# Lab 5 [Getting started with MongoDB]

|  |  |
| --- | --- |
| Lab Title: | Getting started with MongoDB |
| **Expected duration** (hours): | **1 hour 45 mins** |

|  |
| --- |
| Objectives |
| Understanding how to run MongoDB locally |
| Learn how to design database schema |
| Learn how to create collections and documents |

|  |
| --- |
| Requirements |
| A laptop or a desktop with Windows or Mac as an operating system |
| Visual studio code or something similar as a text editor |
| A modern web browser like chrome |

## Lab Instructions:

### **Lab Outline**

1. Introduction to the Product App
2. Designing the Database Schema
3. Creating Collections and Documents
4. Implementing CRUD Operations

### **1. Introduction to the Product App**

Before you start developing, let's make sure you have a free MongoDB atlas account in the cloud.

* Go to the MongoDB Atlas [website](https://www.mongodb.com/cloud/atlas): Open your web browser and navigate to MongoDB Atlas.
* Click on "Try Free": You'll be redirected to the sign-up page.
* Create an account: You can sign up using your Google account or by filling in your email, name, and password. Click on "Get started free".
* Welcome to Atlas: Once signed in, you'll be greeted with the Atlas onboarding screen.
* Build a Cluster:
  1. Select the cloud provider and region for your cluster. For the free tier, choose a region closest to you.
  2. Choose the cluster tier. Select the free tier option, which is usually labelled as M0 Sandbox.
  3. Click "Create Cluster".
* As part of onboarding, ideally your ip-address should be added and also there will be a cluster and database user been created. If not, please reach out to your instructor to troubleshoot the scenario.
* Connect to your cluster via MongoDB shell -
  1. Go back to the "Clusters" page from the left-hand sidebar.
  2. Find your cluster and click the "Connect" button.
  3. Choose a connection method. You can connect using MongoDB Compass (GUI), the MongoDB shell, or directly from your application.
  4. For simplicity, let's choose "Connect with MongoDB Shell".
  5. A connection string will be provided. It will look something like this:
     1. mongodb+srv://<username>:<password>@cluster0.mongodb.net/<dbname>?retryWrites=true&w=majority
     2. Replace <username>, <password>, and <dbname> with your database user credentials and database name.
  6. Open your terminal and type the following command to connect to your Atlas cluster using MongoDB Shell:
     1. mongo mongodb+srv://<username>:<password>@cluster0.mongodb.net/<dbname>?retryWrites=true&w=majority
  7. If your connection is successful, you'll see the MongoDB shell prompt.
  8. You can now start creating databases, collections, and documents.

The Product app allows users to list products and create new products. Each product has the following attributes:

* name (string): The name of the product.
* description (string): A brief description of the product.
* price (number): The price of the product.
* category (string): The category to which the product belongs.
* createdAt (date): The date when the product was added.

### **2. Designing the Database Schema**

In MongoDB, data is stored in collections of documents. Each document is a JSON-like object.

For the Product app, we will create a collection named products with the following schema:

{  
 "name": "string",  
 "description": "string",  
 "price": "number",  
 "category": "string",  
 "createdAt": "date"  
}

### **3. Creating Collections and Documents**

#### **Step 1: Start MongoDB**

* Ensure that your MongoDB server is running. If using MongoDB Atlas, make sure you are connected to your cluster.

#### **Step 2: Create the products Collection**

* Open the MongoDB shell (or use MongoDB Compass) and create the products collection.
* Using MongoDB Shell:

use productApp  
db.createCollection("products")

#### **Step 3: Insert a Sample Document**

* Insert a sample product document into the products collection.
* Using MongoDB Shell:

db.products.insertOne({  
 name: "Laptop",  
 description: "A high-performance laptop",  
 price: 1500,  
 category: "Electronics",  
 createdAt: new Date()  
})

### **4. Implementing CRUD Operations**

#### **Create: Add a New Product**

* Insert a new product into the products collection.
* Using MongoDB Shell:

db.products.insertOne({  
 name: "Smartphone",  
 description: "A latest model smartphone",  
 price: 800,  
 category: "Electronics",  
 createdAt: new Date()  
})

#### **Read: List All Products**

* Retrieve all products from the products collection.
* Using MongoDB Shell:

db.products.find().pretty()

#### **Update: Modify an Existing Product**

* Update the price of a product.
* Using MongoDB Shell:

db.products.updateOne(  
 { name: "Laptop" },  
 { $set: { price: 1400 } }  
)

#### **Delete: Remove a Product**

* Delete a product from the collection.
* Using MongoDB Shell:

db.products.deleteOne({ name: "Smartphone" })