***DOCUMENTATION***:

This documentation is a source of resource that will help in understanding how I built my app and what inspired me to use the famous 10 figures that I used in my app, this document here just summarizes everything, or rather gives out the explanation of how this was done and why it was done in that way. Below my two pictures is a little summary based on documentation that includes designing of the app, purpose of the app, GitHub and GitHub Actions. I would like to think that my documentation here has everything that is needed under documentation and gives you a better understanding of my reasons behind my design but most importantly, educate you on GitHub and GitHub Actions, so that you will have a clue on what is what and know how to separate the two from each other with getting a confusion. My app here was created with passion and love, so that I could get what I want from my app and it for to be catchy and impressive to other people who are going to be looking at my app. I hope that however is going to read this documentation gets a clue of what’s following underneath these pictures, and find it easy and impressive, just because of how everything is explained, and how is looking like. This historical app was not something easy to build nut it was worth the try, because it came beyond the word perfect.

Here are some explanations and pictures, that are going to help in making everything clear, and provide a better understanding:

A person in a garment holding a spear

Description automatically generated

The purpose of this this history app is to show 10 famous figures and what they have achieved, roles they have played and marks they have left behind, that would be used to remember them. It’s here to give us a brief of how they became famous and indicate the reason behind me making them my famous figures. My app was designed like so that it can be catchy and look very much enthusiastic pleasing, designing this app is a tool to make it more attractive to the read, because it will make the want to know what is behind this picture. The consideration of designing is to make the app top notch.

The utilisation of GitHub and GitHub Action:

GitHub: It’s a platform that is primarily used for version control and software development projects. It allows developers to review code, build, host and manage projects. Some most common use of Git include:

Version Control: Developers use GitHub to track changes on their coding over a certain period, enabling them to revert to previous versions that are needed and collaborate with other developers that are using the exact same code.

Collaboration: Many developers can make together on the same project at once, just by forking repositories, making some changes and submitting pull requests on the purpose of those changes that need to merge into the original codebase.

Continuous Integration/Continuous Deployment (CI/CD): GitHub integrates with various CI/CD tools, allowing developers to automate testing and deployment processes, thereby ensuring the quality and reliability of their software’s.

Open-Source Contributions: GitHub hosts millions of open- source projects, making it a hub for developers to contribute to projects they are interested in and to learn from code written by others.

Issue Tracking and Documentation: This app provides tools for tracking bugs, feature requests, and other tasks through its issue tracking system. It allows the team to work effectively.

Overall, GitHub serves as a central platform for developers to collaborate, share knowledge, and build software more efficiently.

GITHUB ACTIONS:

It’s a powerful tool provided by GitHub for automating workflows directly within your GitHub repository. Some common utilization of GitHub Actions include:

Continuous Integration (CI): Automatically build, test, and validate the code and changes whenever a pull requested to open, pushed to a branch, or scheduled basis. This ensures that your code remains stable and functional.

Continuous Deployment (CD): Automatically deploy your application to staging or production environments after passing all necessary validations. This streamlines the deployment process and reduce the risks of human error.

Custom Workflows: Create custom workflows tailored to your project’s specific needs, incorporating multiple actions and triggers to automate complex tasks.

Release Management: Automatically create release notes, tag releases, and publish artifacts whenever a new version of your software is ready. This helps streamline the release process and ensures that all necessary steps are completed consistently.

Issue and Pull Request Management: Automatically assign, label, or close issues and pull requests based on predefined criteria or actions. The streamline issue is triage process and helps keep your project organized.

Scheduled Tasks: Run periodic tasks such as database backups, code analysis, or maintenance scripts on a predefined schedule. This helps automate tasks to allow.

Documentation: GitHub repositories often include documentation for projects, which can be collaboratively edited and maintained. This helps developers understand how to use the software and contribute to its improvement.

Overall, GitHub Actions enables developers to automate various aspects of their development workflow, increasing productivity, improving code quality, and reducing manual overhead.

Here is the difference between GitHub and GitHub Action:

GitHub is a platform for version control and collaboration, primarily used for hosting and managing code repositories. GitHub Actions, on the other hand, is a feature of GitHub that allows you to automate workflows directly within your GitHub repository. With GitHub Actions, you can define custom workflows to build, test, and deploy your code, among other tasks, all within the GitHub the GitHub ecosystem. So, GitHub is the platform itself, while GitHub Actions is a specific feature within that platform for automating workflows.

Advantages and Disadvantages of GitHub:

Advantages:

Integration: It integrates seamlessly with various development tools and services, such as continuous integration (CI) systems, code editors project management tools, and deployment platforms, enhancing the development workflow.

Disadvantages:

Dependency on internet Connection: GitHub is a cloud-based platform, so developers need a stable internet connection to access and work on their projects, which van be a limitation in areas with poor connectivity.

Advantages and Disadvantages of GitHub Action:

Advantages:

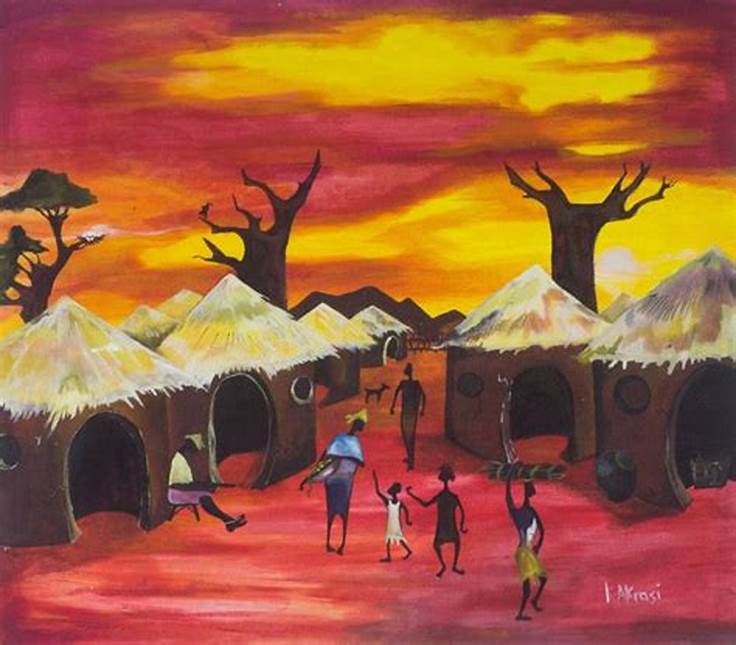
Customization: With GitHub Actions, you can define custom workflows using YAML syntax, allowing you to tailor the automatic process to suit your project’s specific requirements, This Flexibility enables you to create complex workflows with multiple steps and conditions.

Disadvantages:

Complexity: As workflows become more complex with multiple steps and conditions, managing and debugging them can become challenging. Ensuring consistency and reliability across different workflows and environments requires careful planning and testing.

Cost: While GitHub Actions offer free usage for public repositories and a certain amount of free usage for private repositories, exceeding the free usage limits or requiring additional features may incur costs, especially for larger projects or organisation.

Overall, despite these disadvantages, GitHub Actions provides significant benefits in terms of automation, integration, and customization, making it a valuable tool for streamlining and improving your development workflow

  THIS PICTURE HERE CLOSES MY EXPLANATION BASED ON THE QUESTIONS UNDER DOCUMENTATION, I HOPE THAT VERYTHING THAT WAS NEEDED, WAS PROVIDED IN THIS DOCUMENTATION. I WOULD LIKE TO THANK YOU FOR TAKING YOUR PRECIOUS TIME AND READING THIS DOCUMENTATION. HAVE A PHENOMNAL JUBLIANT NIGHT/ DAY.

THANK YOU & THE END OF THIS PROJECT!..

Reference:

OpenAI 2023

<http://sites.google.com>

my knowledge