Lab 3: UberEats (GraphQL)

# Purpose

The purpose of this lab is to develop an application (UberEats) to demonstrate skills in full stack development of distributed systems using MySQL, React, Node and GraphQL. The application is designed to display restaurants associated so that the customers can order food from these restaurants.

# System Design

The current system consists of the **Database (MySQL)** that stores the User and Restaurant data including order, menu, favorites etc. The data is managed by the Backend server, designed in NodeJs and Express with **GraphQL** which interacts with the **Client (React JS)** to complete the actions performed by the user and display appropriate data. The different states in the client are maintained using **Redux.**

# Git Commit History

A screenshot of a computer screen

Description automatically generated with medium confidence

A screenshot of a computer screen

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

# Steps Screenshots

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

# Answers

1. **How will you enable multi-part data in GraphQL?**

There is no way to enable multi part data using native graphql. This can be achieved using 3rd party libraries.

1. **Discuss the architecture for using multi-part data in GraphQL without using any open source library from Git.**

As GraphQL does not allow multi part data transfer using mutations, REST calls can be made in the mutations and the result of the upload, the URL, is added using GraphQL. Base64 encoded strings can also be passed in with GraphQL mutations. The encoded strings are usually large than their binary counterparts but a third. The encoding of strings can also become resource intensive pretty quickly and it is sometimes fraught with errors.

1. **State any open source library for enabling multi-part data transfer using GraphQL with sample code. Argue why do you think that this particular library is a good fit?**

apollo-upload-server library can be used to enable transferring of multi-part data in GraphQL.