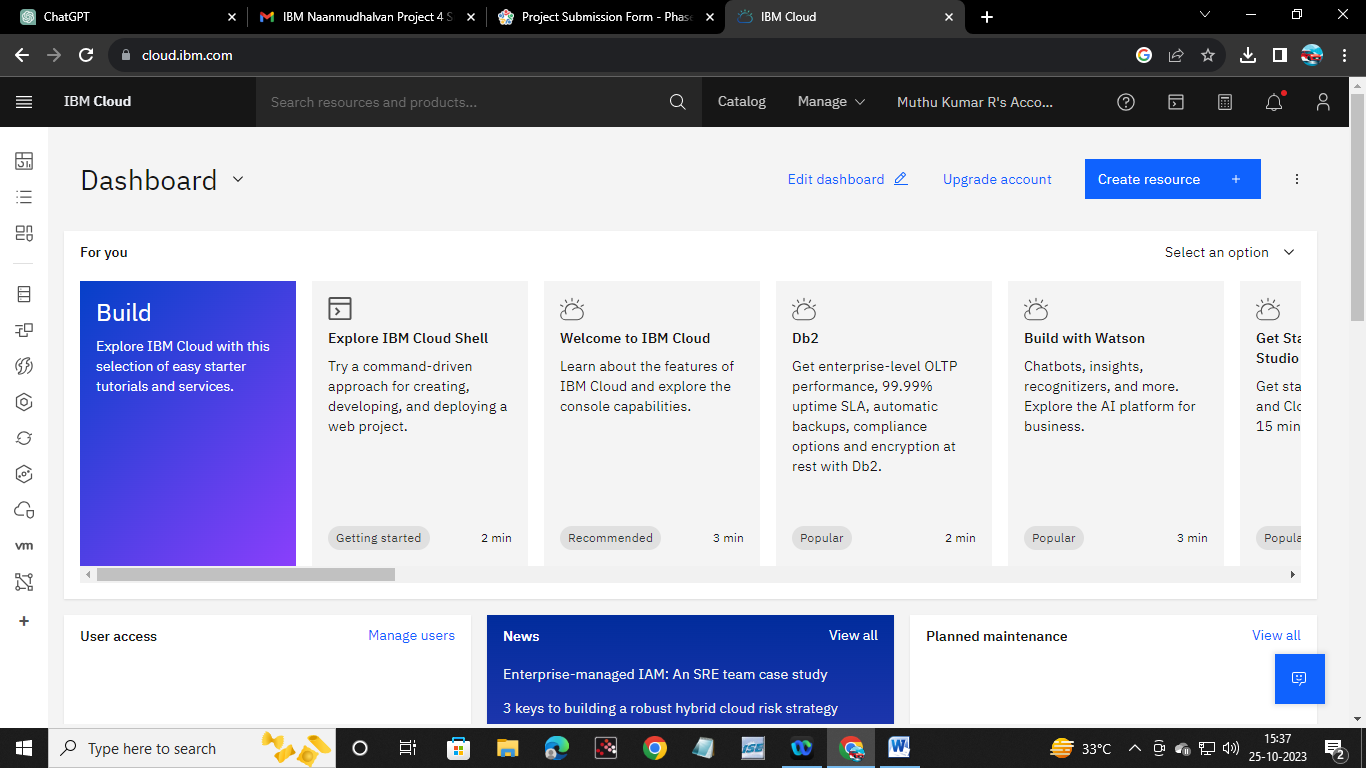
**PHASE 3:**

**DEVELOPMENT PART 1:**

1. **TO CREATE AN IBM CLOUD ACCOUNT:**



2.) **choose the appropriate database service:**

**-> Install MongoDB:**

First, you need to install MongoDB on your system. MongoDB offers various installation methods for different operating systems

# Import the MongoDB GPG key

wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -

# Add the MongoDB repository

echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-5.0.list

# Update your package repository

sudo apt update

# Install MongoDB ; sudo apt install -y mongodb-org

* **Start the MongoDB Service:**

# Start MongoDB

sudo systemctl start mongo

# Enable MongoDB to start on boot

sudo systemctl enable mongod

* **Create a Database:**

use <your-database-name>

**3.) Develop queries or scripts to explore and analyze the selected dataset:**

1. **Basic Query:** Find all documents in the collection

db.mycollection.find({})

**2. Filter Documents:**

* Find documents that match a specific condition, e.g., where the "age" field is greater than 30:

db.mycollection.find({ age: { $gt: 30 } })

**3.)Sorting:**

* Sort the documents by a specific field (e.g., name) in ascending order:

db.mycollection.find().sort({ name: 1 })

4.) **Update Documents:**

* Update documents that match a specific condition (e.g., update the age for a person with a specific name):

db.mycollection.updateOne({ name: "John" }, { $set: { age: 35 } })

**4.)Perform basic data cleaning and transformation as needed.**

a. **Data Filtering**:

* . If you want to remove specific documents based on a condition, you can use the **deleteMany** method

b. **Data Modification**:

* To update or modify existing documents, use the **updateOne** or **updateMany** methods.

db.your\_collection.updateMany(

{ field\_to\_match: "value\_to\_match" },

{ $set: { field\_to\_update: "new\_value" } }

)

c. **Data Transformation**:

* You can use the aggregation framework to perform more complex data transformations. For example, you can reshape documents, calculate new fields, or aggregate data.

db.your\_collection.aggregate([

{

$match: { someField: "value" }

},

{

$group: {

\_id: "$grouping\_field",

total: { $sum: "$numeric\_field" }

}

}

])