Assignment Documentation: Deploying a Python Web Application on Google Cloud Run

Introduction:-

This document outlines the process of deploying a Python web application that performs face registration and recognition using the DeepFace library on Google Cloud Run. The assignment aimed to evaluate the deployment skills, coding practices, and understanding of containerization using Docker and deployment on Google Cloud Platform (GCP).

Project Overview:-

The assignment involved creating a Python web application using Flask, implementing face registration and recognition using the DeepFace library, containerizing the application with Docker, and deploying it on Google Cloud Run. The application allows users to register faces and recognize registered faces from uploaded images.

Prerequisites:

Before starting the assignment, make sure you have the following:

* Python installed on your local machine.
* Google Cloud Platform account with a project created.
* Docker installed for containerization.

Steps to Complete the Assignment:

Step 1: Creating a Python Web Application

1. Clone the assignment repository to your local machine.
2. Navigate to the project directory: cd Assignment-Deploy-a-Python-Web-Application-on-Google-Cloud-Run.
3. Create a virtual environment (recommended): python3 -m venv venv and activate it.
4. Install required packages: pip install -r requirements.txt.
5. Implement the Flask application according to the provided code in app.py.
6. Create an upload folder within the project directory to store registered images.
7. Design the front-end HTML template in the templates folder.

Step 2: Containerizing the Application with Docker

1. Create a Dockerfile in the project directory.
2. Define the base image, working directory, and required dependencies.
3. Copy all project files into the container.
4. Specify the port to expose and the command to run the Flask app.

Step 3: Deploying on Google Cloud Run

1. Create a project on Google Cloud Platform (GCP) if not already done.
2. Enable the Cloud Run API for your project.
3. Build the Docker image: docker build -t your-image-name ..
4. Tag the image for the Google Container Registry: docker tag your-image-name gcr.io/your-project-id/your-image-name.
5. Push the image to the Google Container Registry: docker push gcr.io/your-project-id/your-image-name.
6. Go to Google Cloud Console, select Cloud Run, and click "Create Service".
7. Configure the service with your desired settings, including the image from the Container Registry.
8. Deploy the service and wait for it to be provisioned.
9. Access the deployed app via the provided URL.

Step 4: Testing and Verification

1. Access the deployed web application URL to test the registration and recognition functionalities.
2. Register a face with a name and an image using the provided form.
3. Try recognizing the registered face using the recognition form.
4. Verify that the application behaves as expected and displays the recognized name or a suitable message.

Challenges Faced:

Throughout the assignment, several challenges were encountered and addressed:

1. Code Debugging: Debugging the Flask application code to ensure proper integration with the DeepFace library for face recognition.
2. File Paths: Managing file paths correctly to ensure that registered images were saved and loaded properly.
3. Docker Configuration: Properly configuring the Dockerfile with the right dependencies and commands to run the application.
4. Front-End Interactivity: Ensuring that the front-end interface (HTML and JavaScript) interacted correctly with the Flask backend.
5. Environment Variables: Configuring environment variables for the deployed application on Google Cloud Run.
6. Google Cloud Run Setup: Configuring Google Cloud Run settings, networking, and authentication.

Conclusion:

This assignment provided hands-on experience in creating a Python web application, integrating external libraries for specialized tasks, containerizing the application with Docker, and deploying it on Google Cloud Run. The challenges faced throughout the process contributed to a deeper understanding of web development, containerization, and cloud deployment, as well as the problem-solving skills required to overcome obstacles in the development and deployment process.