**MLOPS(S1-25\_AIMLCZG523) Assignment-1**

**Group 92**

|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **Contribution** |
| **ADITYA BHUVANGIRI** | **2024aa05257@wilp.bits-pilani.ac.in** | **100%** |
| **BODDUPALLI VENKATA NAGA LOKESH** | **2024aa05360@wilp.bits-pilani.ac.in** | **100%** |
| **PUTHINEEDI VENKATA SAI CHARAN** | **2024aa05606@wilp.bits-pilani.ac.in** | **100%** |
| **ABHISHEK SRIVASTAVA** | **2024aa05125@wilp.bits-pilani.ac.in** | **100%** |
| **SHUBH SAXENA** | **2024aa05250@wilp.bits-pilani.ac.in** | **100%** |

|  |  |
| --- | --- |
| **Resource** | **Link / Details** |
| **GitHub Repository** | [**https://github.com/Lokesh-100/mlopsAssignment1**](https://github.com/Lokesh-100/mlopsAssignment1) |
| **Live Application** | [**https://heart-service-prediction.c-321a6c0.stage.kyma.ondemand.com/docs**](https://heart-service-prediction.c-321a6c0.stage.kyma.ondemand.com/docs) |
| **Explanation Video Link** | [**Click Here for Video**](https://wilpbitspilaniacin0-my.sharepoint.com/personal/2024aa05257_wilp_bits-pilani_ac_in/_layouts/15/stream.aspx?id=%2Fpersonal%2F2024aa05257%5Fwilp%5Fbits%2Dpilani%5Fac%5Fin%2FDocuments%2FMLOps%20assignment%20recording%2FHeart%20Disease%20Prediction%20Model%20%2D%20Slow%2Emp4&nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJTdHJlYW1XZWJBcHAiLCJyZWZlcnJhbFZpZXciOiJTaGFyZURpYWxvZy1MaW5rIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXcifX0&referrer=StreamWebApp%2EWeb&referrerScenario=AddressBarCopied%2Eview%2E0ccf02cc%2Df19e%2D4585%2D9404%2D80ac14b1041d&ga=1) |
| **Docker Hub Image** | [**https://hub.docker.com/r/aditya3298/mlops/tags**](https://hub.docker.com/r/aditya3298/mlops/tags) |
| **Model Artifacts (Hugging Face)** | [**https://huggingface.co/adityabhuvangiri/heart\_models/tree/main**](https://huggingface.co/adityabhuvangiri/heart_models/tree/main) |
| **Grafana Dashboard** | [**https://grafana-mlops.c-158d220.kyma.ondemand.com/dashboards**](https://grafana-mlops.c-158d220.kyma.ondemand.com/dashboards) |
| **Grafana Credentials** | **Username: admin Password: admin** |
| **CI Pipeline (GitHub Actions)** | [**https://github.com/Lokesh-100/mlopsAssignment1/actions/workflows/ci.yml**](https://github.com/Lokesh-100/mlopsAssignment1/actions/workflows/ci.yml) |
| **CD Pipeline (GitHub Actions)** | [**https://github.com/Lokesh-100/mlopsAssignment1/actions/workflows/cd.yaml**](https://github.com/Lokesh-100/mlopsAssignment1/actions/workflows/cd.yaml) |

**Architecture Summary Content**

**Project: End-to-End MLOps Pipeline for Heart Disease Prediction**

**Overview**

This document outlines the architecture of a complete MLOps pipeline designed to train, version, test, package, deploy, and monitor a machine learning model. Using the Heart Disease UCI Dataset as a foundation, this project implements a full lifecycle workflow with an emphasis on automation, reproducibility, and observability through modern open-source tools.

The objective is to design a **reproducible, automated, containerized, and deployable ML system** following modern MLOps best practices.

## **Architecture Components**

### **Source Code & Version Control (Git & GitHub):**

* All project assets are version-controlled using Git and hosted on GitHub.
* This includes:
  + Application source code
  + Configuration files (Dockerfile, Kubernetes deployment manifests, Grafana setup)
  + CI and CD workflow definitions
  + Unit and integration test scripts
* Version control ensures traceability, collaboration, rollback capability, and controlled evolution of the system.

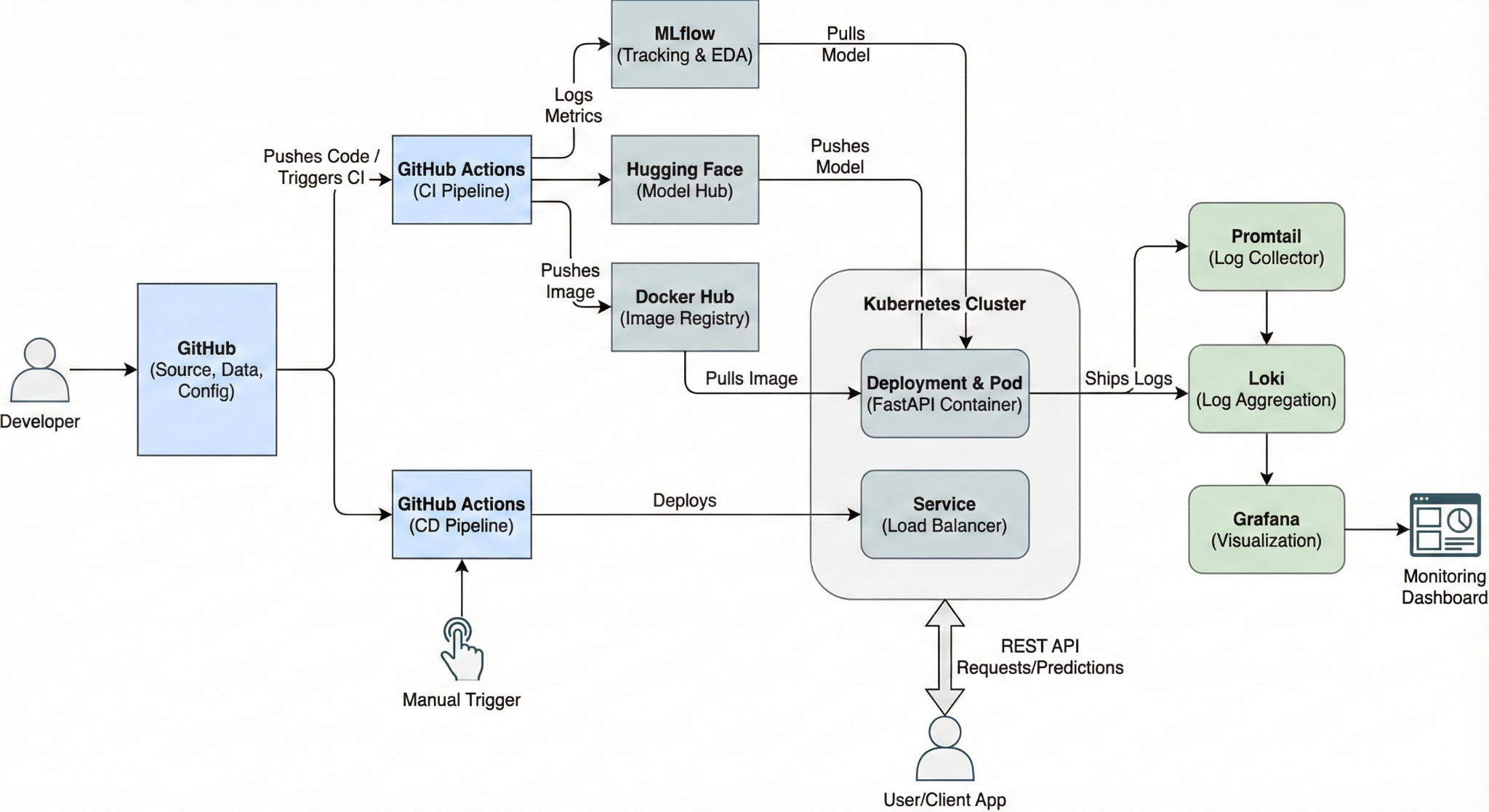
### **Data & Model Versioning (Git, MLflow, Hugging Face):**

* **Data Versioning:**
  + The UCI Heart Disease dataset is versioned directly within the Git repository.
  + This guarantees reproducibility of experiments across different runs.
* **EDA Artifacts:**
  + Exploratory Data Analysis outputs such as plots and statistical summaries are stored as MLflow artifacts.
* **Model Versioning:**
  + Trained model artifacts are uploaded and versioned in the Hugging Face Model Hub under the heart\_models repository.
  + Each model version is tagged and traceable to a specific CI run.

### **Experiment Tracking (MLflow) Summary:**

* MLflow is used to track all model training experiments.
* Logged information includes:
  + Hyperparameters
  + Evaluation metrics
  + Training artifacts
* This enables experiment comparison, auditing, and complete reproducibility of results.

**Architecture diagram:**



### 

### **Model Serving & Containerization (FastAPI & Docker):**

* Multiple baseline models were evaluated using a consistent preprocessing pipeline:
  + Logistic Regression
  + Random Forest Classifier
* The Random Forest model was selected as the final model due to:
  + Higher ROC-AUC score
  + Better recall for the positive (heart disease) class, which is critical in healthcare use cases
* The trained model is exposed via REST APIs using FastAPI.
* The application is containerized using Docker and pushed to Docker Hub for deployment.

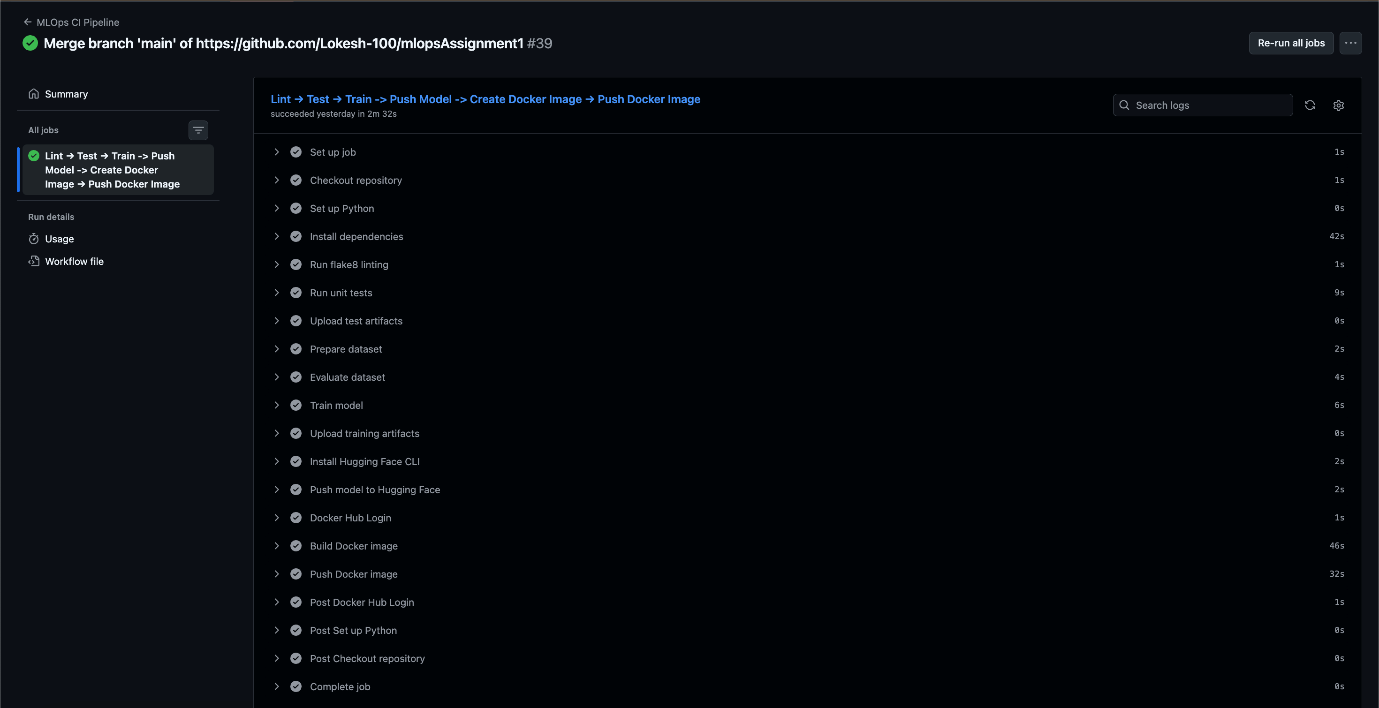
### **Continuous Integration & Testing (CI with GitHub Actions):**

* A CI pipeline is triggered automatically on every push to the main branch.
* The pipeline performs the following automated steps:
  + Code linting and quality checks
  + Unit testing
  + Dataset preparation and evaluation
  + Model training and evaluation
  + Docker image creation
  + Uploading model and training artifacts
* Test reports and artifacts are published as part of the pipeline run for transparency.

**Screenshots of CI and CD Pipeline and Steps:**

**CI Pipeline:**

**CI Pipeline Steps:**



### **Continuous Deployment (CD with GitHub Actions):**

* A separate CD pipeline is configured and triggered manually.
* The pipeline:
  + Deploys the containerized application to the Kubernetes environment
  + Uses predefined deployment manifests to ensure consistency and repeatability

**CD Pipeline:**



### **CD Pipeline Steps:**

### 

### **Live Deployment & Live Monitoring Stack (Kubernetes, Grafana):**

**Deployment:**

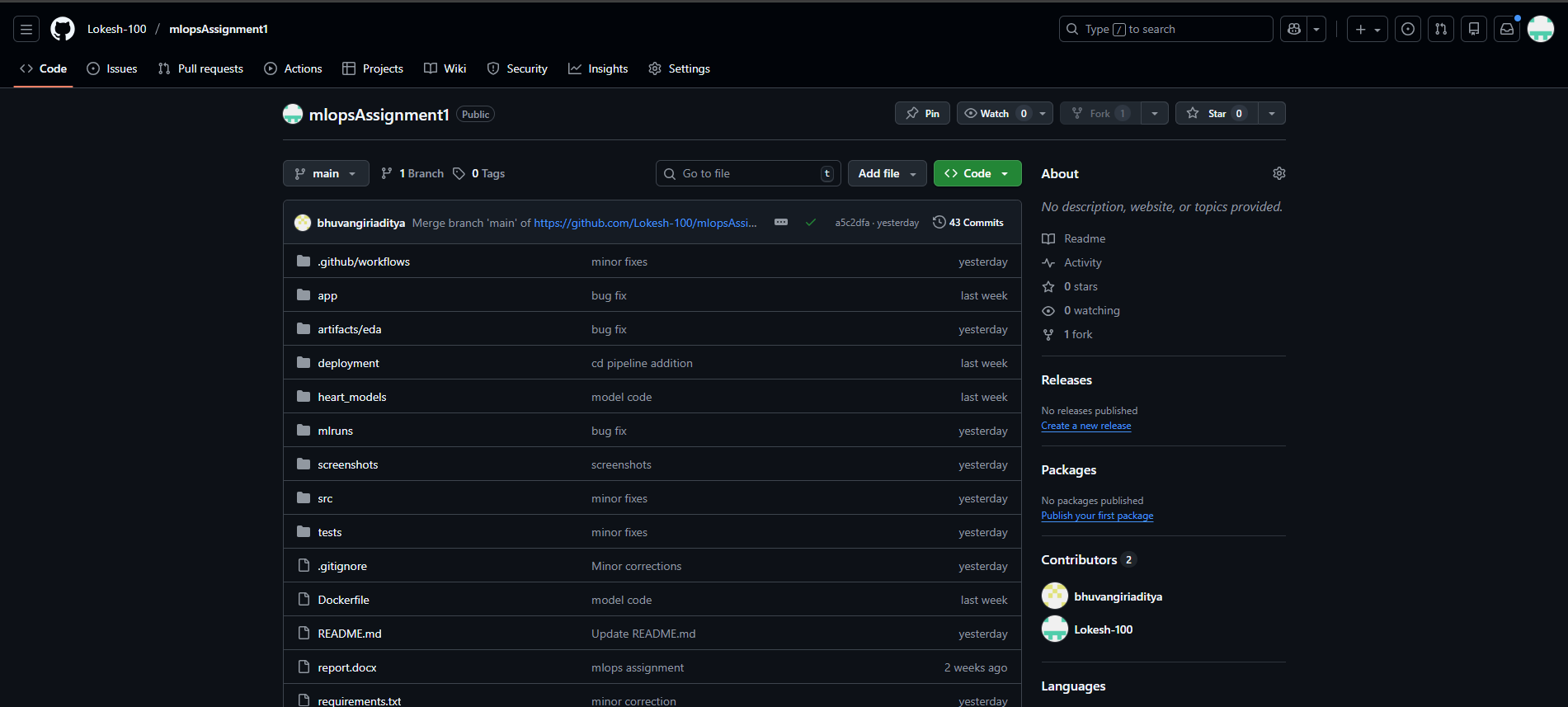
* The complete application stack is deployed on a Kubernetes cluster.
* Deployment is handled using a deployment.yml manifest as part of the CD pipeline.

**Monitoring:**

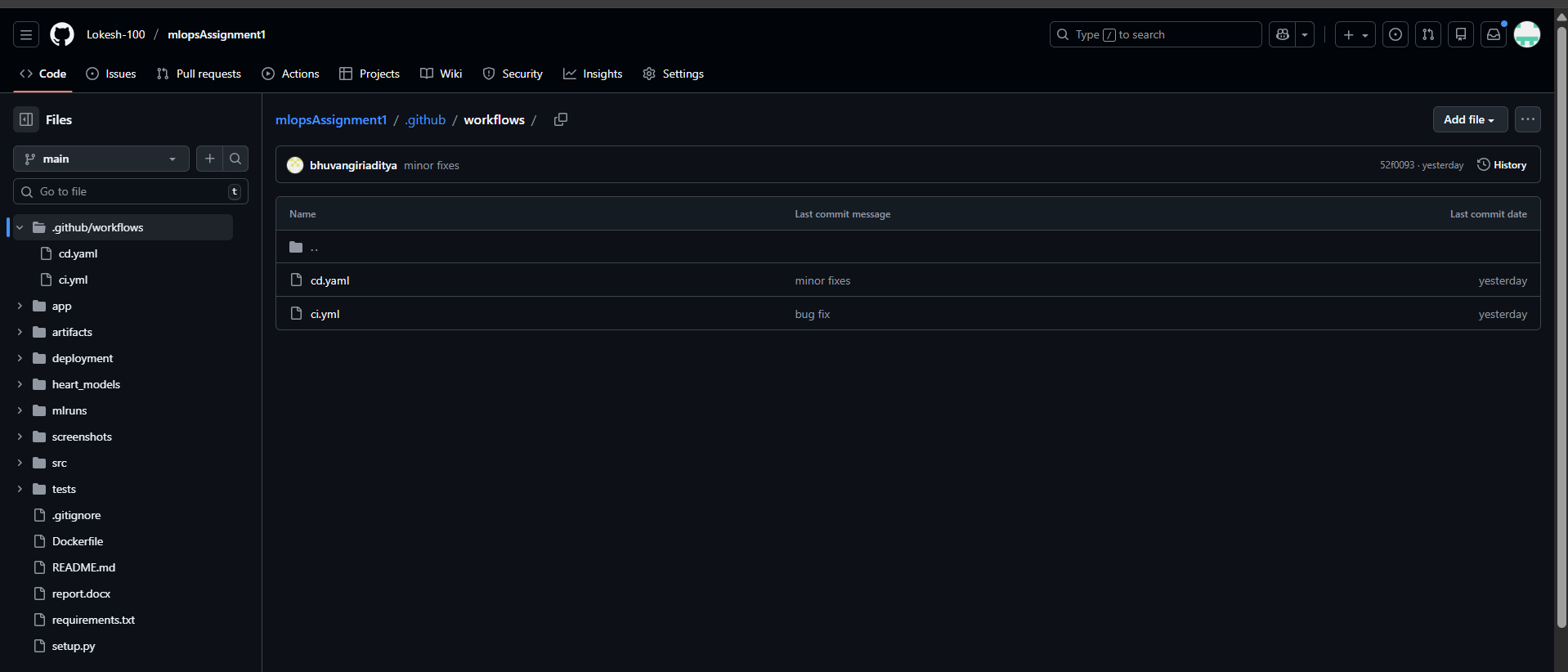
* The monitoring stack is set up using Helm charts for:
  + Promtail (log collection)
  + Loki (log aggregation)
  + Grafana (visualization)
* Application logs are streamed and visualized in real time using Grafana dashboards.
* A live monitoring link is provided to observe system behavior and logs.

**Screenshots**

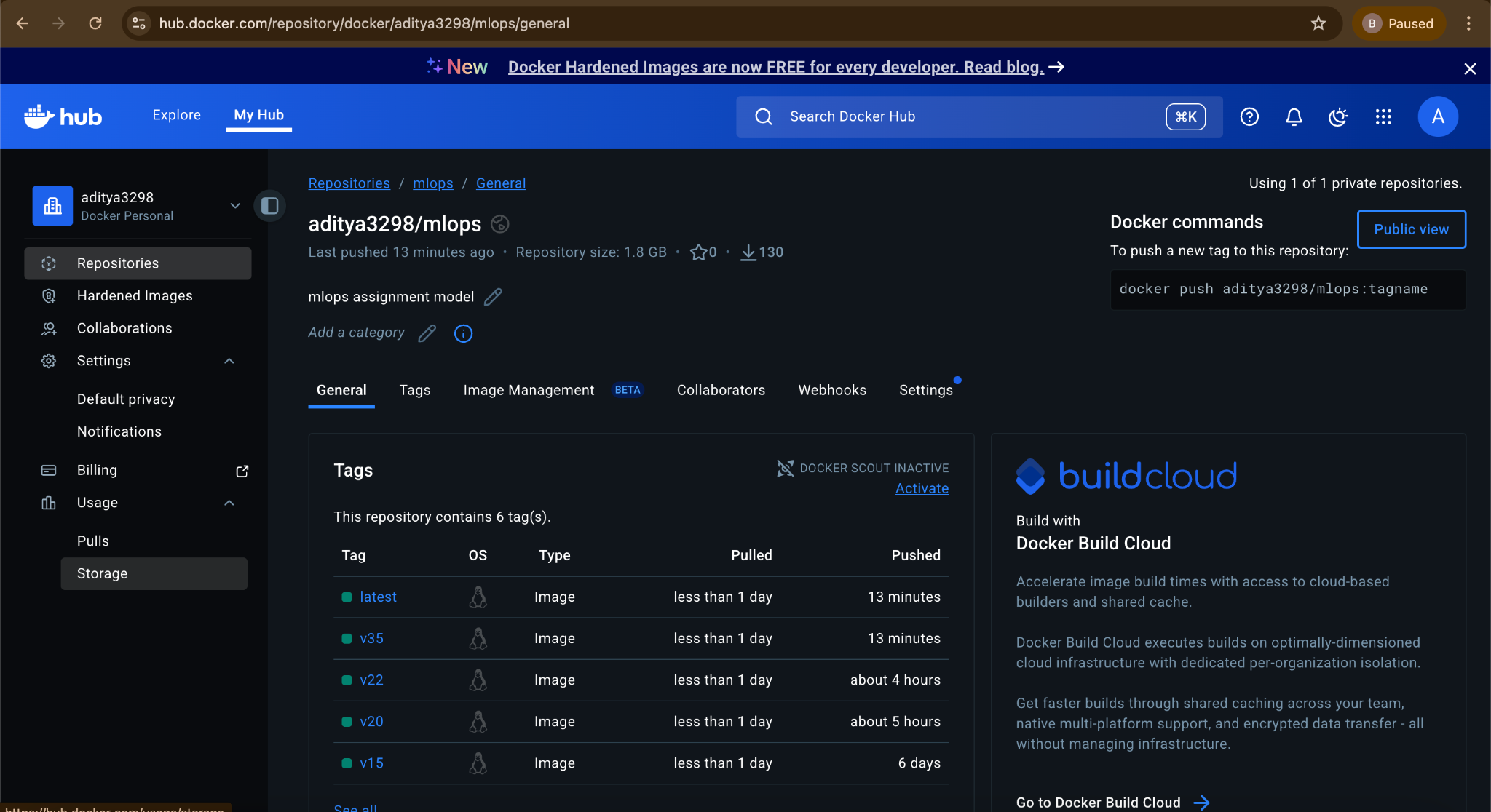
**GitHub Page:**

****

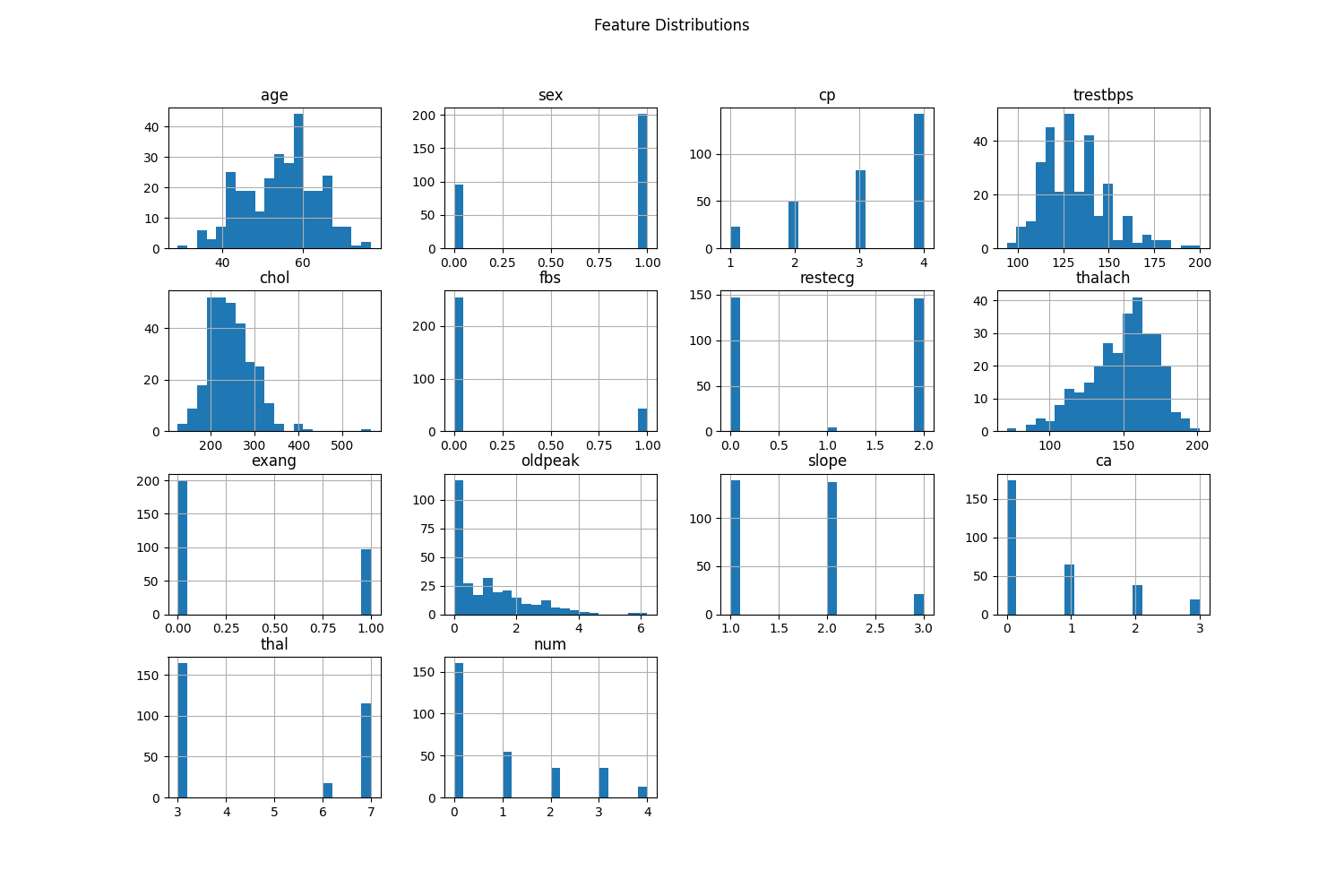
**GitHub Workflows yml files:**



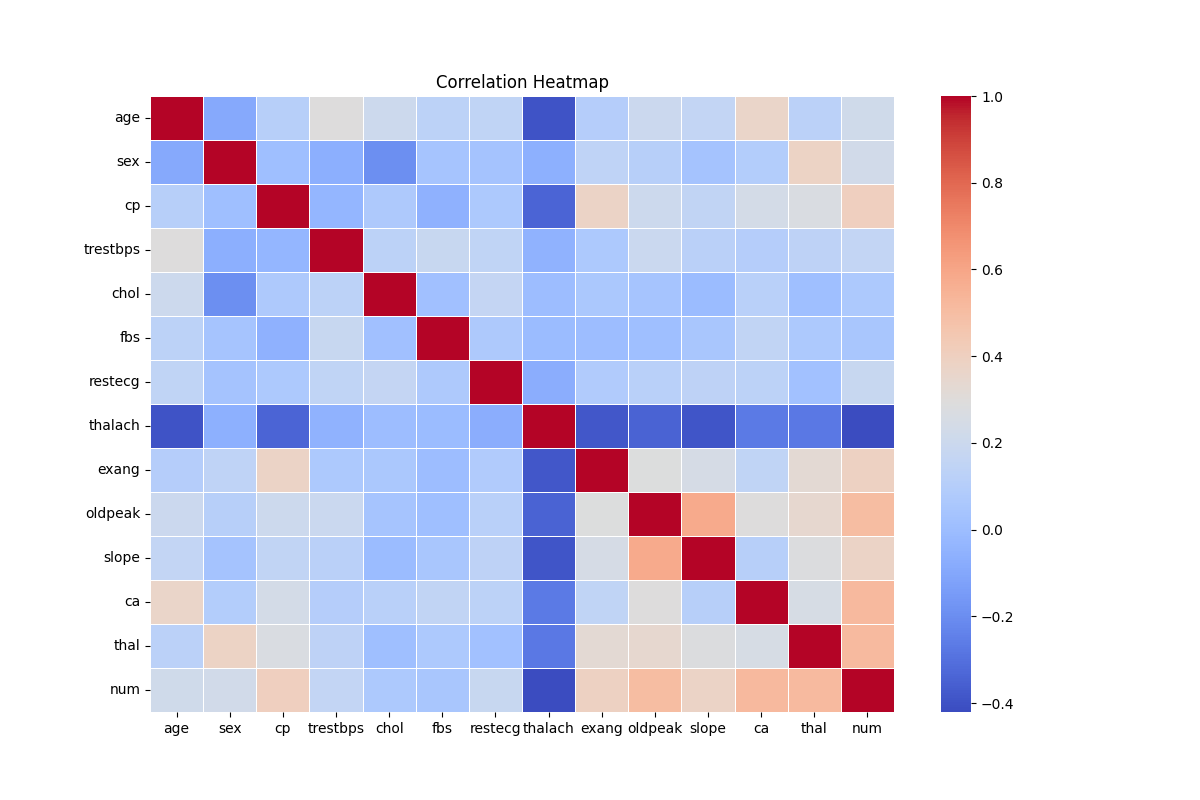
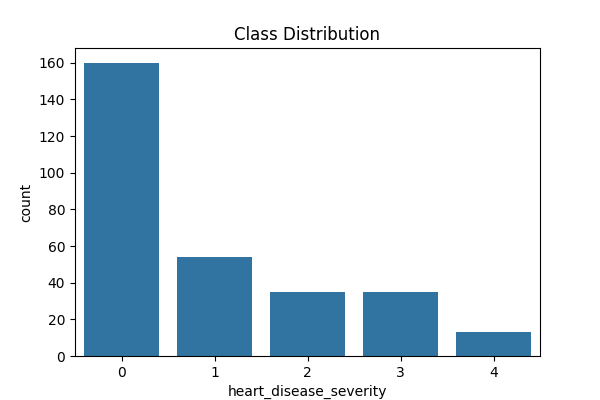
**Docker Registry:**

****

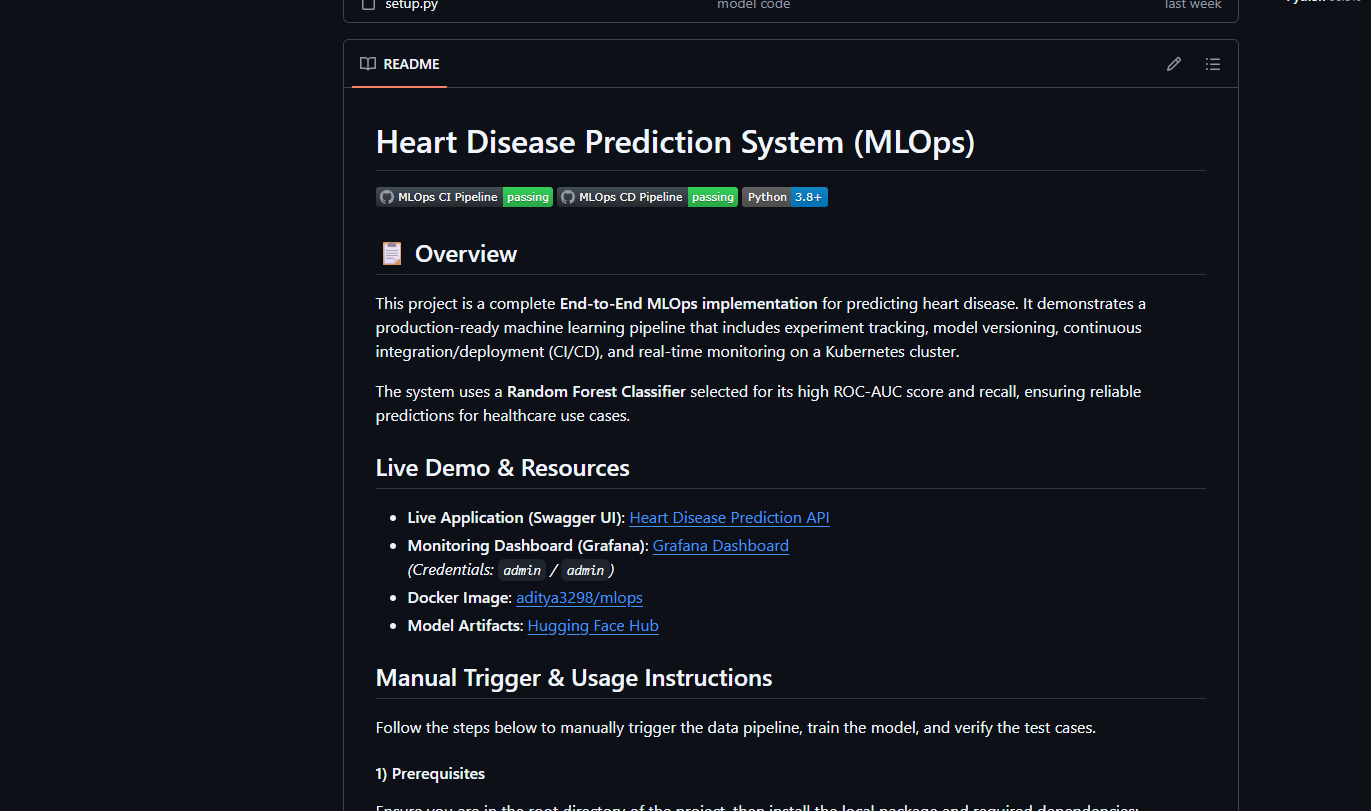
**Exploratory Data Analysis:**

**Feature Distribution Histogram:**

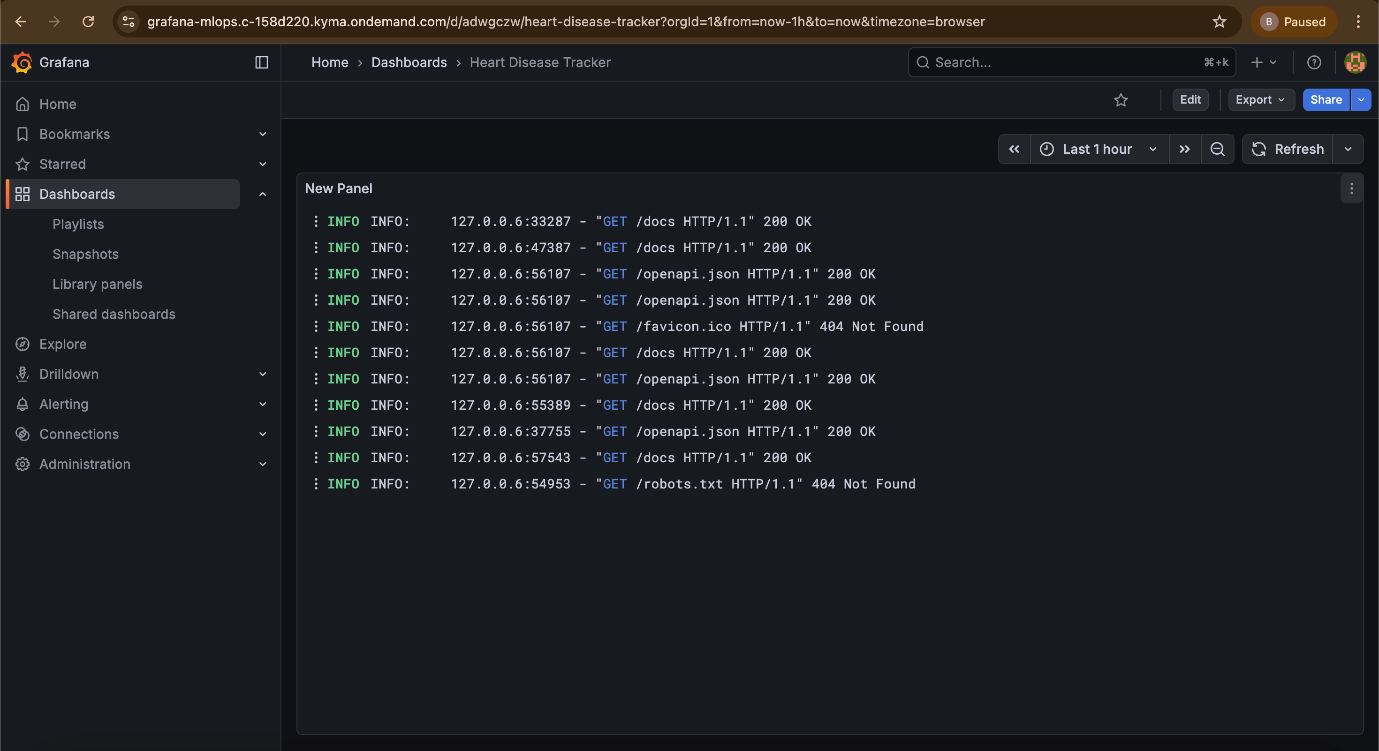
**Correlation Heatmap:**

**Class Distribution:**

**Readme in GitHub:**

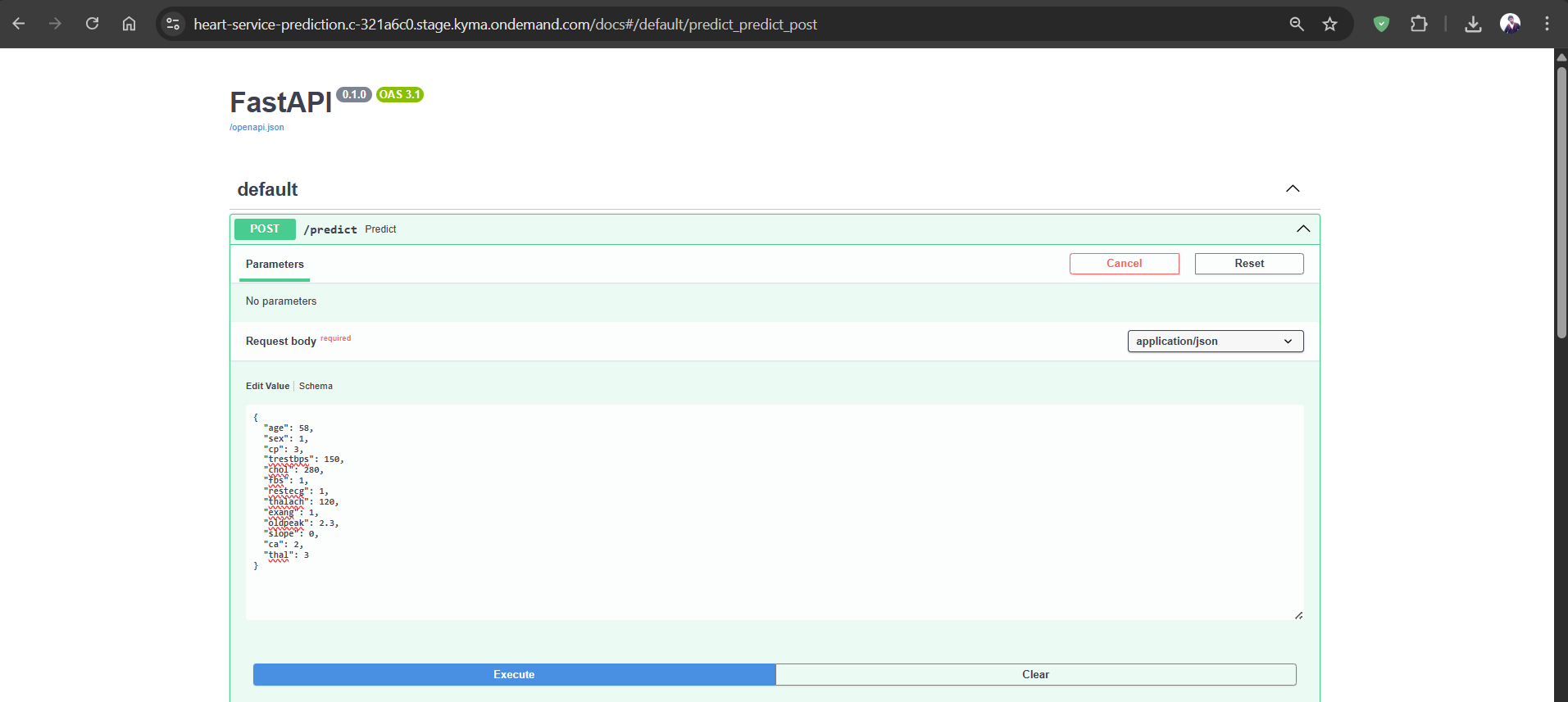
****

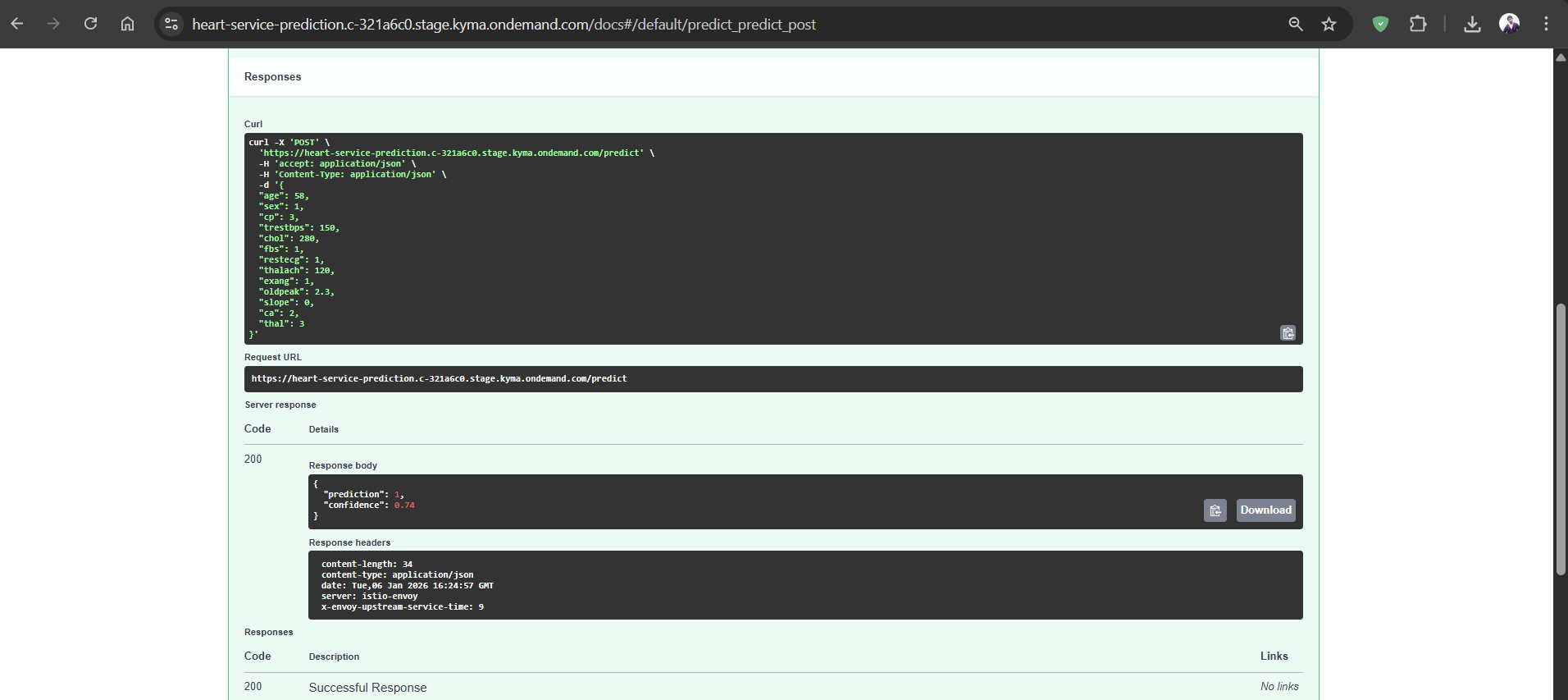
**Graphana:**

****

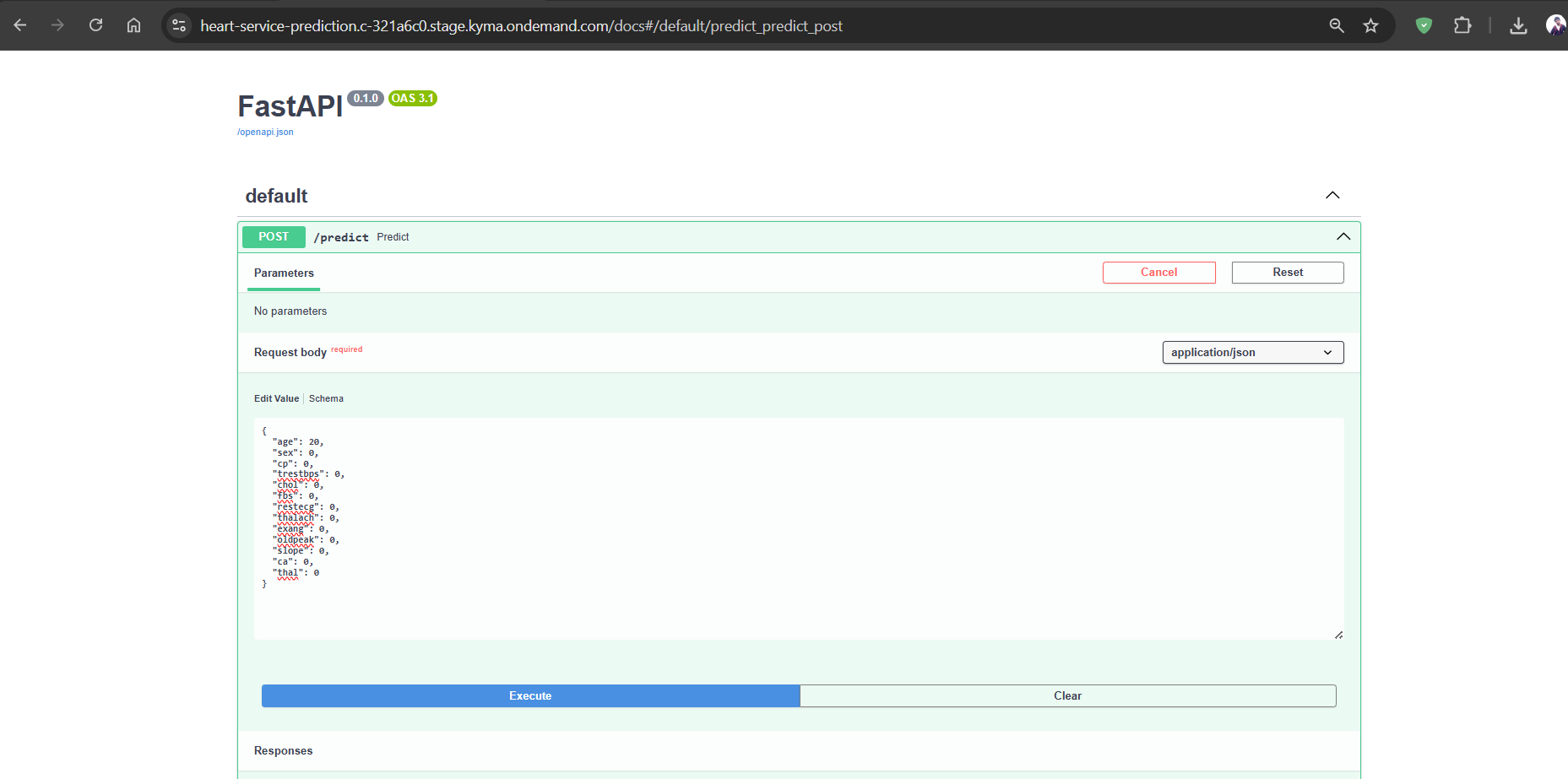
**FastAPI:**

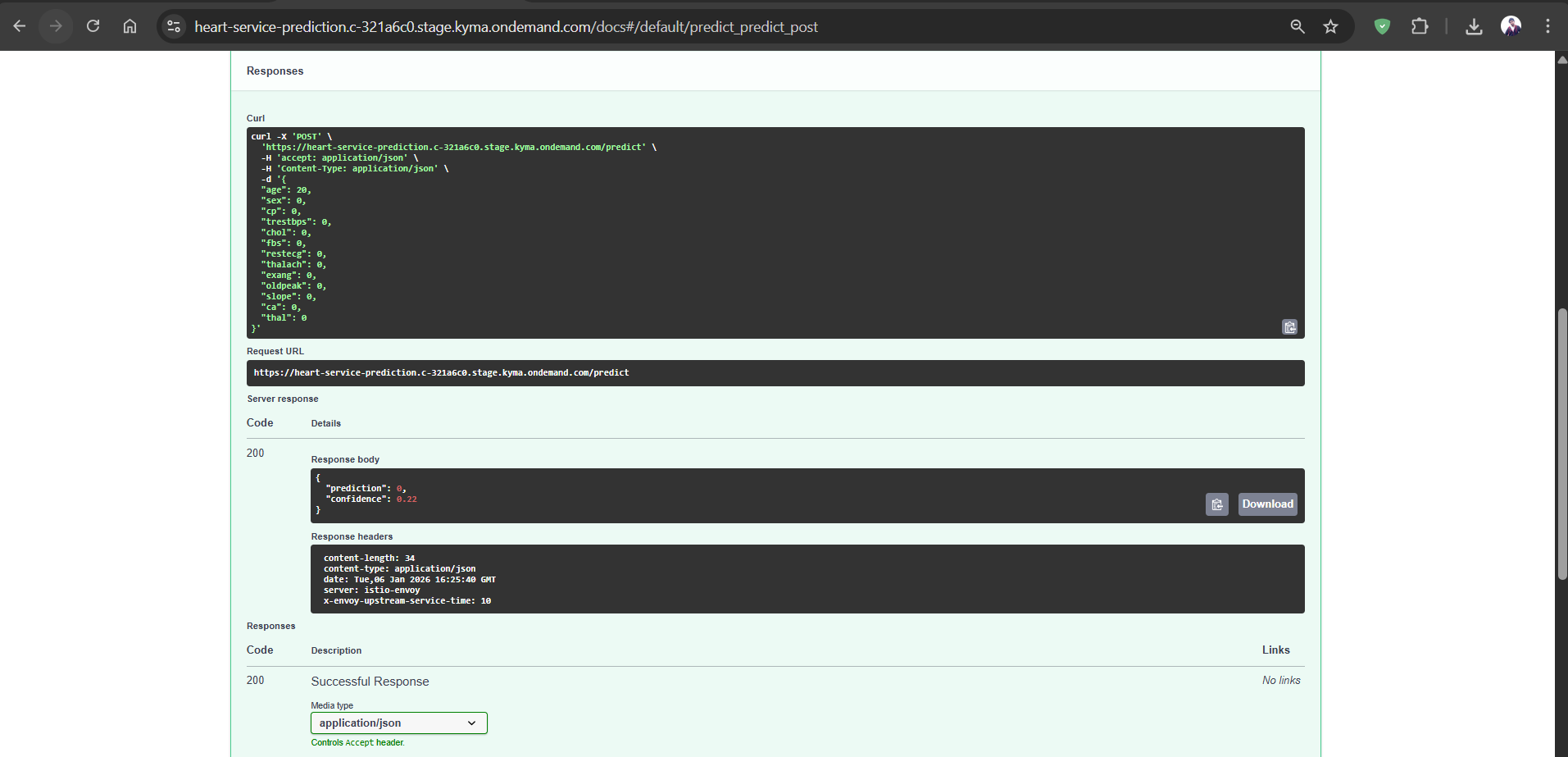
**Scenario-1: Model predicting Heart Disease will occur:**

****

****

**Scenario-2: Model predicting of No Heart disease:**

****

****