# 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.

- 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**

CoGrammar

# Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:



Ian Wyles
Designated Safeguarding Lead

Simone Botes

Nurhaan Snyman

Rafiq Manan

Ronald Munodawafa

Tevin Pitts

**Scan to report a safeguarding concern**



or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar    HyperionDev

# Polls

Please have a look at the poll notification and select an option.

**Which MongoDB driver do you use with Node.js**

A.   Official MongoDB Node.js Driver
B.   Mongoose (ODM)
C.   Other (please specify)

CoGrammar

# Polls

Please have a look at the poll notification and select an option.

**How do you manage MongoDB connection strings in your Node.js app?**

A.  Hardcoded in the application
B.  Environment variables
C.  Secret management tools (AWS Secrets Manager)
D.  Other (Please specify)

CoGrammar

# Databases

**A large container of data with the ability to order the data in multiple ways, while providing access to the data itself.**

❖ **Data** refers to **raw, unprocessed** facts. Once data has been processed, we call it **information**.

❖ The production of accurate, timely and relevant information is the key to good **decision-making**, which is the key to a **business' survival** in a competitive global environment.

❖ Timely and useful information requires accurate data, which must be captured properly and stored in a format that is easy to access and process

CoGrammar

# DBMS

❖ A database is usually controlled by a database engine, commonly known as a **Database Management System (DBMS).**

❖ DBMSs serve as a **tool** between a user and their data, **organising** and **cataloging** the data for **quick and easy retrieval.**

❖ The data and the DBMS, and the applications associated with them are referred to as a **database system**, usually shortened to **database.**
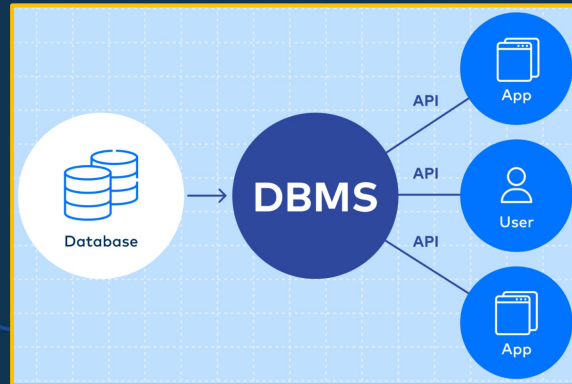
# DBMS

❖ The **advantages** of the DBMS are:

➢ **Data sharing:** Better access to more, better managed data across applications and users.

➢ **Data integration:** Unified view of well-managed data combined from multiple sources.

➢ **Data consistency:** Minimise risk of different versions of the same data stored in different places.

➢ **Data access:** The DBMS makes it possible to produce quick answers to spur-of-the-moment requests for data.

CoGrammar

# Relational Databases

**Any database system that allows data to be associated and grouped by common attributes.**

❖ Relational databases are comprised of a number of tables (**relations**), within each are:
  ➢ Rows also known as records or tuples
  ➢ Columns also known as attributes or fields

❖ Each record is identified with a **unique key**, known as the **primary key**.

❖ Records from one table can be references in other tables using their key, in this case they are called **foreign keys**.

❖ Each table/relation represents one **"entity type"**.

CoGrammar

# NoSQL Databases

❖ The performance of relational databases degrades as the volume of data increases.

❖ Web applications usually have to store massive amounts of data, so NoSQL databases were developed to improve performance.

❖ NoSQL databases have the following characteristics:

  ➢ Not based on the relational model.
  ➢ Support distributed database architectures.
  ➢ High scalability, high availability and fault tolerance.
  ➢ Support large amounts of sparse data.
  ➢ Geared toward performance rather than transactional consistency

CoGrammar

# MongoDB

A document store and NoSQL database, made up of collections and documents.

❖ Collections: A group of documents, similar to an entity or table in RDBms.

❖ Documents: Equivalent to a record in an RDB (or row in a RDB table).

❖ MongoDB uses **Binary JSON** (BSON) which uses JSON files and stores **type** information, which makes it **quicker** and **more efficient** to use.

❖ If a user wants to access, add, or change any information that needs to persist, they will need access to the MongoDB database.

❖ Clients interact with a web server that runs Node.js, which makes use of **MongoDB drivers** to communicate with MongoDB.

CoGrammar

# Mongoose

**A library that makes working with the MongoDB driver simpler.**

1. Install Mongoose using NPM:

   a. `npm install mongoose`

2. Create a schema which outlines the data in our database and how it is organised and structured.

3. Create a controller file to perform data manipulation.

4. Connect to the database and execute operations.

CoGrammar

```javascript
//MongoDB Connection
/**
 * @NOTE - Insert username and password accordingly
 */
const username = process.env.MONGO_USER
const password = process.env.MONGO_PASSWORD

const uri = `mongodb+srv://${username}:${password}@cluster0.fhgaqrt.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0`;
console.log(uri)

const clientOptions = { serverApi: { version: '1', strict: true, deprecationErrors: true } };

mongoose.connect(uri, clientOptions)
.then(()=>{
    console.log('Connection to mongodb done successfully...')
})
.catch((error)=>{
    console.log('Error connecting to mongodb', error)
})
```

# Questions and Answers

CoGrammar

# Thank you for attending

**SKILLS FOR LIFE**
*SKILLS BOOTCAMPS*

Department for Education

CoGrammar