeline," n.d.)

# Exploring the Efficiency of DevContainer Environments

***"But it works on my machine?!"***

- Leveraged Docker & dev containers for optimized Flutter dev environment
- Utilized a Dockerfile to encapsulate setup
- Dockerfile ensured consistency & efficiency
- Seamlessly integrated dev environment, simplifying CI/testing
- DevContainers enabled easy sharing & reuse
- Enhanced productivity & proficiency in Docker




**My Projects Dockerfile**

# Harnessing Git and GitHub for Version Control Integration

## Git

- Utilized Git for version control due to its flexibility and robust features
  - Experience with Git and familiarity with its functionalities like branching, merging, and commit history

## GitHub

- Leveraged GitHub as the platform for hosting repositories
  - Collaborative tools such as pull requests, code reviews, and issue-tracking

# Configuring a Seamless CI/CD Pipeline for Flutter

## Configuration Overview

- Utilized GitHub Actions to orchestrate the CI/CD pipeline
- Triggered on push events to the main branch for automated builds and deployments

## Streamlined Workflow

- Installed and checked Flutter app dependencies to ensure consistency
- Automated building and testing of Flutter web application within the pipeline

## Efficient Deployment

- Leveraged GitHub Pages to deploy the Flutter web application
- Utilized GitHub Actions for seamless integration and deployment

# My Flutter Application Deployment Strategy

Deployment Approach:

- Utilized GitHub Pages to host the Flutter application using peaceiris/actions-gh-pages@v3.

- Implemented a CI/CD pipeline to automate the deployment process.

Understanding Deployment Environments:

- Leveraged GitHub Pages to create a static website, ideal for hosting Flutter applications.

- Demonstrated proficiency in navigating various deployment environments, from local development to production.

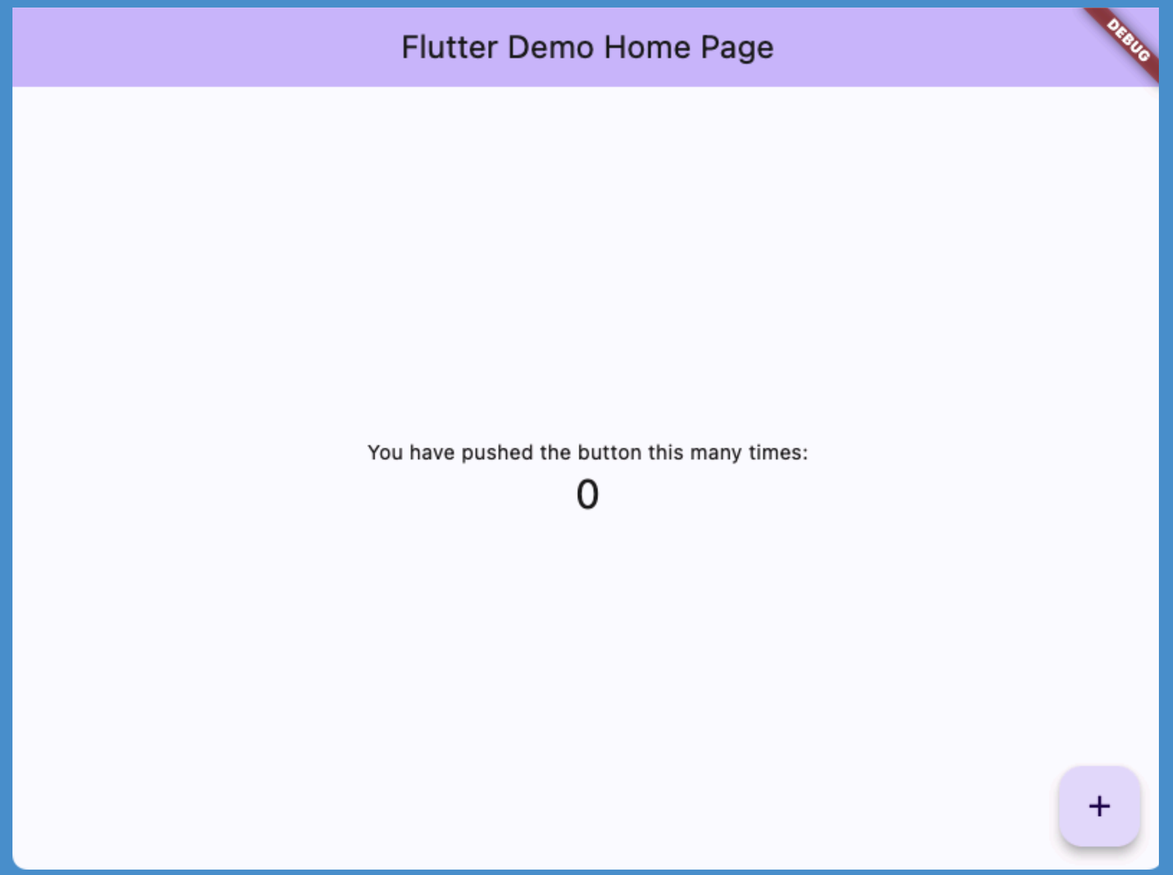
# My Flutter App

***It's just the demo flutter app***

Keeping it basic gave me time to focus on the CI/CD pipeline

I didn't have to worry about learning Dart and the learning curve associated with it

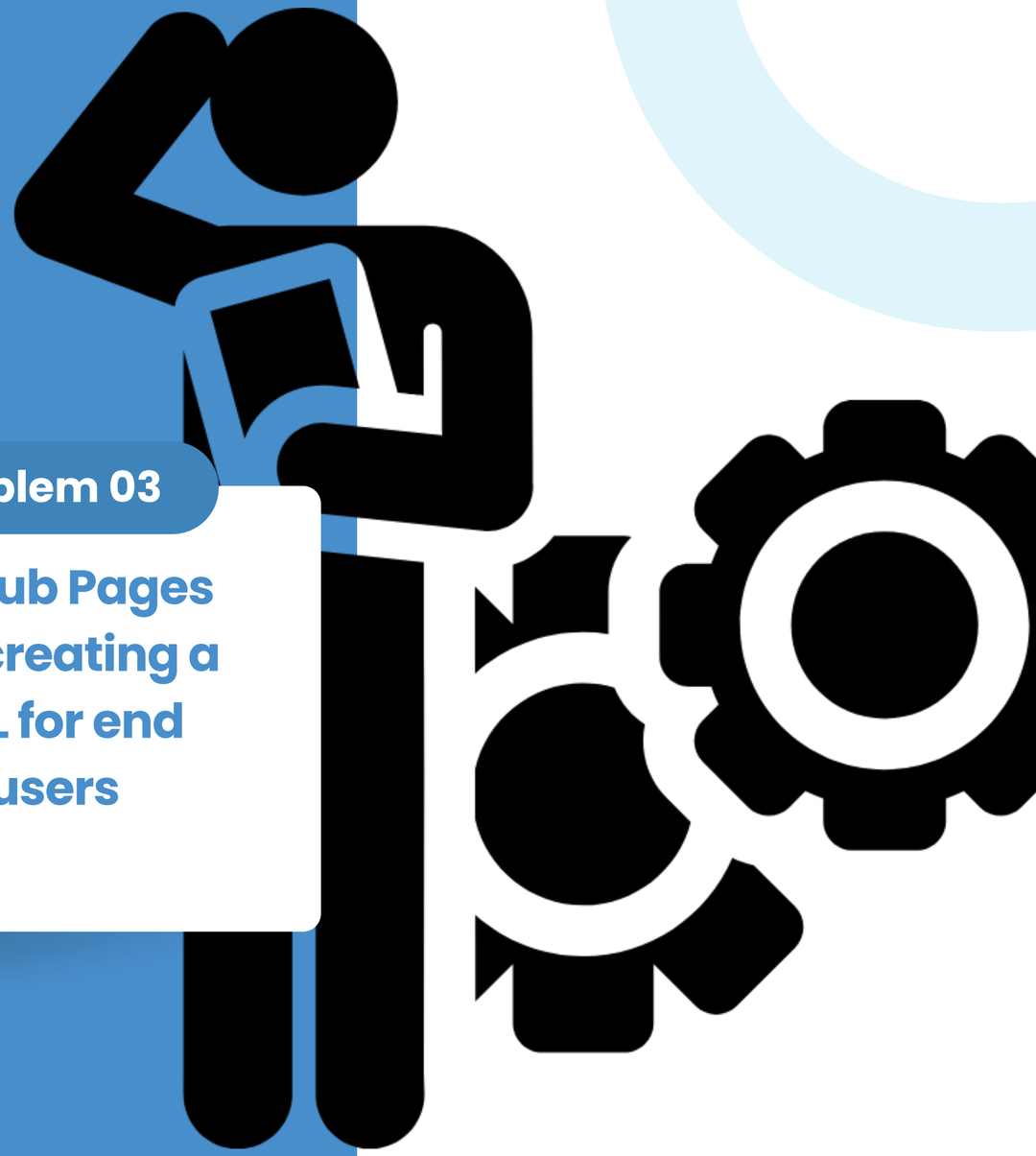
Allowed me to focus on other issues as the app already had working functionality



# Problems

*This was truly a challenging project for me. I ran into so many issues that I started reading official Flutter pull requests to fix my issues.*

| Problem 01 | Problem 02 | Problem 03 |
| --- | --- | --- |
| Setting up the Flutter Environment and learning Docker | Setting up the deployment phase in the pipeline | GitHub Pages not creating a URL for end users |



# What I gained from this project

### Software Development Skills

- Gained experience in CI/CD pipelines.
- Gained knowledge of the setup of devcontainers and Docker images for workflows.

### Creating a Flutter Web Application

- Understand the requirements for successful Flutter web applications.
- Expanded proficiency in Flutter, Dart, YAML, and Markdown languages.

### Debugging skills

- Leveraged documentation and known errors for issue resolution.
- Fostered collaboration through sharing repository for effective problem-solving.

# QUESTIONS?


**Bibliography**

Brand. (n.d.). Retrieved April 29, 2024, from //flutter.dev/brand/

Build software better, together. (2024). GitHub. https://github.com

CI/CD Pipeline. (n.d.). CyberArk. Retrieved April 29, 2024, from https://www.cyberark.com/what-is/ci-cd-pipeline/

Development containers. (n.d.). Retrieved April 29, 2024, from https://containers.dev/

Git—Logo Downloads. (n.d.). Retrieved April 29, 2024, from https://git-scm.com/downloads/logos