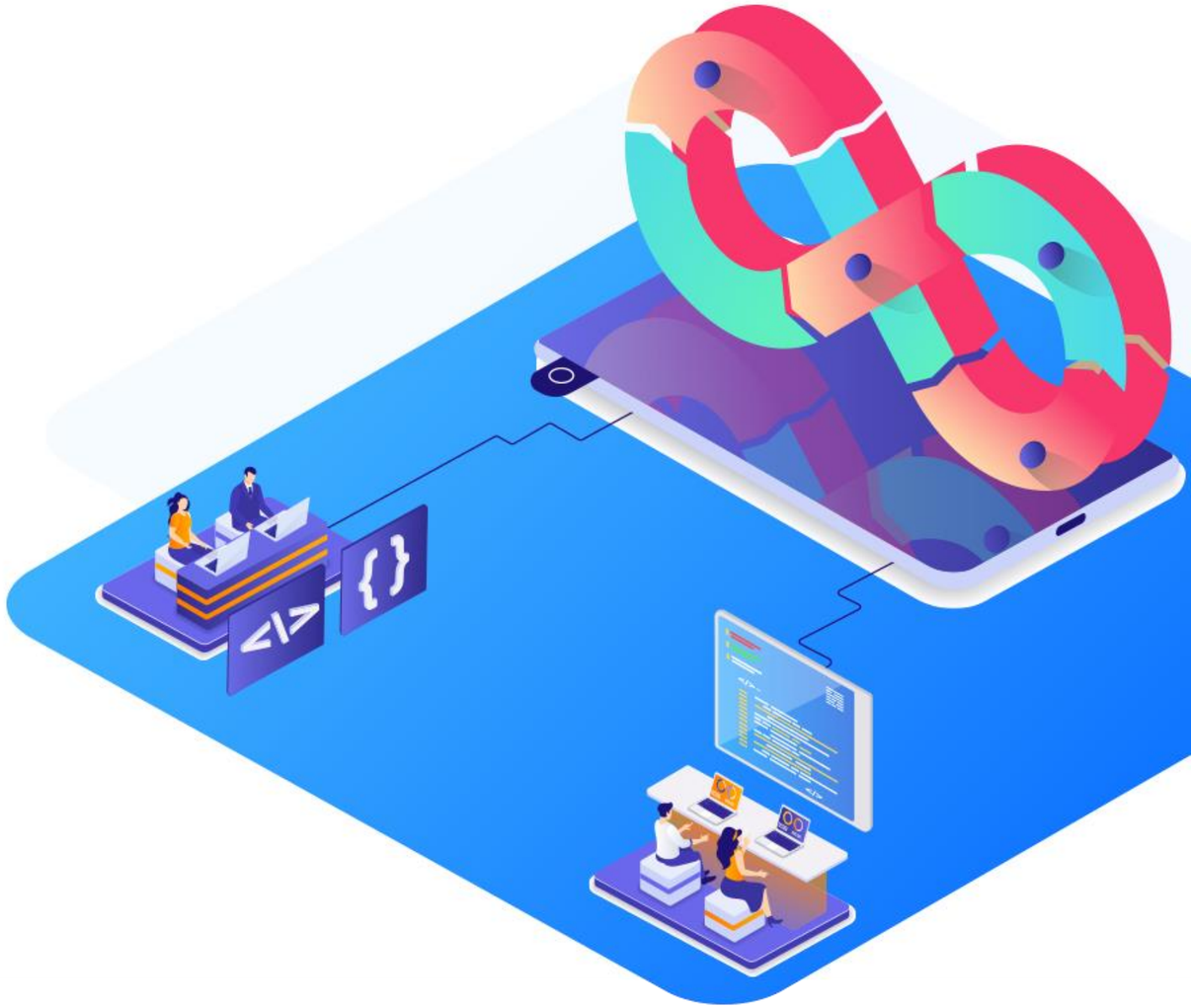
Containerization with Docker



Course-End Project

Deploying Multi-Tier Application

Objective

To demonstrate the deployment of a multi-tier application using Docker compose, including configuring public IP addresses and setting up necessary security group rules for accessing the frontend, API, and database services.

Problem Statement and Motivation

Real-time scenario:

A user is tasked to deploy a multi-tier application on a cloud-based virtual machine using Docker Compose. After installing Docker, Docker Compose, and Git, they clone the application repository and update configuration files with the machine's public IP to ensure proper communication between the frontend, API, and backend components.

Post these updates, the user runs docker compose to bring up the application, adjust the security group settings to allow traffic on specific ports (8080, 5000), and verify the deployment by accessing the application via the public IP. This confirms that all components are accessible and functional.

Industry Relevance

The following tools used in this project serve specific purposes within the industry:

1. **Docker compose**: A tool for defining and managing multi-container Docker applications using a YAML file, simplifying the orchestration of services

2. **Git**: A version control system used to clone the application repository from GitHub, enabling easy access to and management of code

3. **Security groups (Cloud provider tool)**: A cloud provider feature that controls inbound and outbound traffic to and from your virtual machine by defining specific rules for various ports

Tasks

The following tasks outline the process of:

1. Create an EC2 instance
2. Set up the environment by installing the required software
3. Deploy the multi-tier application
4. Access and verify the application

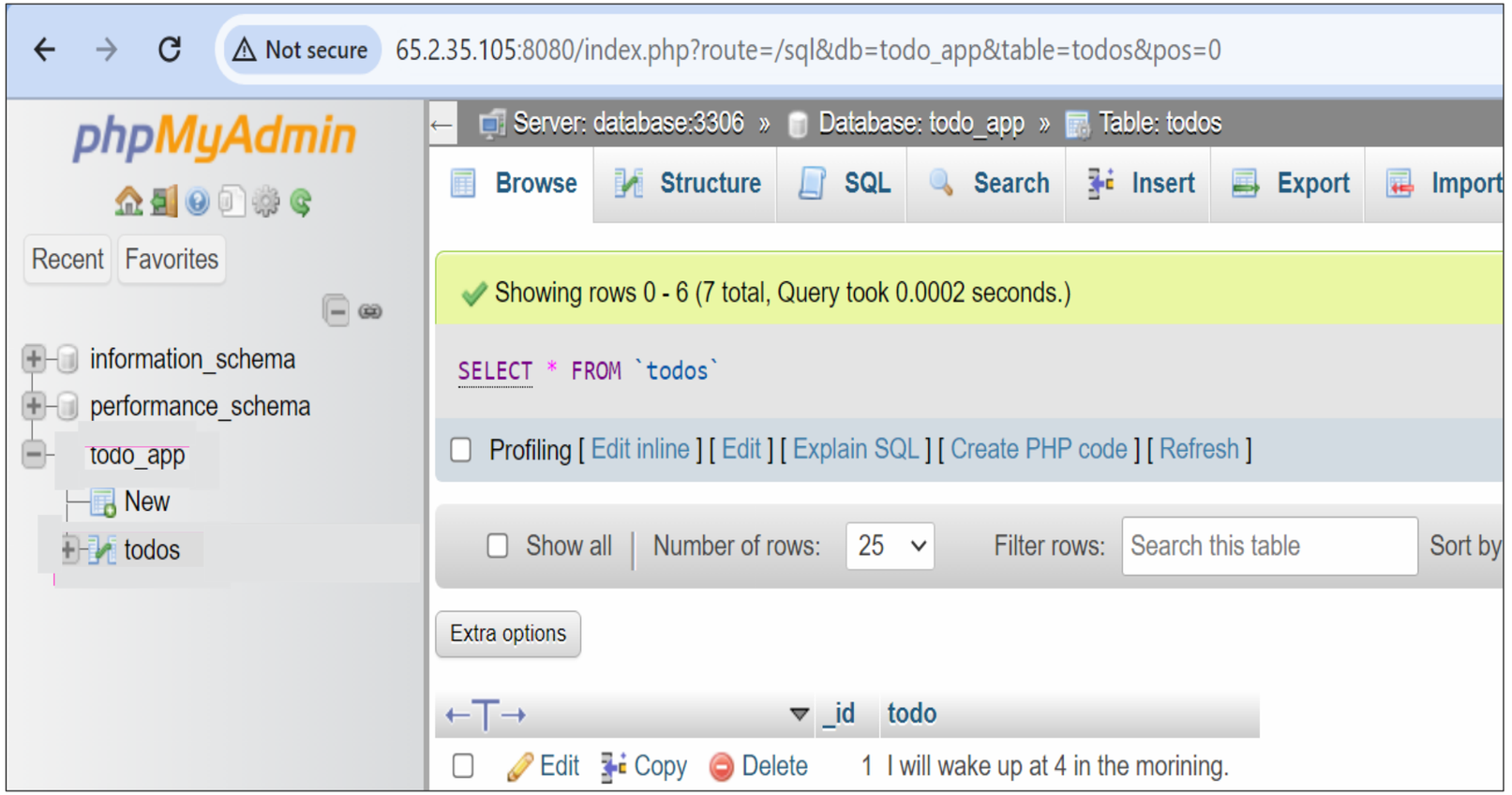
Project References

- **Task 1:**
- **Task 2:**
- **Task 3:**
- **Task 4:**

COURSE-END PROJECT

Output Screenshots

phpMyAdmin interface page



Thank you