Welcome to the CoGrammar Extension: Introduction to GraphQL

The session will start shortly...

Questions? Drop them in the chat. We'll have dedicated moderators answering questions.

CoGrammar

Full Stack Web Development Session Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: **Questions**

Full Stack Web Development Session Housekeeping cont.

- For all **non-academic questions**, please submit a query: www.hyperiondev.com/support

- Report a **safeguarding** incident: www.hyperiondev.com/safeguardreporting

- We would love your **feedback** on lectures: Feedback on Lectures

Skills Bootcamp
8-Week Progression Overview

Fulfil 4 Criteria to Graduation

✅ Criterion 1: Initial Requirements

Timeframe: First 2 Weeks
Guided Learning Hours (GLH): Minimum of 15 hours
Task Completion: First four tasks

Due Date: 24 March 2024

✅ Criterion 2: Mid-Course Progress

60 Guided Learning Hours

Data Science - **13 tasks**
Software Engineering - **13 tasks**
Web Development - **13 tasks**

Due Date: 28 April 2024

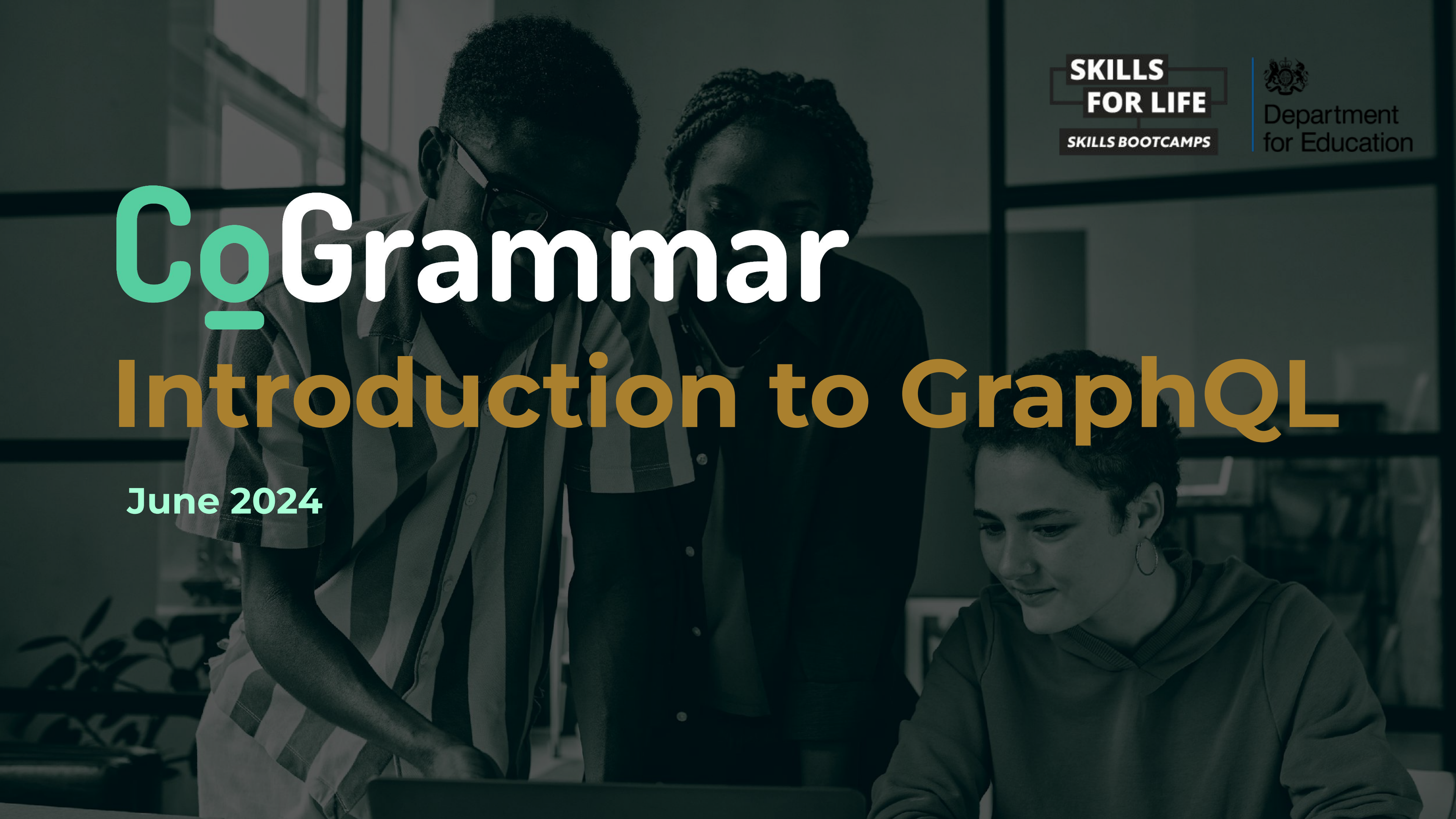
Skills Bootcamp Progression Overview

✅ Criterion 3: Course Progress

Completion: All mandatory tasks, including Build Your Brand and resubmissions by study period end
Interview Invitation: Within 4 weeks post-course
Guided Learning Hours: Minimum of 112 hours by support end date (10.5 hours average, each week)

✅ Criterion 4: Demonstrating Employability

Final Job or Apprenticeship Outcome: Document within 12 weeks post-graduation
Relevance: Progression to employment or related opportunity



SKILLS FOR LIFE

SKILLS BOOTCAMPS

Department for Education

CoGrammar

Introduction to GraphQL

June 2024

Learning Objectives

- ❖ Explain the differences and advantages of GraphQL over REST.
- ❖ Set up a basic GraphQL server using Node.js.
- ❖ Create GraphQL queries and mutations.
- ❖ Integrate a GraphQL API with a React application using Apollo Client.

What is GraphQL?

- A query language for your API and a server-side runtime for executing queries.
- Developed internally by Facebook in 2012 before being publicly released in 2015.

Key Concepts of GraphQL

- **Queries, Mutations, and Subscriptions**
  - Queries are used to fetch data.
  - Mutations are used to modify server-side data.
  - Subscriptions allow clients to subscribe to real-time messages through a web socket.
- **Schema and Types**
  - The schema defines the capabilities of the API by specifying how a client can fetch and update data.
  - Types are custom data structures (e.g., String, Int, Boolean, custom objects).

Advantages of GraphQL Over REST

- Single Endpoint:
  - Uses a single endpoint unlike REST which typically uses multiple endpoints.
- Fetch What You Need:
  - Clients have the power to ask for exactly what they need, no more, no less.
- Less Overfetching and Underfetching:
  - Solves issues common in REST where you fetch too much or too little data.
- Easier to Aggregate Data from Multiple Sources:
  - Allows a server to easily delegate to other services.

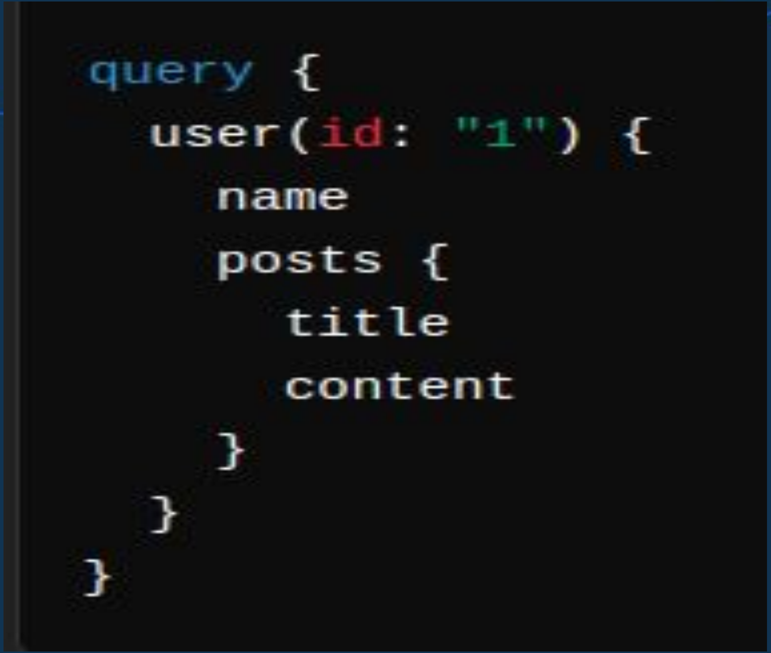
GraphQL vs REST

- Efficiency in Data Retrieval:
  - REST:
    - RESTful APIs typically require clients to make multiple round-trip requests to fetch complex, nested data. This is because each resource, like a user or their posts, is accessed via a separate endpoint.
    - For example, if you need to retrieve a user and their posts, you might first make a call to /users/{id} to get user details and then a call to /users/{id}/posts to fetch their posts. This increases the total number of requests and data transmitted over the network, potentially leading to slower performance, especially in mobile or low-bandwidth environments.

GraphQL vs REST

- GraphQL:
  - GraphQL allows clients to request exactly the data they need in a single request. Clients can specify nested resources within the same query, reducing the number of requests made to the server.

CoGrammar



```
query {
  user(id: "1") {
    name
    posts {
      title
      content
    }
  }
}
```

- This single-query structure drastically cuts down on the network traffic and load times, as the server processes one query that includes all the necessary information, eliminating the overhead of multiple API calls.

Let’s Breathe!

Let’s take a small break before moving on to the next topic.

CoGrammar

Setting Up a GraphQL Server

- Initialize a Node.js Project:
  - First, create a new directory for your project and initialize a Node.js application:
    - mkdir graphql-react-tutorial
    - cd graphql-react-tutorial
    - npm init -y
  - Install dependencies:
    - npm install apollo-server-express express graphql

Create a Simple GraphQL Server

- Create a file named `server.js`

Set up a React Application

- ❖ npx create-react-app client
- ❖ cd client
- ❖ npm start

- ❖ Install Apollo Client:
  - ➢ npm install @apollo/client graphql

Integrate Apollo Client

❖

CoGrammar

Fetch Data with Apollo Client

❖

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar