

## CMPE 273 – Enterprise Distributed Systems

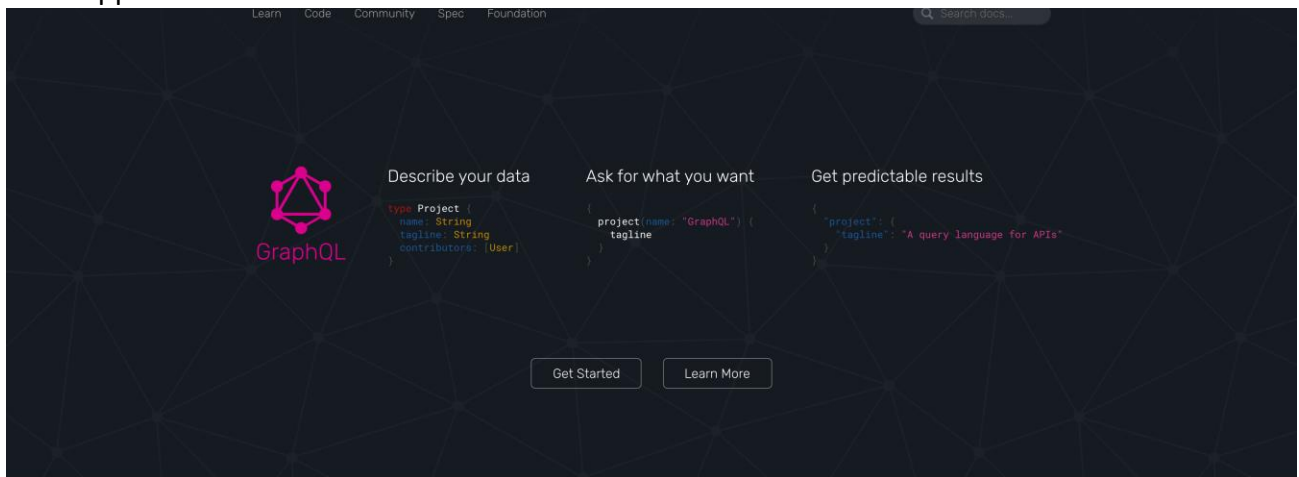
### Lab3 Assignment: Using GraphQL and React Apollo client

**Due: Dec 10, 2018 (11:59 pm)**

This lab assignment covers developing GraphQL Application. This lab assignment is graded based on 20 points and is an **individual effort** (e.g.: no teamwork allowed)

#### Prerequisites

- You must have carefully read concepts of Query and Mutations. You should be able to run the Book App Demo from Canvas.



- You must know Mutation, Query, Mongoose, React Apollo Client.

#### The Assignment

You will be developing two clients and servers during this lab.

On the due date, turn in the following (via canvas):

- A code listing of each of your clients/servers
- Screen captures of each client/server during execution

## Grading

- Late assignments will be accepted, but will be subject to a penalty of -5 points per day late:
- Submissions received at or before the class on the due date can receive maximum

### “HomeAway application” to demonstrate GraphQL Services (10 pts)

Everyone should create the account on HomeAway and see how it functions. This server should perform the following tasks:

- Basic **Traveler and Owner** functionalities:
- Sign up new Traveler/Owner (at least first name, last name, Email and password)
- Sign in existing Traveler/Owner (Encrypt Passwords)
- Sign out. Sign Up should have first name, last name, Email and password.
- List properties on dashboard (Without Images)
- Update Profile of Traveler/Owner (Without Profile Image update).
- Make a Booking.
- View Bookings (Both Owner and Traveler).
- Creation of property is excluded. Use dummy data (Without Images) to list the properties on Home Screen to the traveler and to the owner.(You don't need to create a new database, use existing database of lab2 and do smart work by )
- Use of PassportJs, Kafka and redux is optional. (hint: If you are using PassportJs for authentication then use REST route for login. Do login using REST which will be a non-protected route and then obtain token from this route. Then on GraphQL route(which is a single route) use this token to access your mutation and query calls.)

The Service should take care of exception that means validation is extremely important for this server. **Good exception handling and prototype similar to actual HomeAway application would attract good marks.**

### Client 2 – “HomeAway Client” (8 Points)

Client must include all the functionalities implemented by the web services. Develop the Client using ReactJS and React Apollo Client. Client similar to HomeAway will attract good marks.

### Question (2 points):

How will you enable multi-part data in GraphQL. Discuss two things:

1. An architecture for using multi-part data in GraphQL without using any open source library from Git.
2. 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?

**Create private repository on the GitHub or bitbucket to manage source code for the project. Add description for every commit. Description should consist of one-line overview on what is committed. Include GitHub/bitbucket project link with access credentials in your report.**

**Deliverables Required:**

- ☑ Submit **architecture diagram**.
- ☑ Submissions shall include **source code only** for each client/server pair
- ☑ Project directory must include the group ID/Name (e.g., Lab3-caffeine)
- ☑ Archive the project, and report into one archive file (e.g., zip)
- ☑ Include the Readme file to document the steps to run the application.
- ☑ Project report
  - ☑ Introduction: state your goals, purpose of system,
    - System Design: Describe your chosen system design
    - Results: Screen image captures of each client/server pair during and after run.

For example:

Smith is submitting a project. You have provided the following files and source directory:

•• smith-lab3-report.doc

**Submission**

- ☑ **On-line submission:** Submissions shall be made via Canvas.