MongoDB- Storing Reviews

Tutorial – 3, Download, Set up and Implementation

CSP 584 – Enterprise Web Application

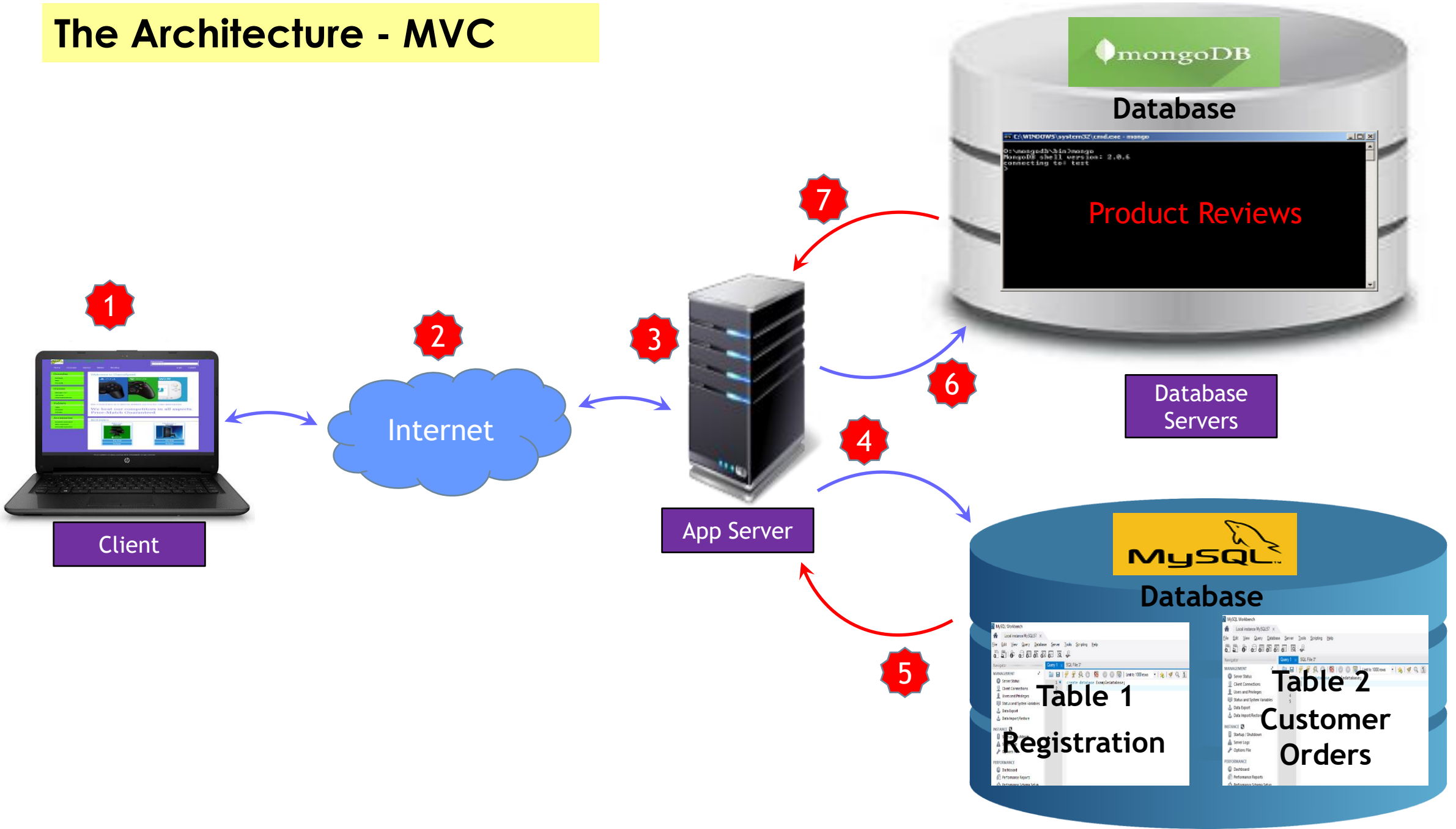
Dr. Atef Bader

Illinois Institute of Technology

- Presentation By

Pradeep Kumar Lokesh

The Architecture - MVC



1. Mongo DB - Overview

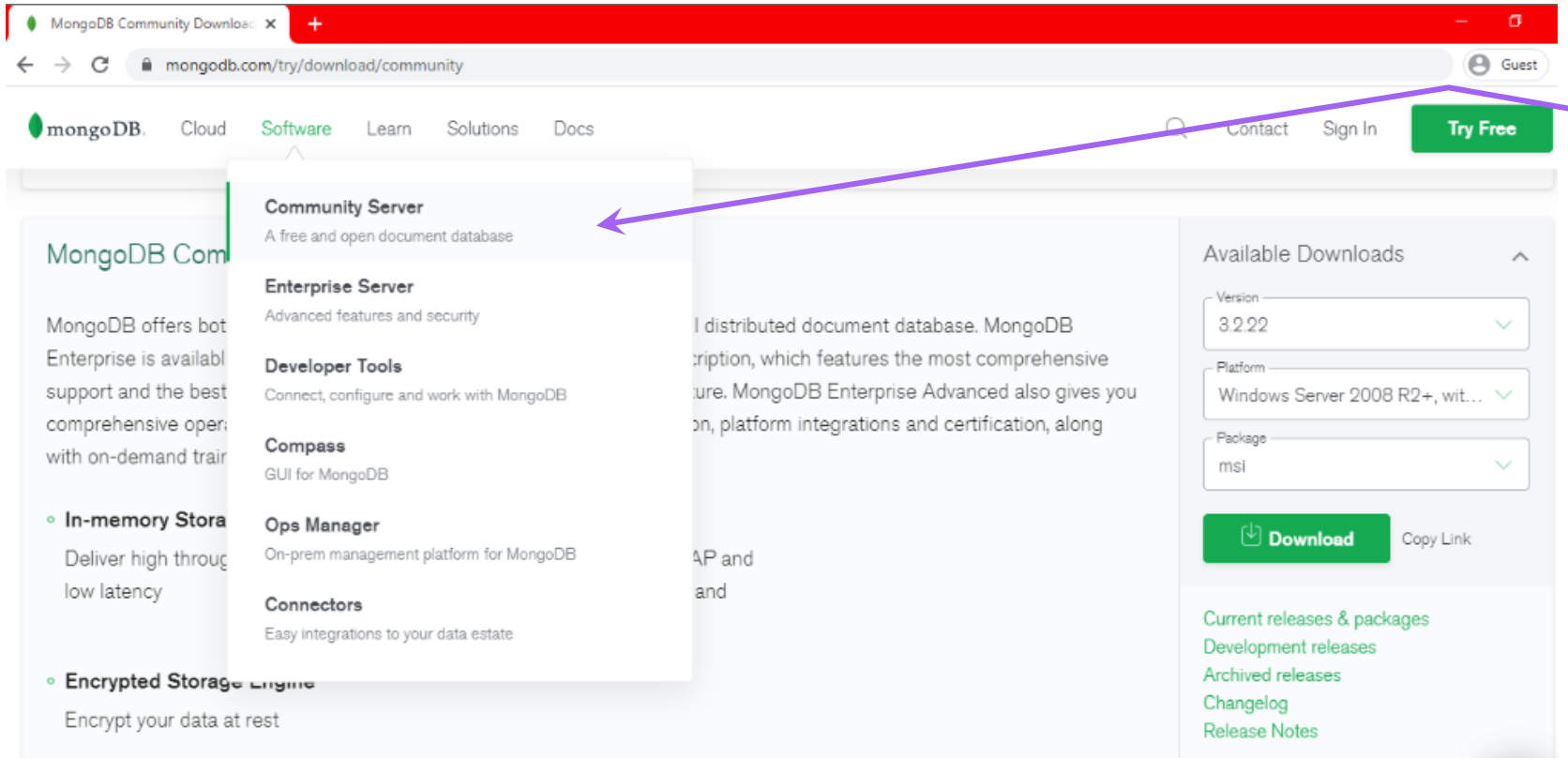
- Mongo DB is a cross platform, document oriented database
- Mongo DB works on the concept of Collections and documents

- Terminologies:
  - Database: This is the physical container for the collections
  - Collection: Collection is a group of Mongo DB documents
  - Document: Document is a set of key – value pairs

- Advantages:
  - Schema-less: The number of fields, content and size of the document can vary from one another
  - Scalability: Mongo DB is easy to scale

2. Mongo DB - Download

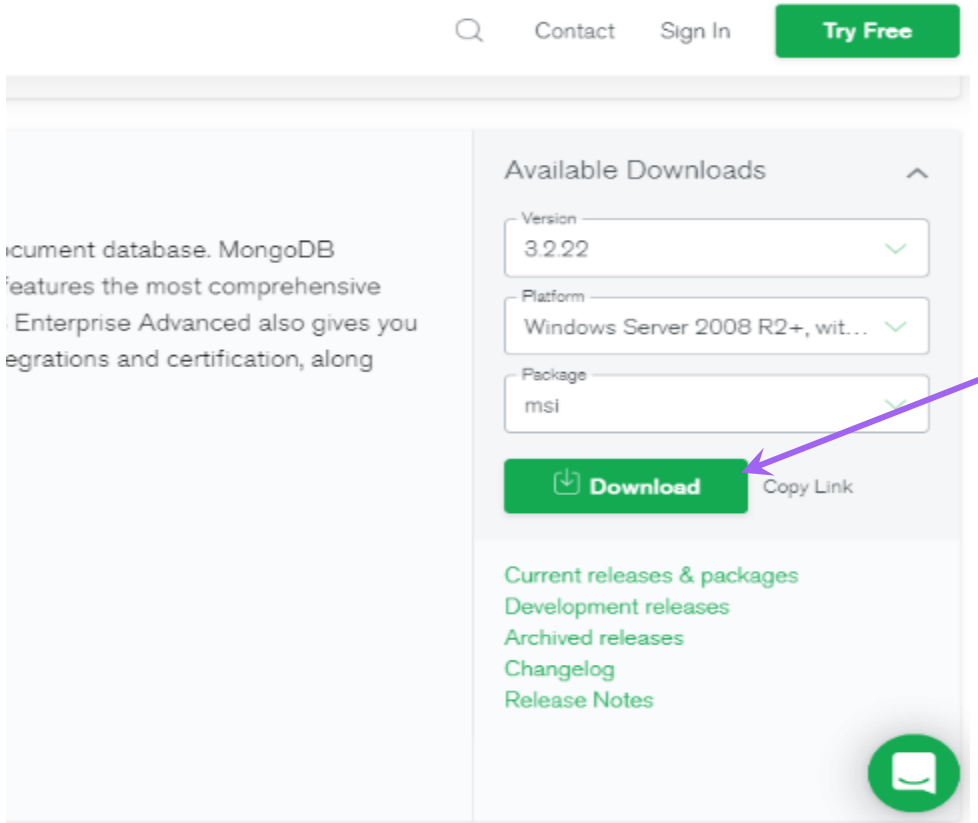
- Go to https://www.mongodb.com/ and click on the 'Products' button and select Mongo DB Server under the software to download Mongo DB (Direct Link : https://www.mongodb.com/try/download/community)



Click on the Community Server for downloading MongoDB

2. Mongo DB - Download

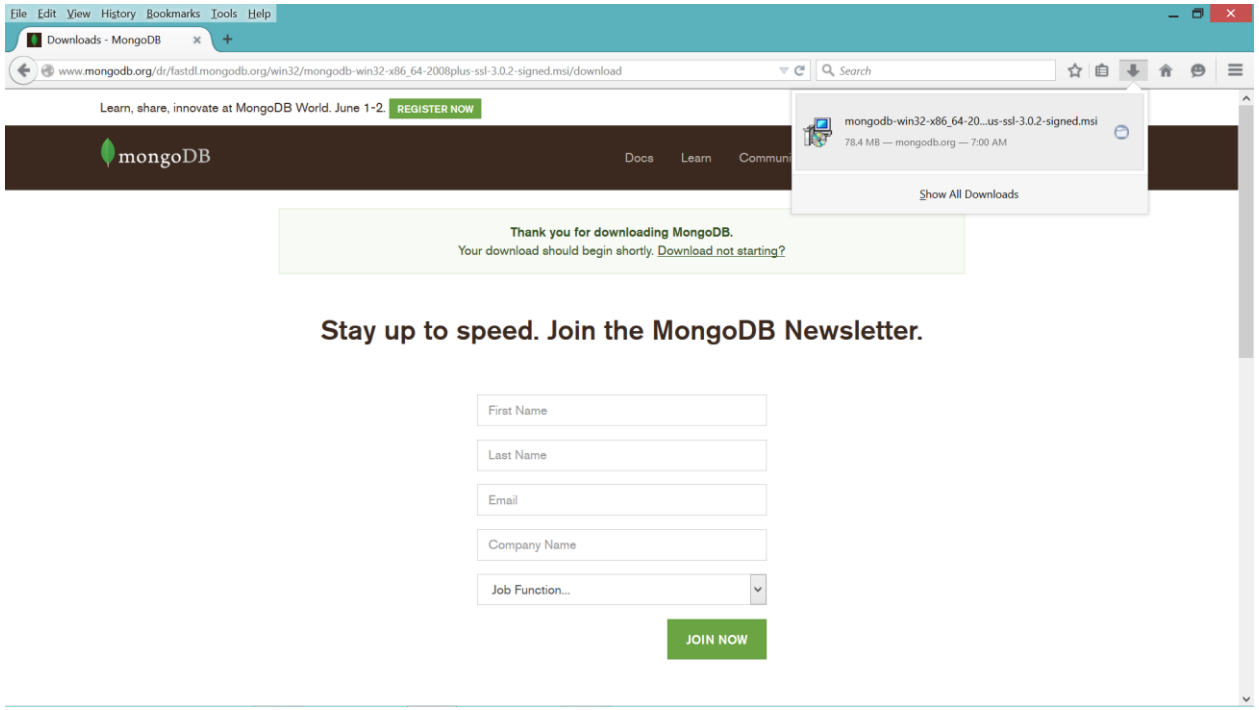
- Select the Version, Platform as Windows and the version as ‘Windows Server 64 – bit 2008 R2 64 bit and later with SSL support x64’
- Click on the ‘Download’ to begin the download



Click on the Download Button

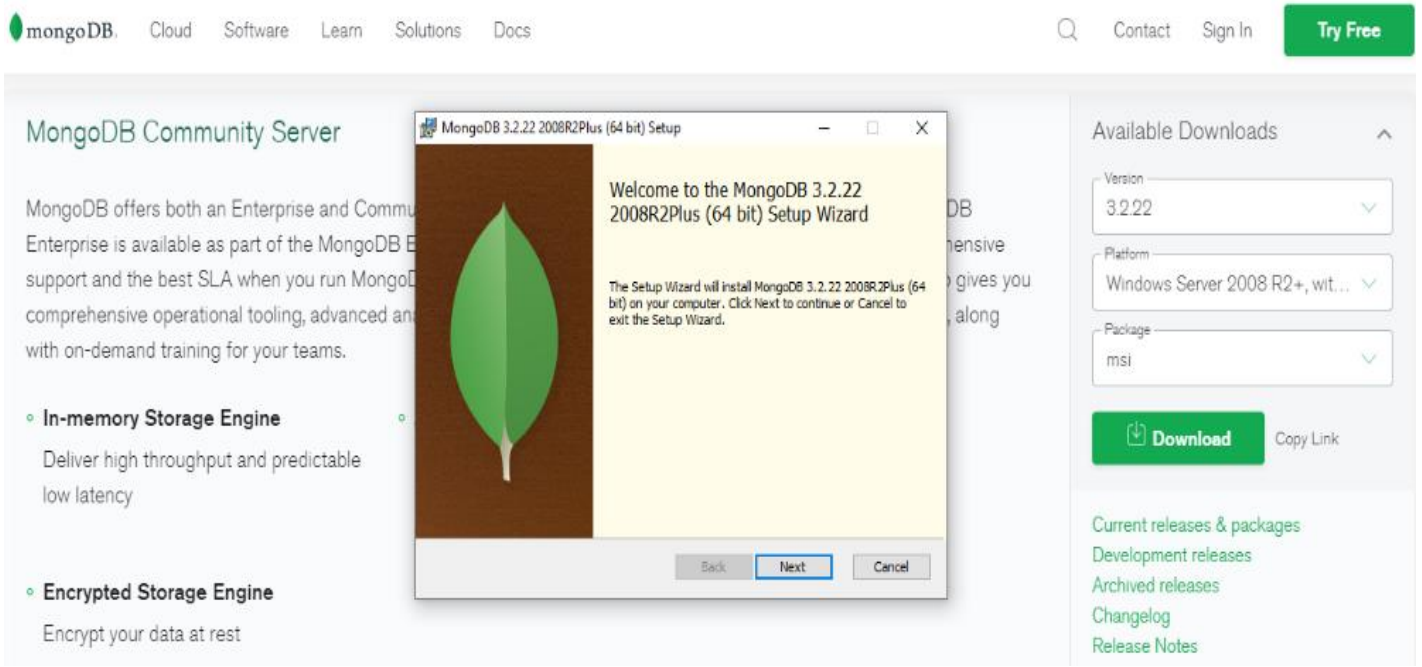
2. Mongo DB - Download

- Please note the location of the folder where MongoDB is being downloaded



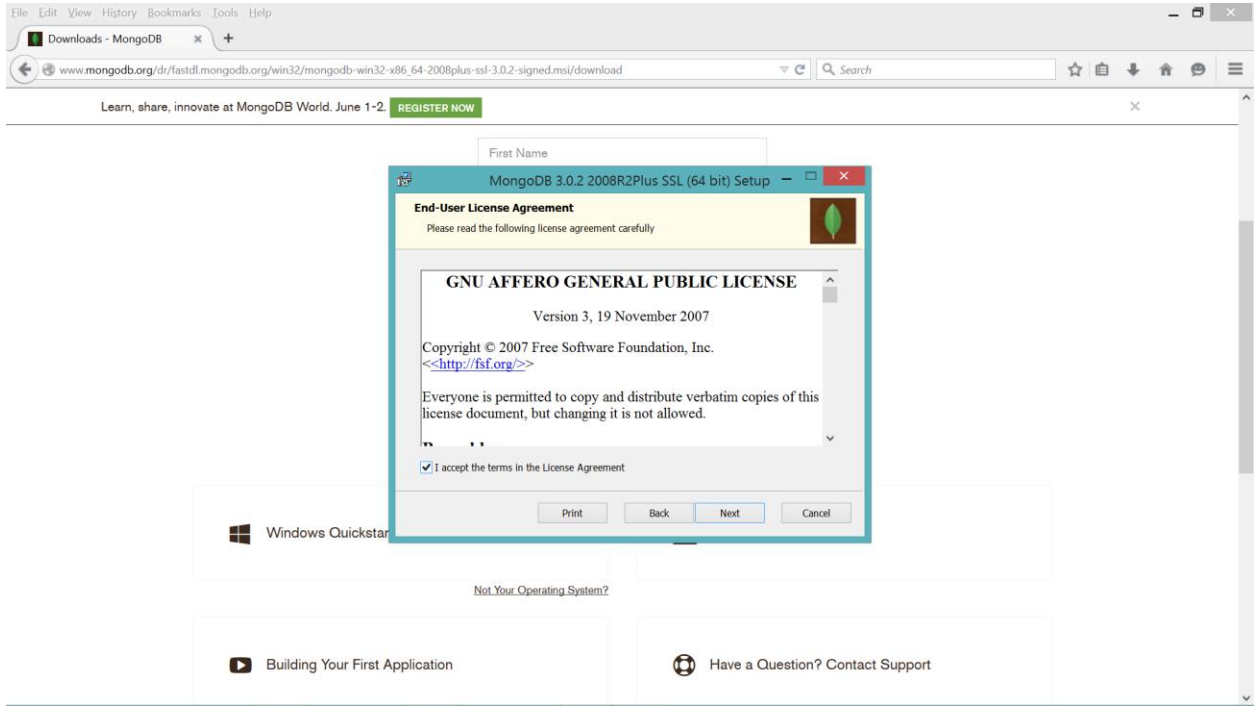
3. Mongo DB - Setup

- To start the installation, go to the folder where MongoDB has been downloaded and double click on the installation file
- This should open the MongoDB setup wizard as shown below
- Click on ‘Next’ to proceed with the installation



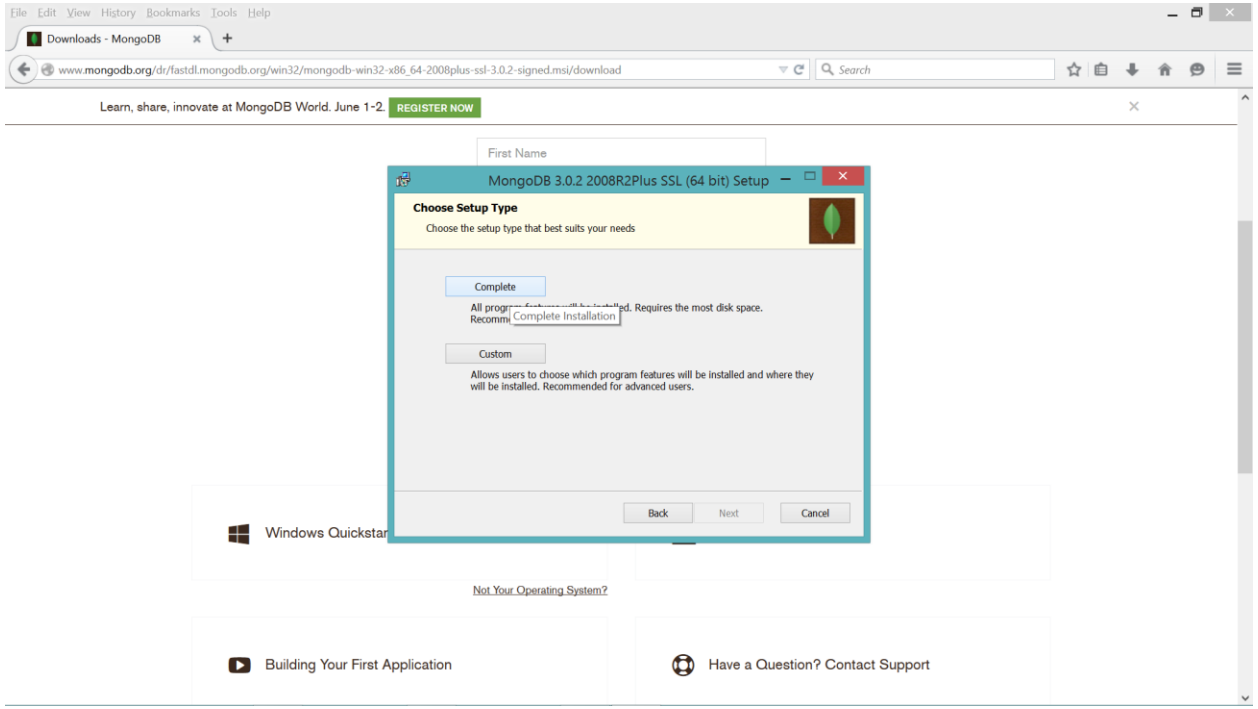
3. Mongo DB - Setup

- Accept the license agreement and proceed by clicking on ‘Next’



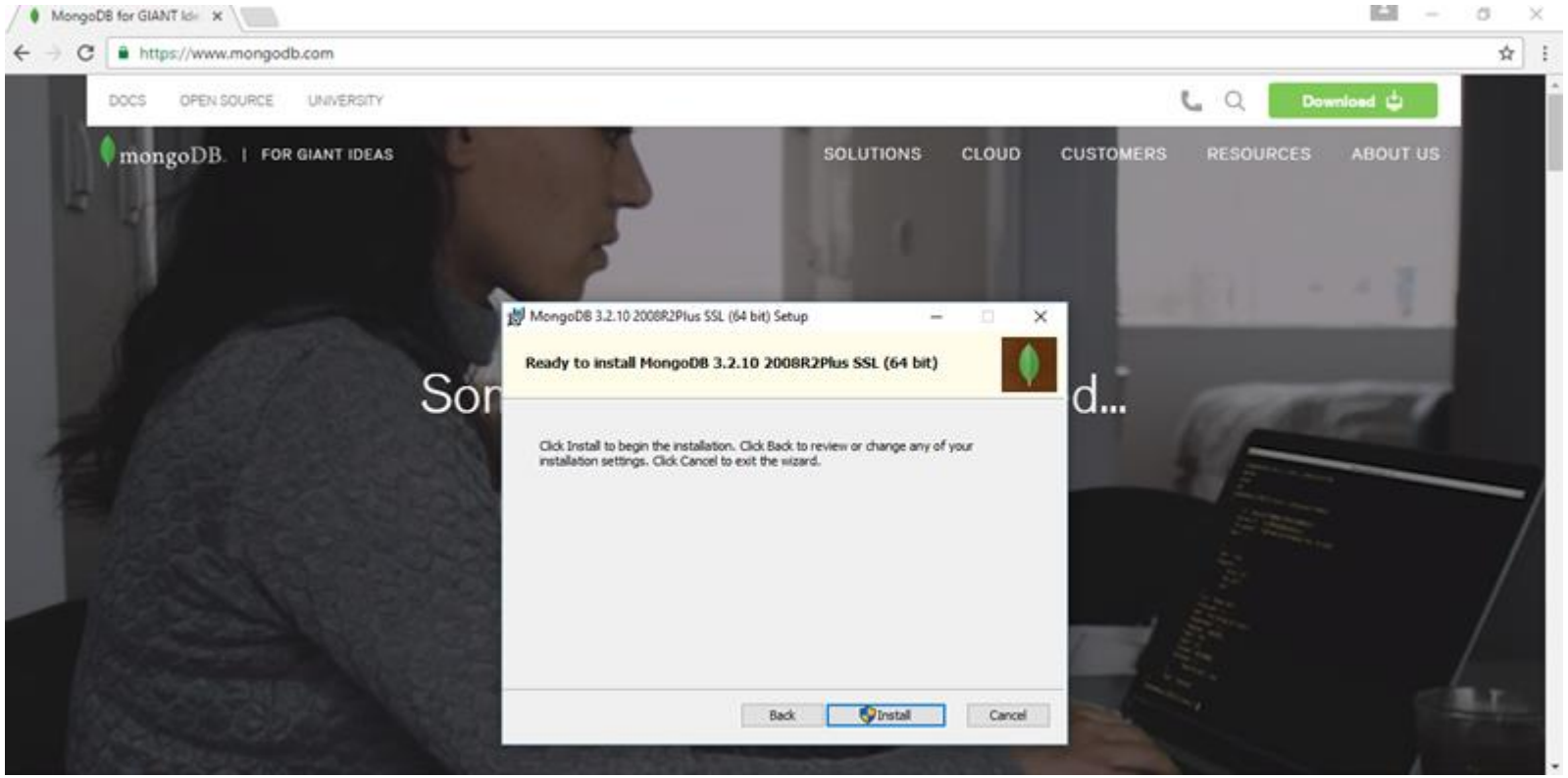
3. Mongo DB - Setup

- Select the setup type as ‘Complete’ and then click on ‘Next’
- Since we are at the beginners level with MongoDB, hence, it is recommended that you select the setup type as ‘Complete’



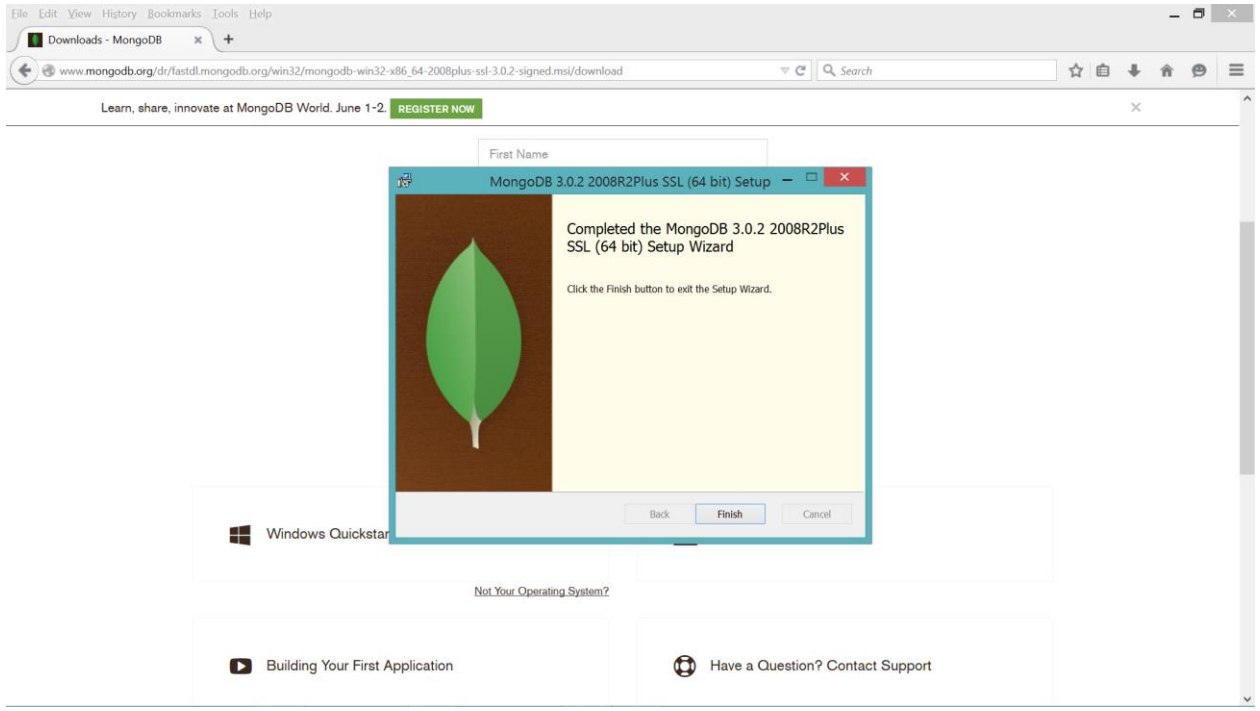
3. Mongo DB - Setup

- Click on install to install Mongo database



3. Mongo DB - Setup

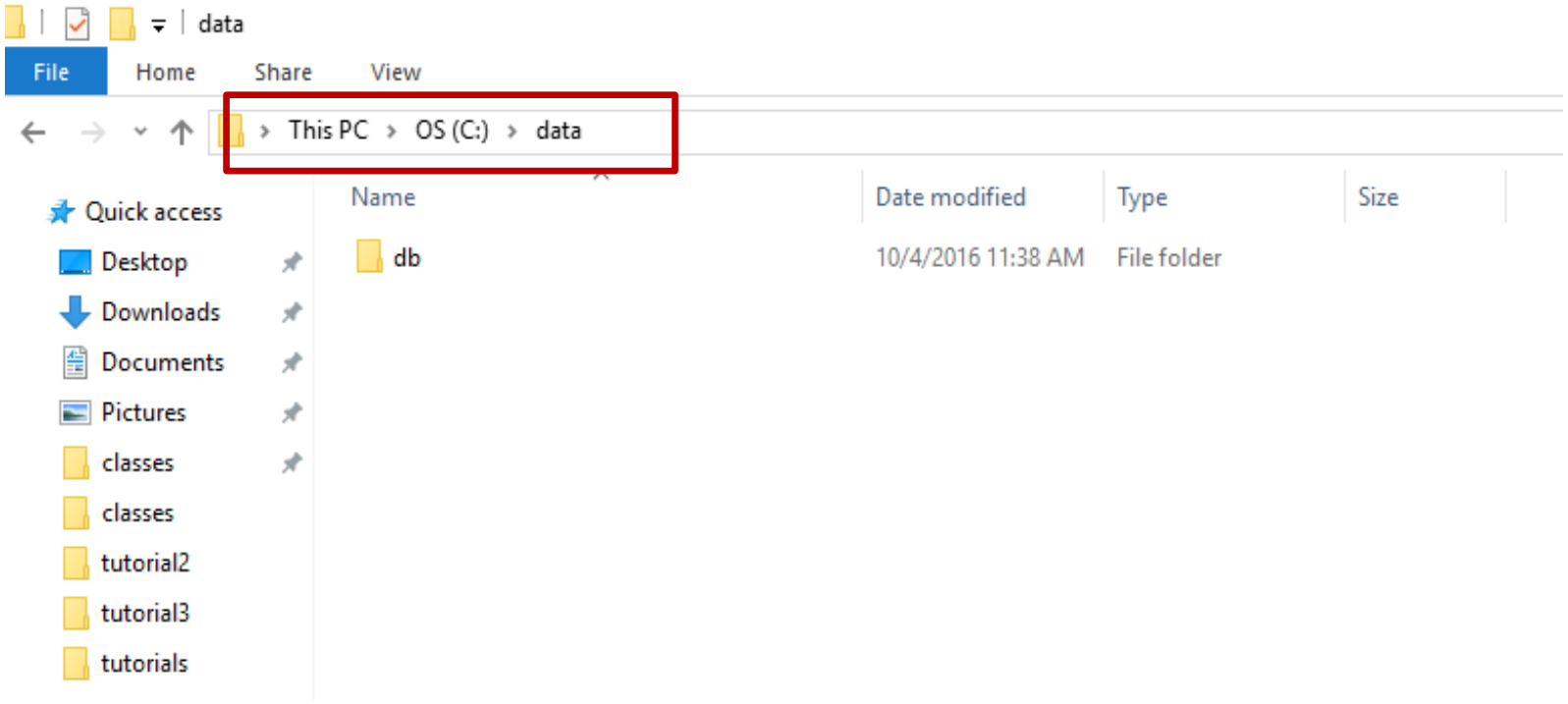
- Once the installation is complete, click on ‘Finish’ to complete the process



4. Mongo DB – Startup Instructions

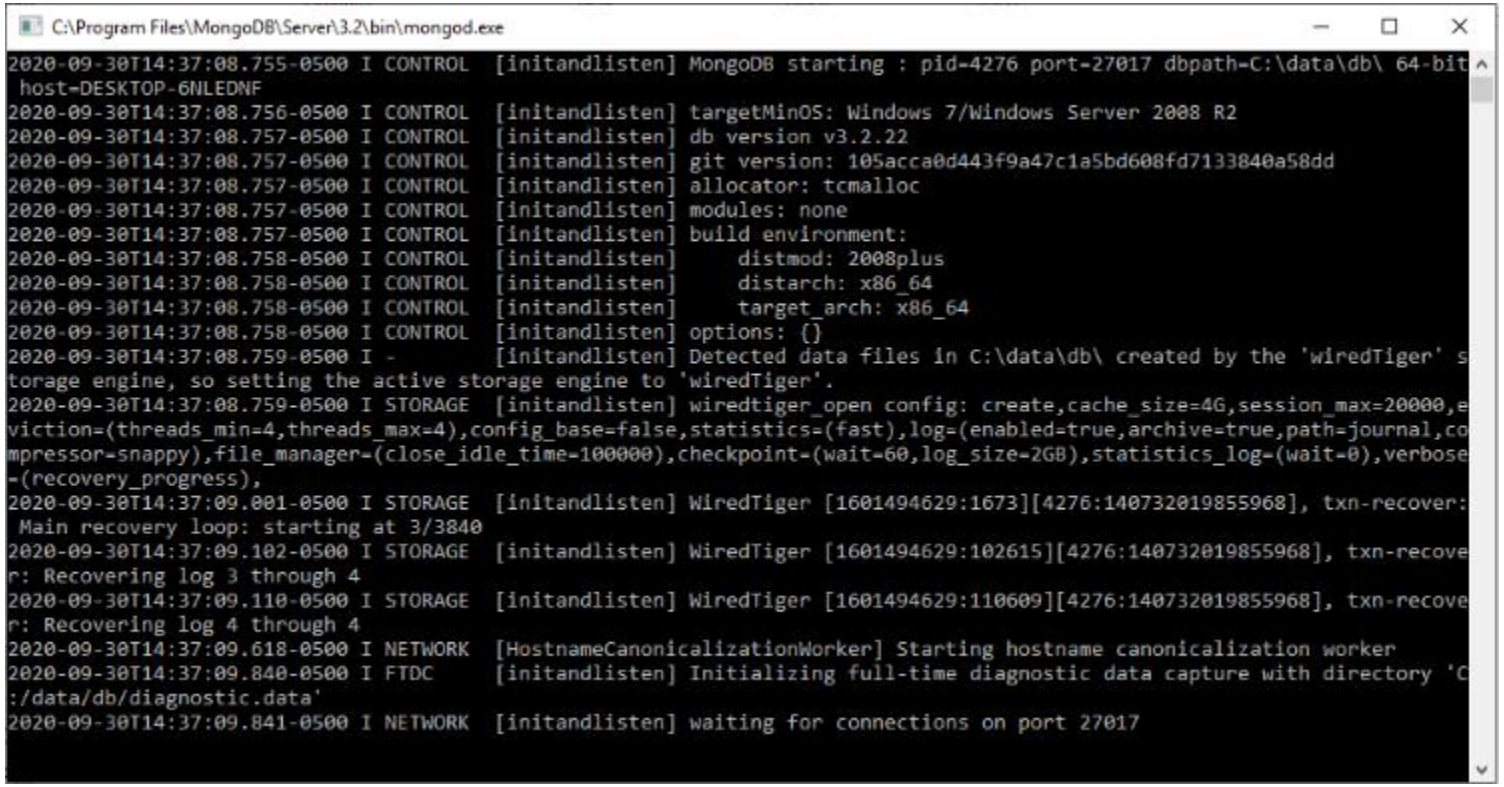
Create a data and db folder inside C drive as c:\data\db

Make sure that you directly create data\db folder inside c drive only



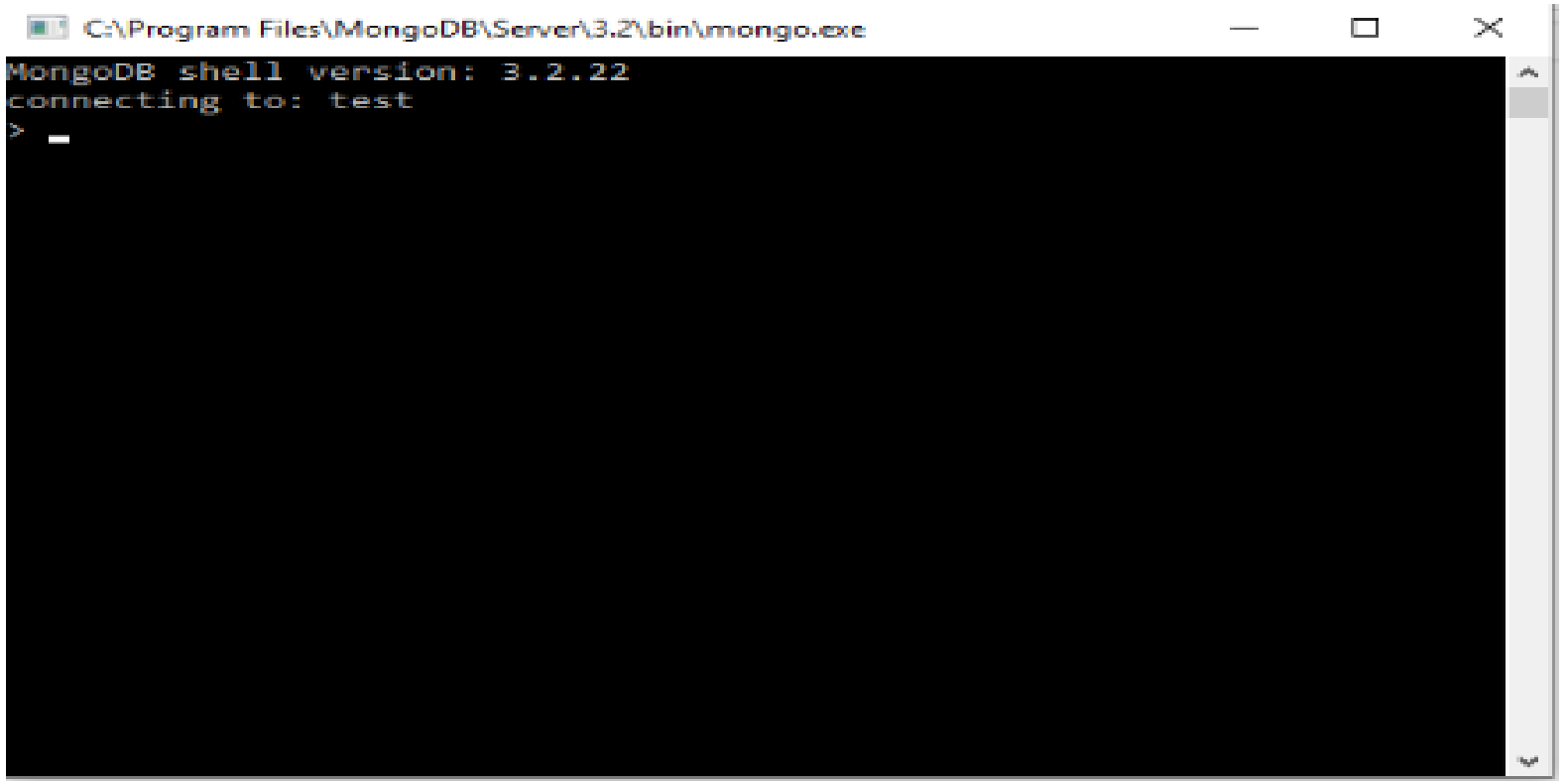
4. Mongo DB – Startup Instructions

- To start Mongo DB, open command prompt and enter the command ‘mongod’ .
- Mongo DB is usually installed under C:/Program Files/MongoDB
- To start Mongo DB server process , locate the “mongod.exe” stored in C:\Program Files\MongoDB\Server\3.2\bin and click it.



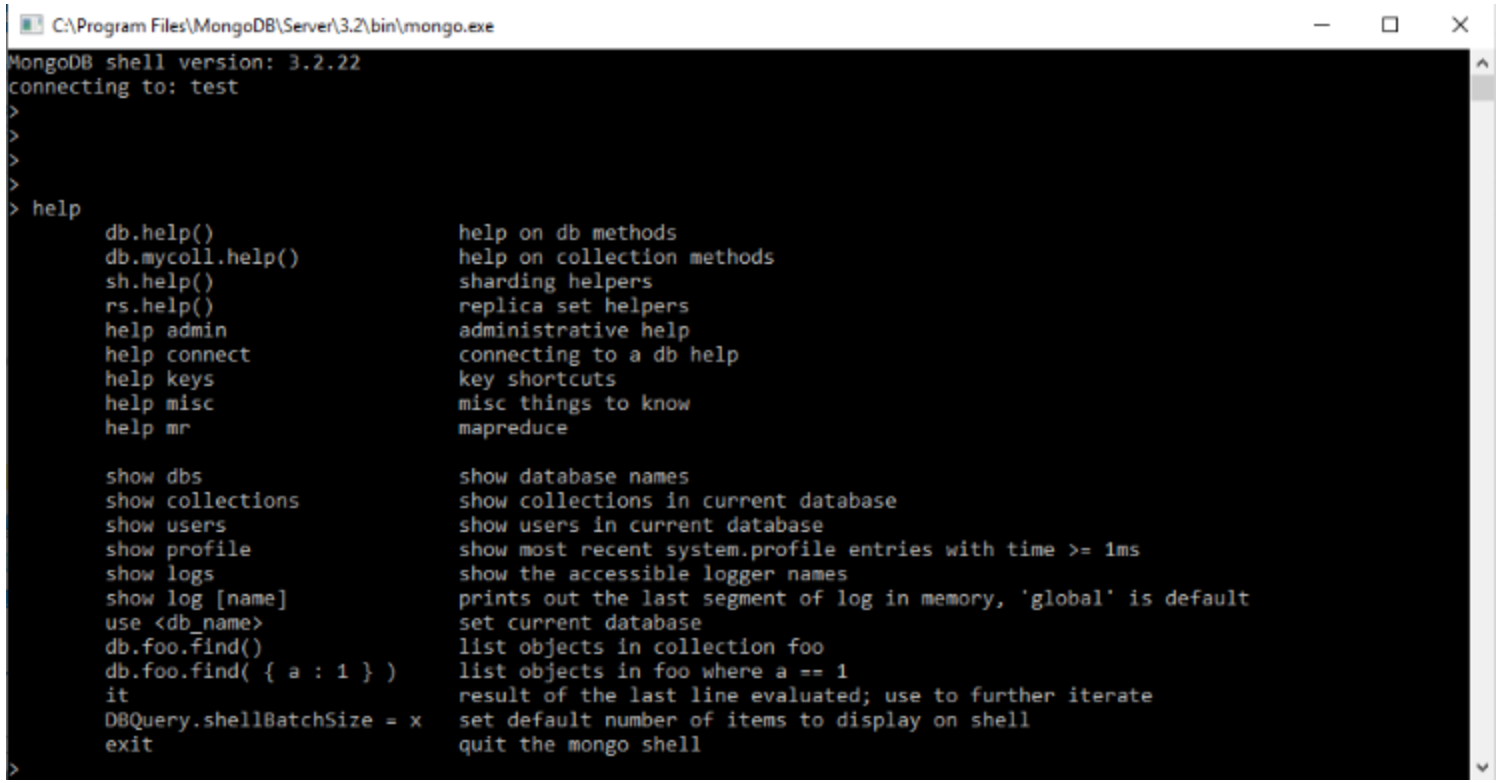
4. Mongo DB – Startup Instructions

- To start Mongo shell, open command prompt and enter the command ‘mongo’
- Mongo DB is usually installed under C:/Program Files/MongoDB
- To start Mongo shell, locate the “mongo.exe” stored in C:\Program Files\MongoDB\Server\3.2\bin and click it.



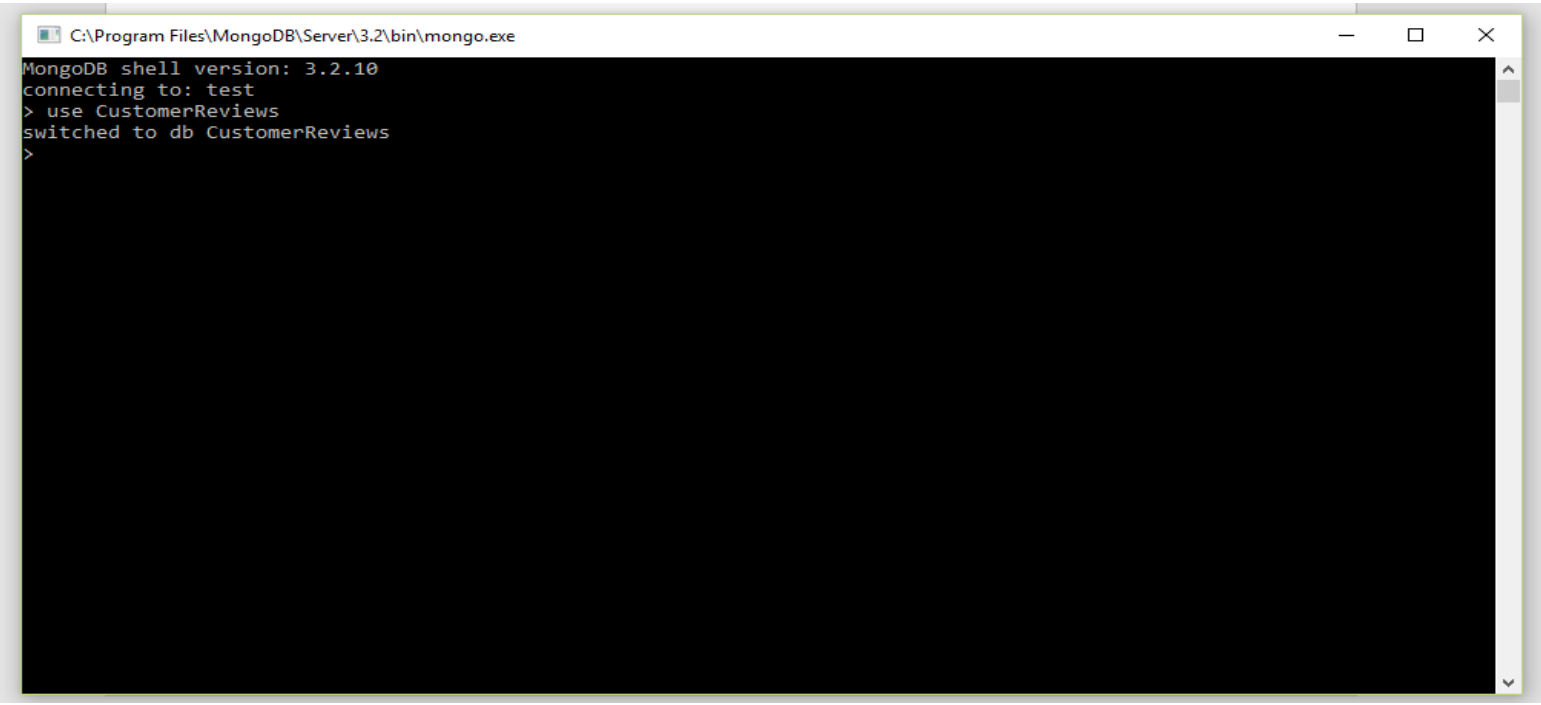
4. Mongo DB – Help command & Documentation

- The ‘Help’ command is a very handy command and can be used to check various commands available with Mongo DB
- To learn more on MongoDB Commands , visit: https://docs.mongodb.com/manual/reference/mongo-shell/



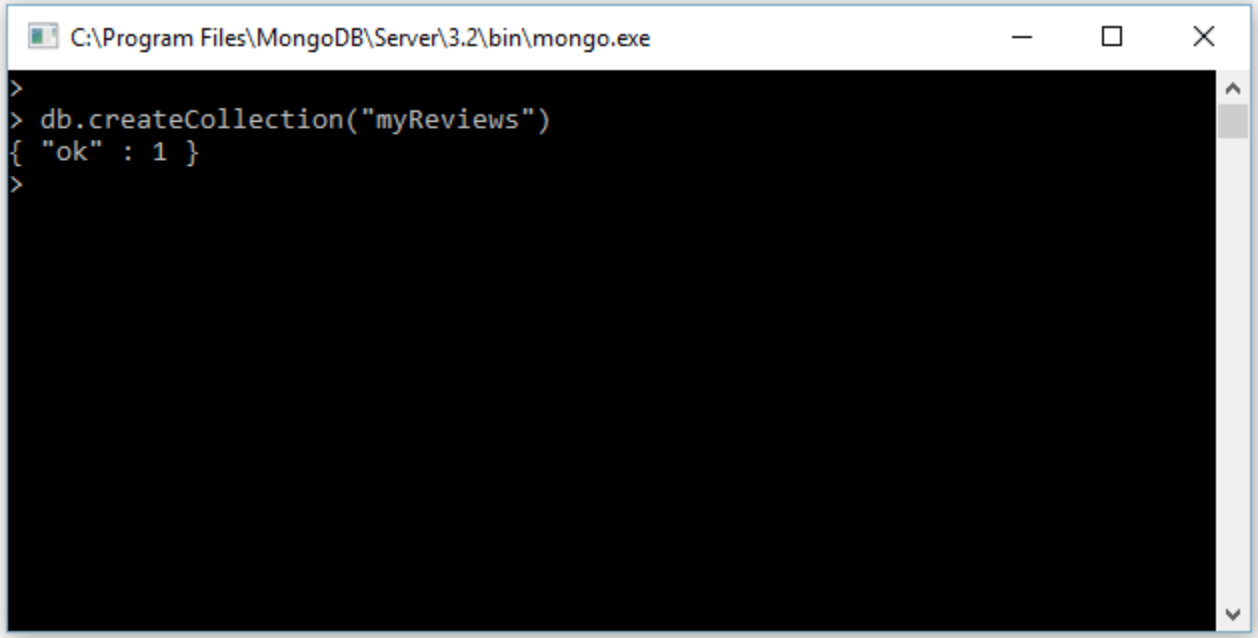
4. Mongo DB – Use a database

- In order to use a database, you must select it first
- To select a database along with the startup, use the command ‘use databasename’
- Example, to select the ‘CustomerReviews’ database, the command is ‘use CustomerReviews’
- You can then check the db you are in by typing db command



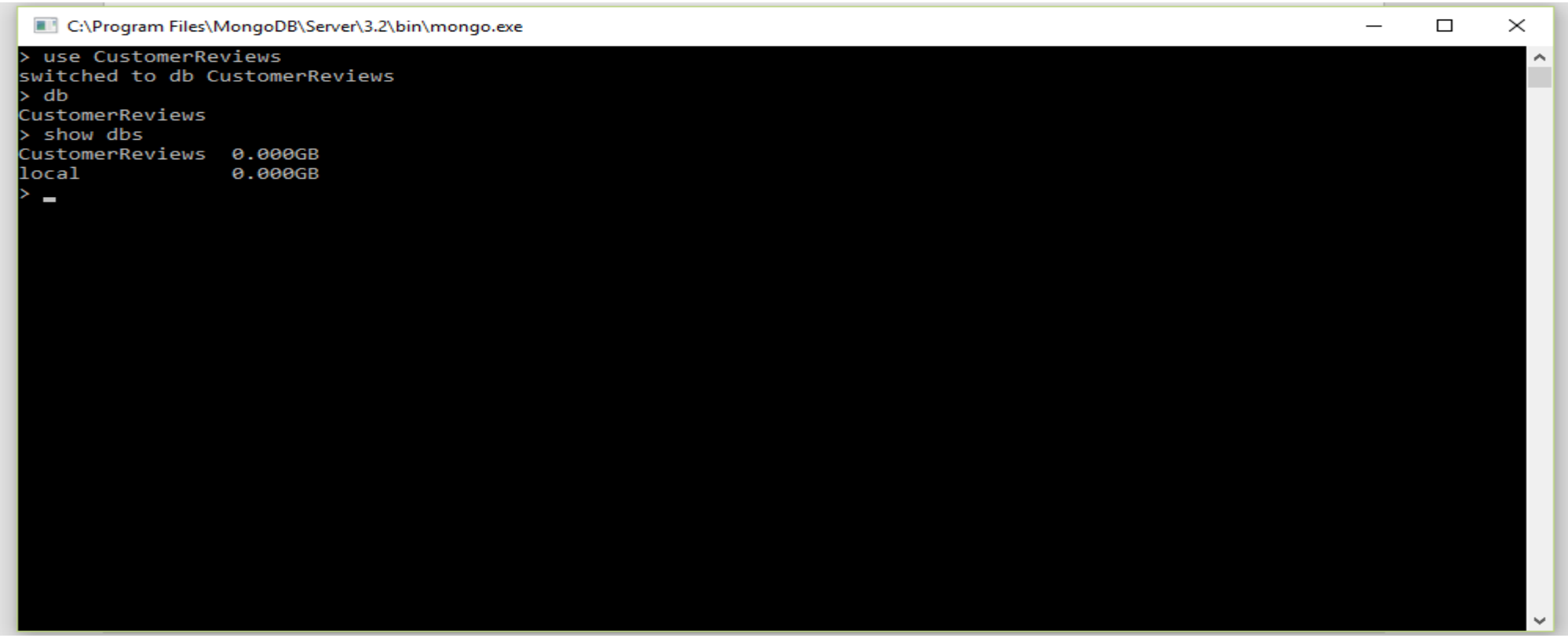
4. Mongo DB – Create Collections

- You can manually create collection or automatic by running your java program
- To create a collection manually type db.createCollection(collectionname)



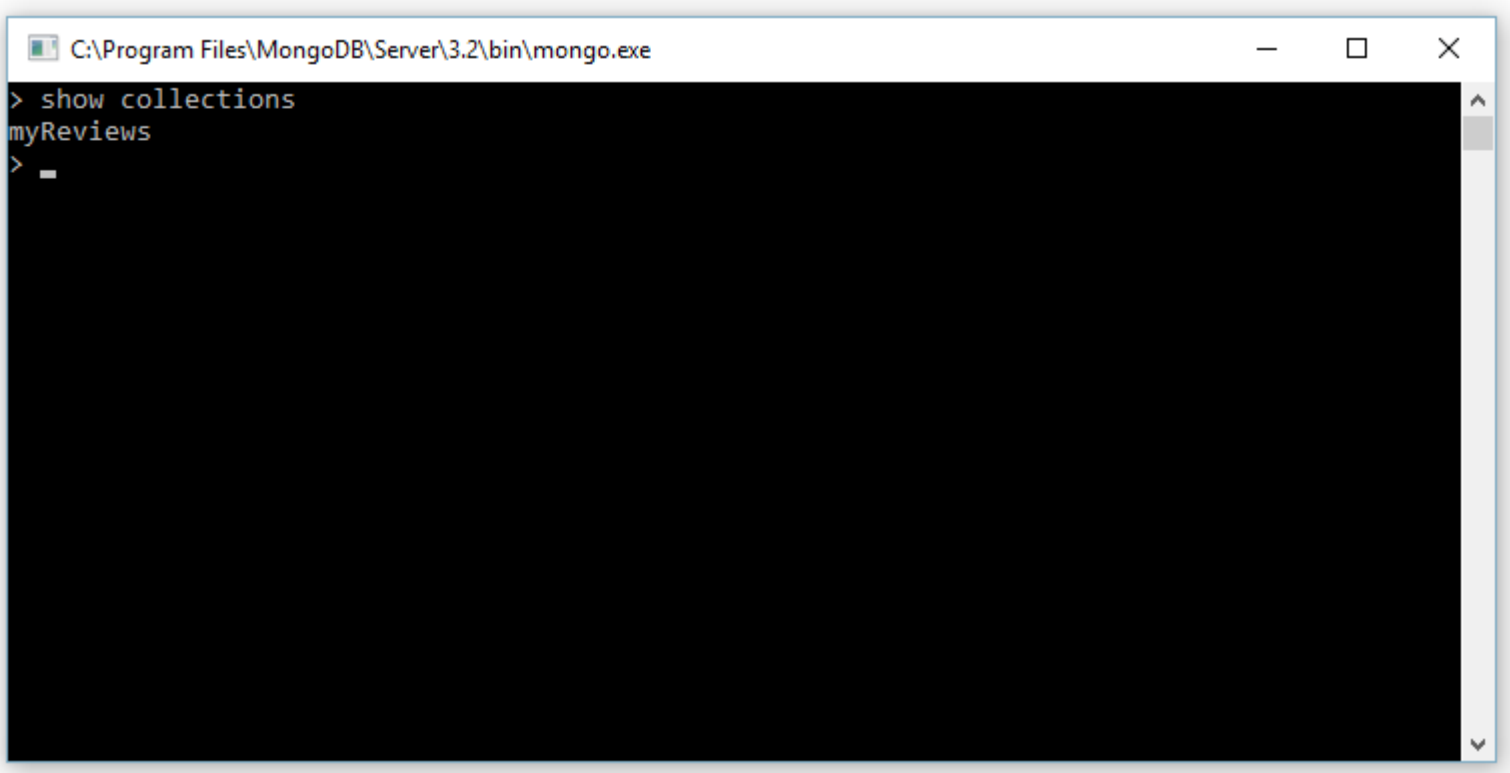
4. Mongo DB – Display list of available databases

- To check the databases that exist, use the command ‘show dbs’
- This will show the list of available databases



4. Mongo DB – Show collections

- Use the command ‘ show collections’ to view the list of available collections in the selected database



4. Mongo DB – Query data

- In order to query data, use the command ‘db.COLLECTION_NAME.find()’
- The find() queries the data available in the selected collection.
- Example, to query the ‘myReviews’ collection we use the command ‘db.myReviews.find()’

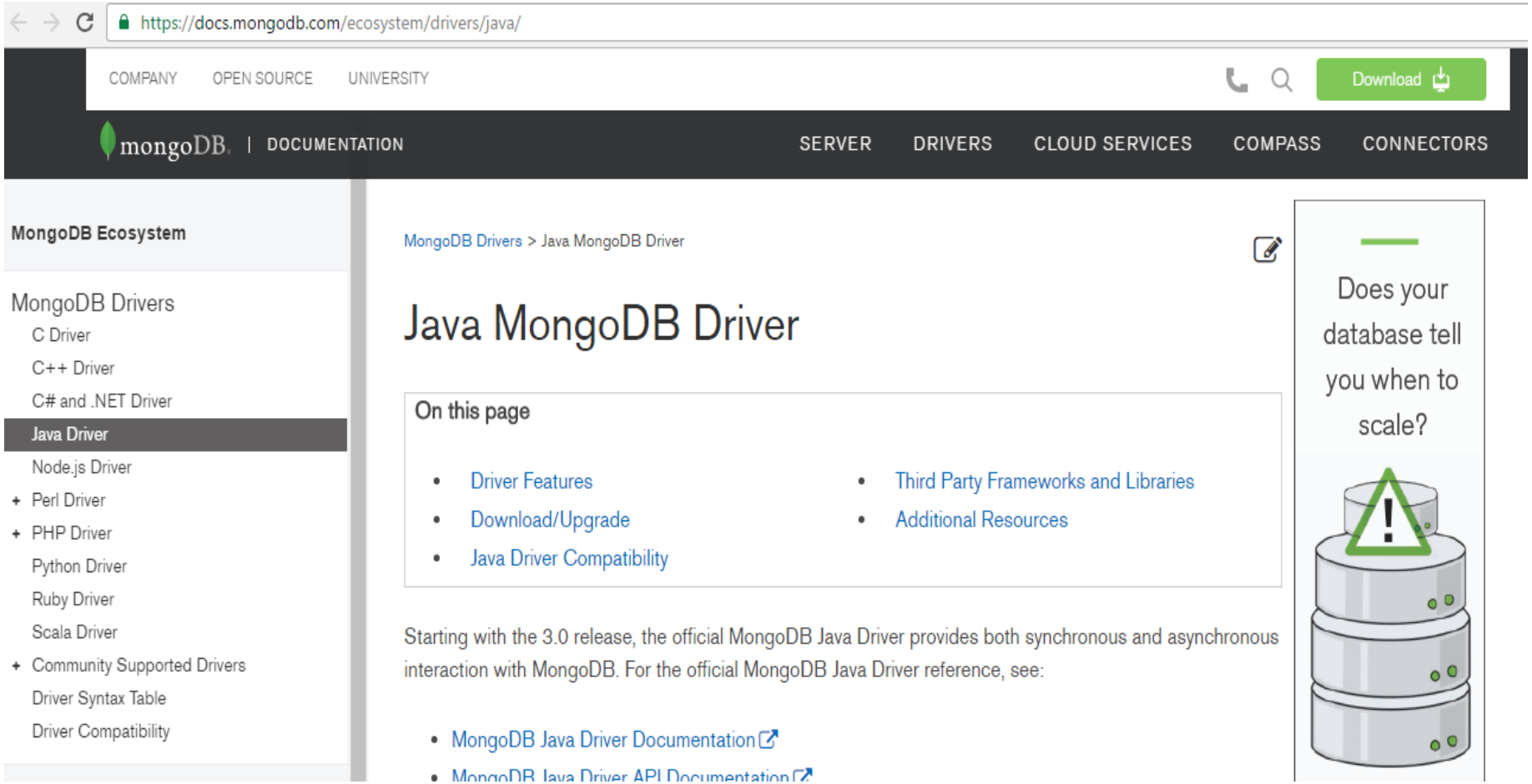
5. Compile and run

- You need to include all the JAR files before you compile your Java program which imports external libraries (Such as Servlets, MongoDB in this tutorial)
- To download MongoDB Connector jar go to http://mongodb.github.io/mongo-java-driver/?_ga=1.142913397.1760375742.1470875192
- To see the documentation for Java MongoDB Driver go to https://docs.mongodb.com/drivers/java
- To include these external JAR files, make the changes to the 'CLASSPATH' variable in your 'env-setup-for-tomcat_backup.bat' file
- Locate and copy the location of the JAR files on your computer and edit the 'CLASSPATH' variable accordingly

- **NOTE:** Make sure you have the necessary JAR files on your computer

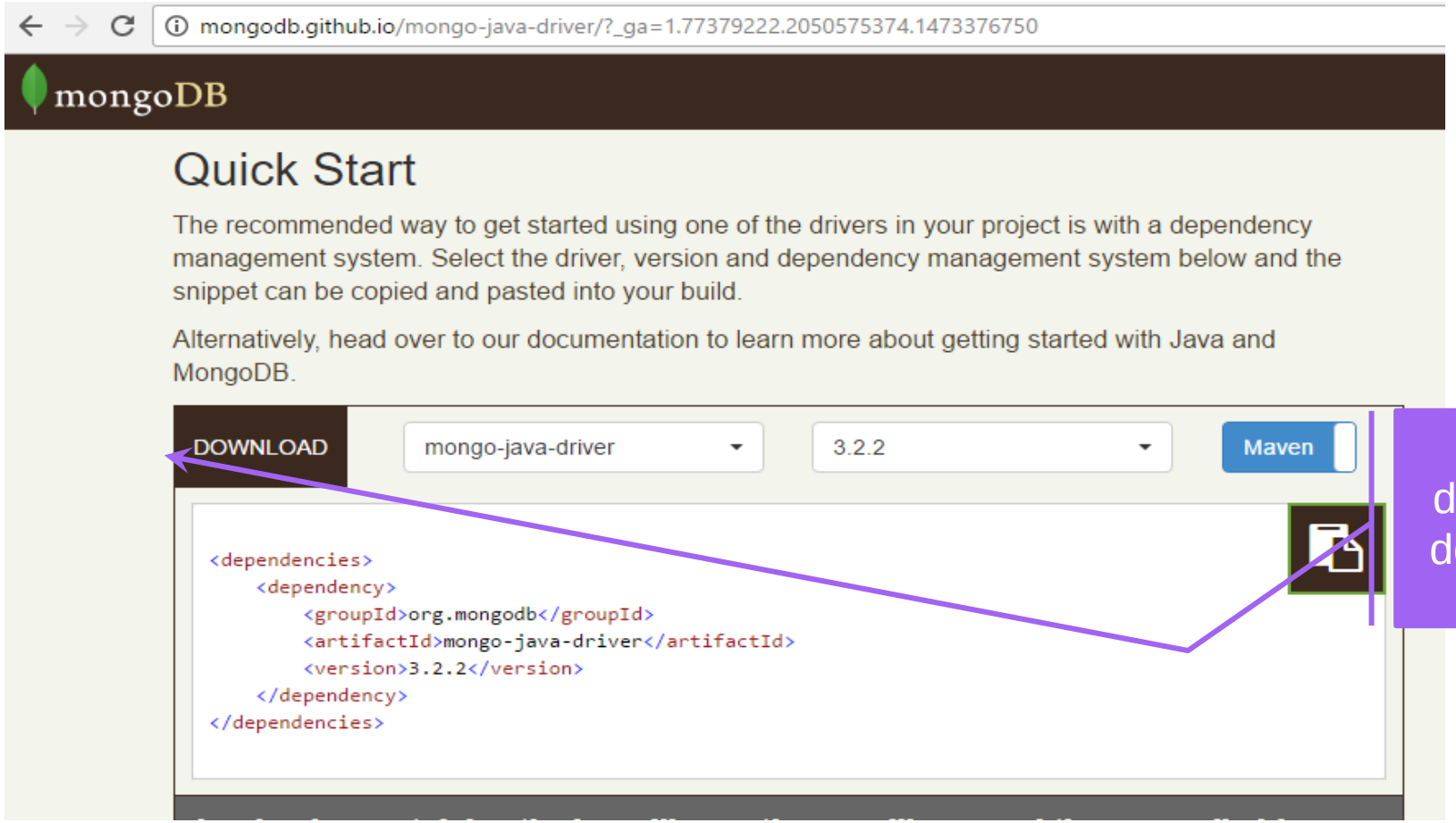
5. Compile and run

- Go to https://docs.mongodb.com/drivers/java to see documentation for Java MongoDB Driver .



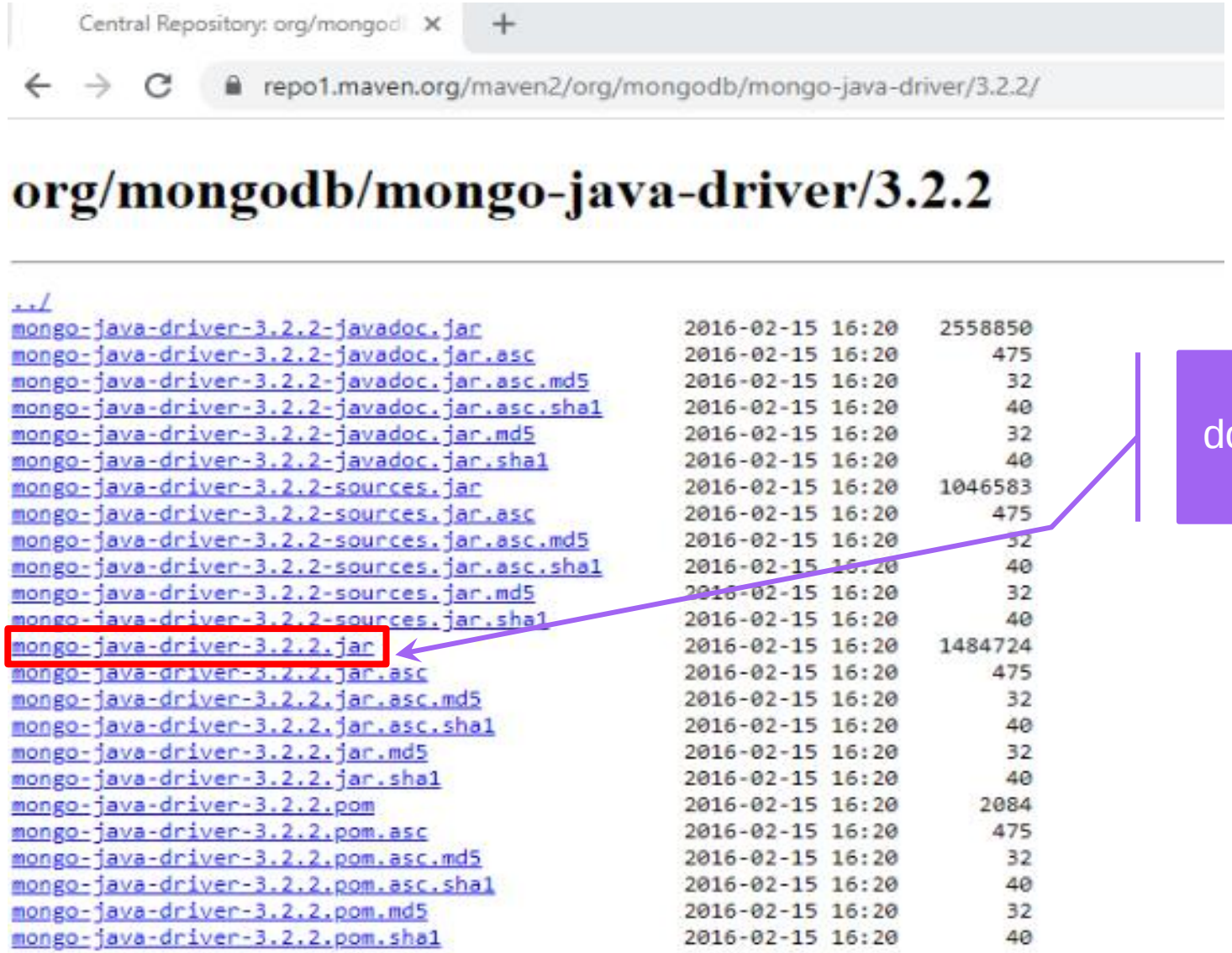
5. Compile and run

- Go to http://mongodb.github.io/mongo-java-driver/?_ga=1.142913397.1760375742.1470875192 to Download jar File.
- Select mongo-java-driver and version 3.2.2 and click Download Button. (Direct link: https://repo1.maven.org/maven2/org/mongodb/mongo-java-driver/3.2.2/)



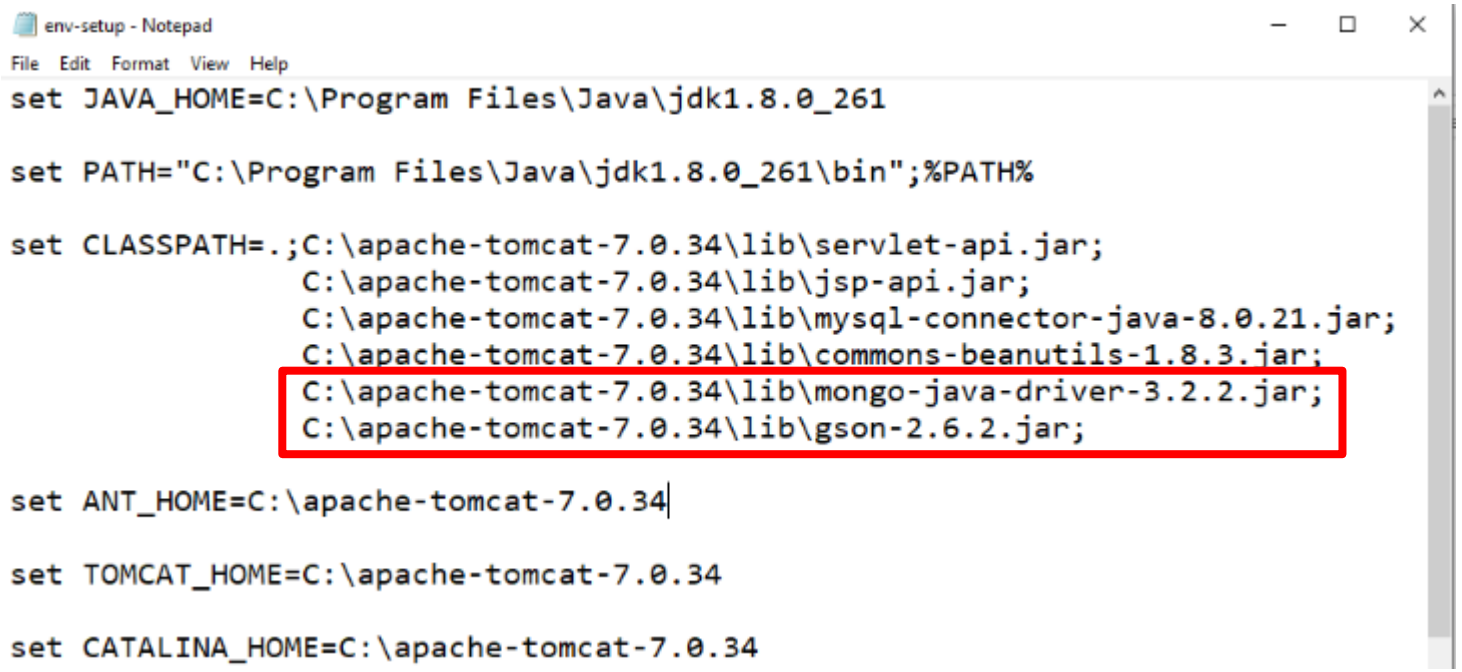
5. Compile and run

- To download click on mongo-java-driver-3.2.2.jar



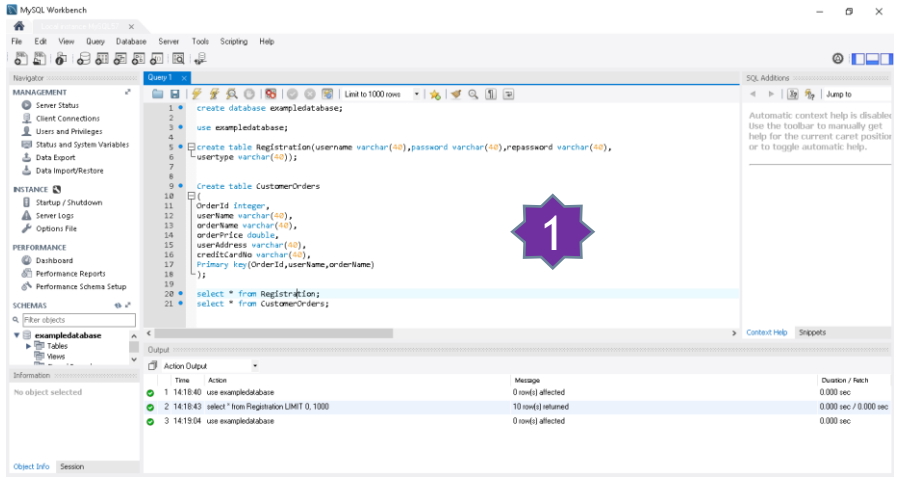
5. Compile and run

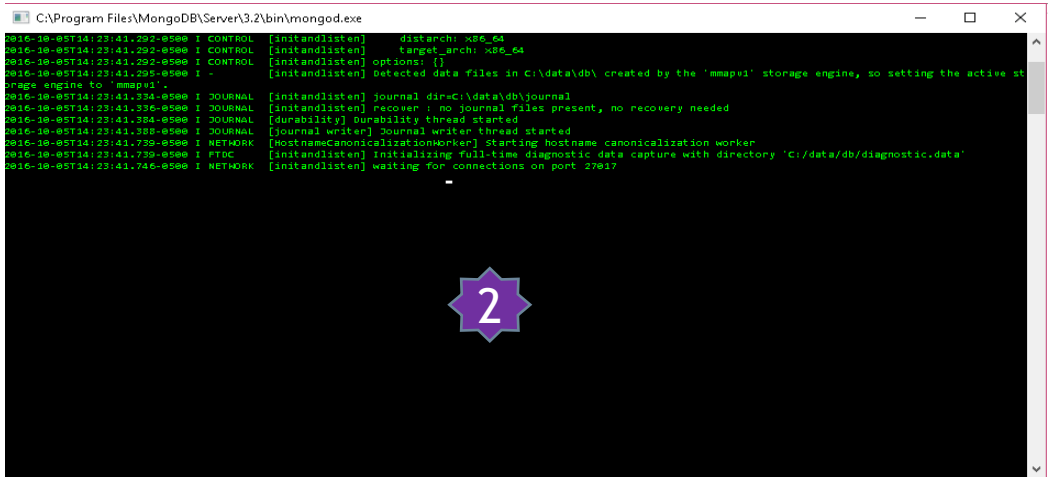
- Here is the snapshot of my ‘env-setup-for-tomcat_backup.bat’
- The location of the JAR files highlighted will differ based on where they are present on your computer. Please make sure you do the changes accordingly
- For Tutorial_3 to work, you also need to install **gson-2.6.2.jar** into the TOMCAT_HOME\lib folder and give the path into env-setup-for-tomcat_backup.bat or class path in system similarly to this (https://repo1.maven.org/maven2/com/google/code/gson/gson/2.6.2/)

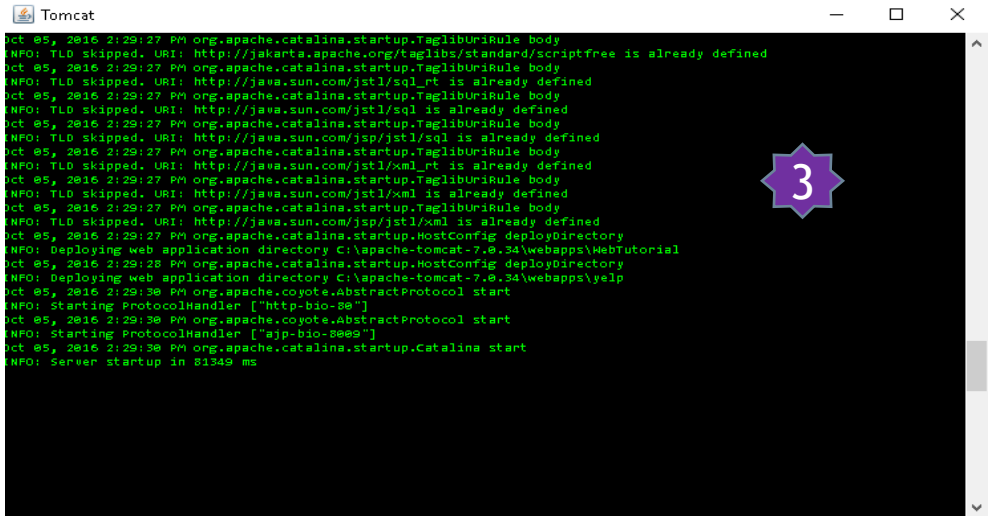


Things to Remember Before Running your Application in localhost:

- Check MySQL Server is up and Running or else start the MySQL Server .
- Check MongoDB Server is up and Running or else start the MongoDB Server .
- Check Apache Tomcat is up and Running or else start the Apache Tomcat .

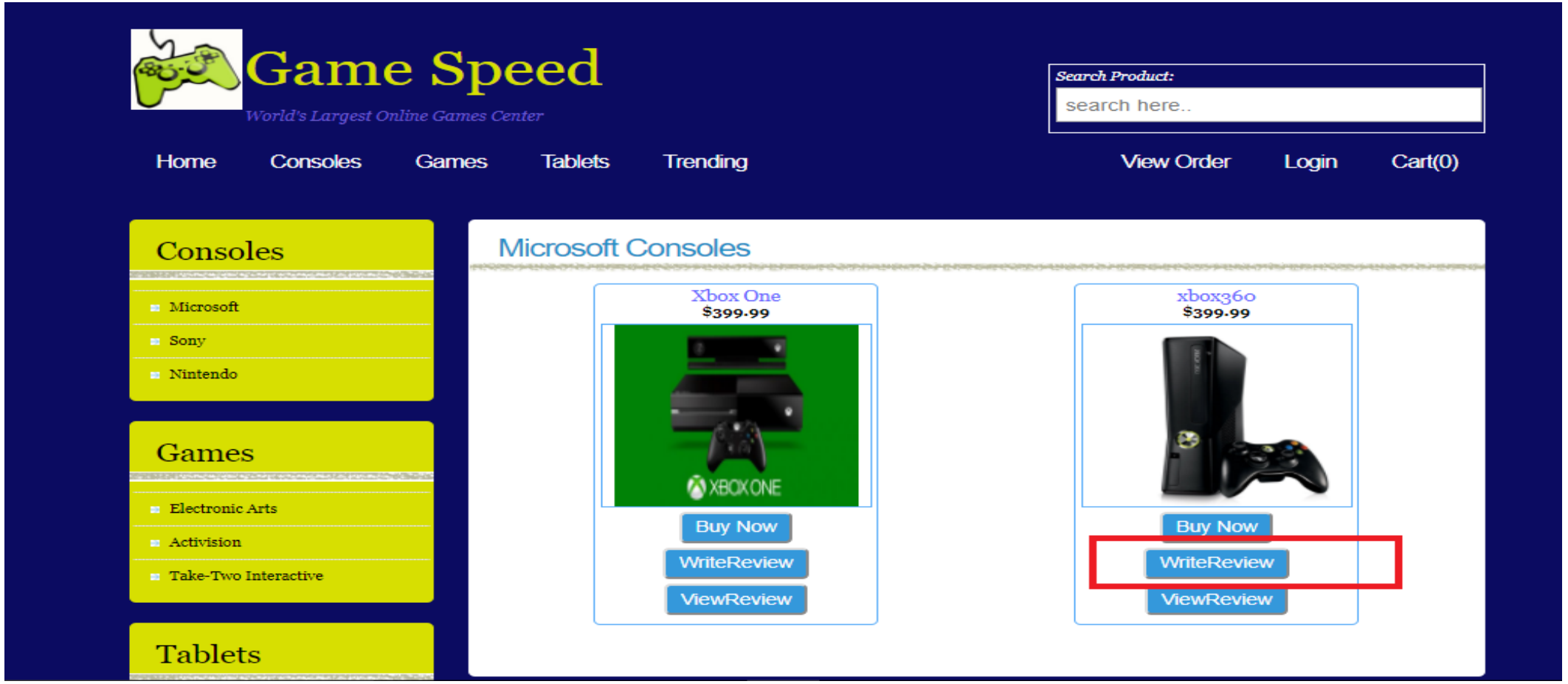






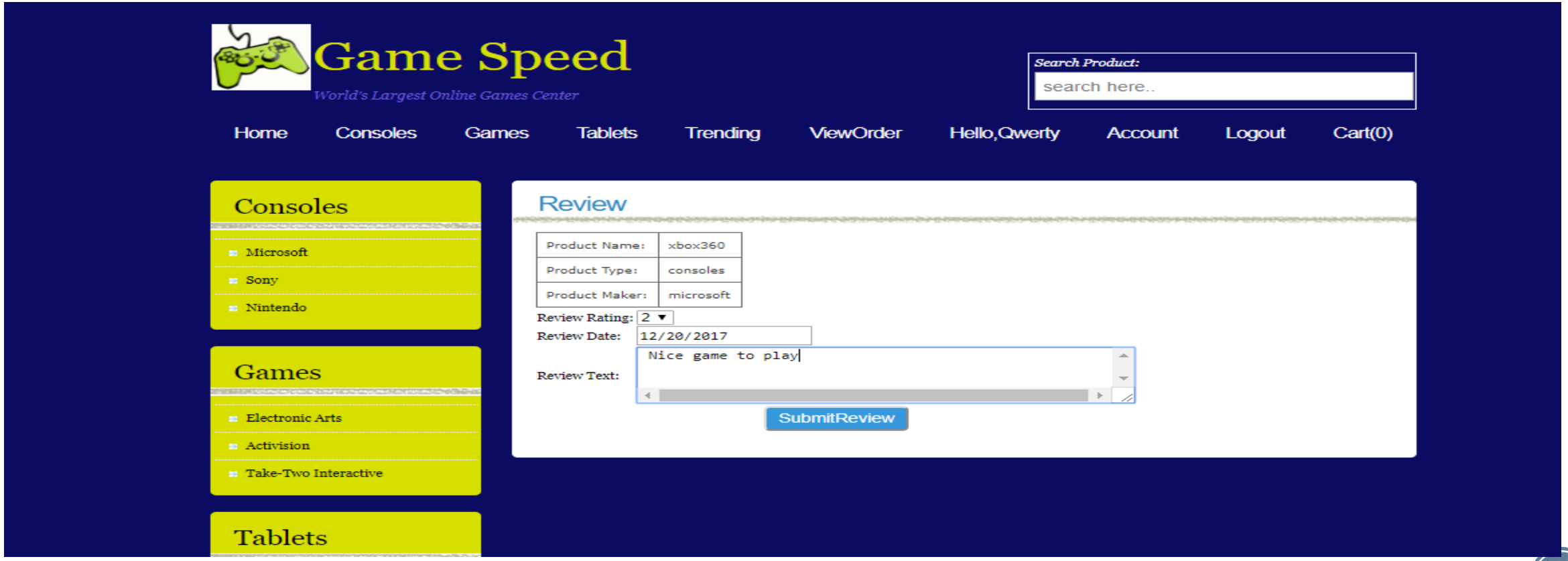
6. Example – Write Review:

To write a review for the product, click on ‘Write Review’ button on the products page



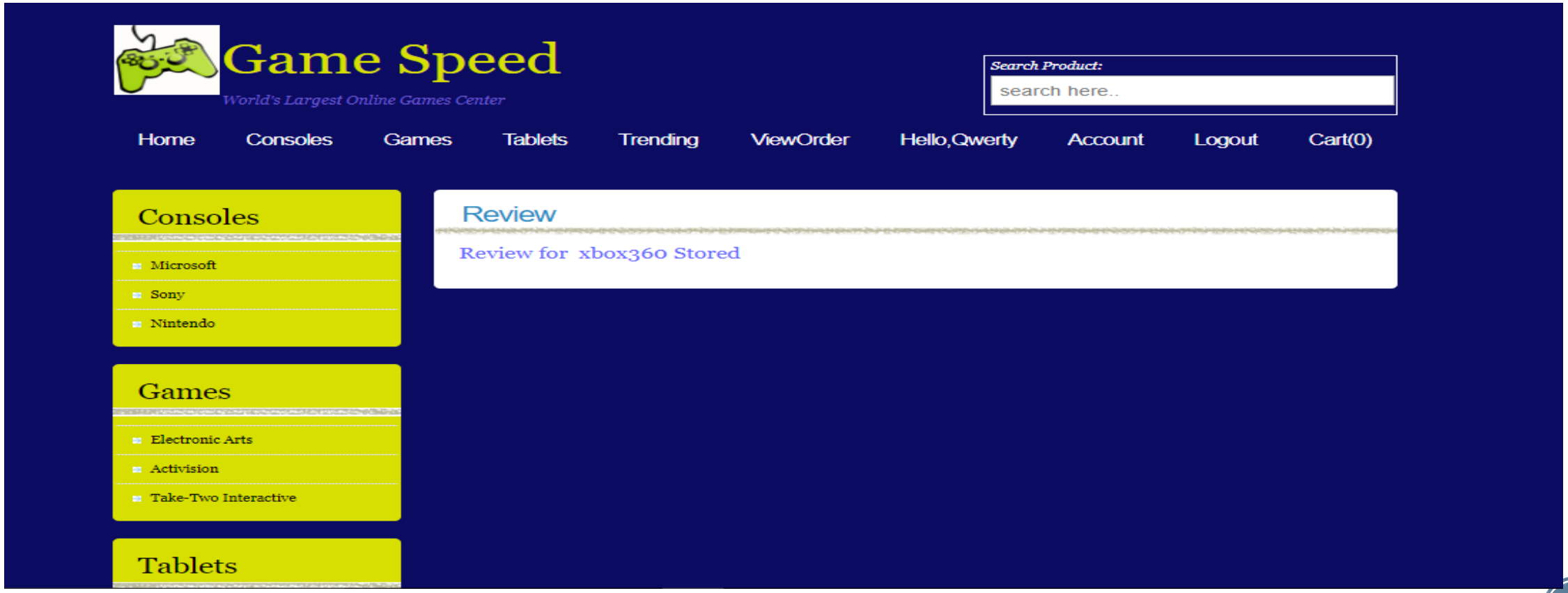
6. Example – Write Review:

On clicking the WriteReview Button from products page user will be directed to WriteReview webpage where he can give review for product.
Click the SubmitReview button to store the review in Mongo database



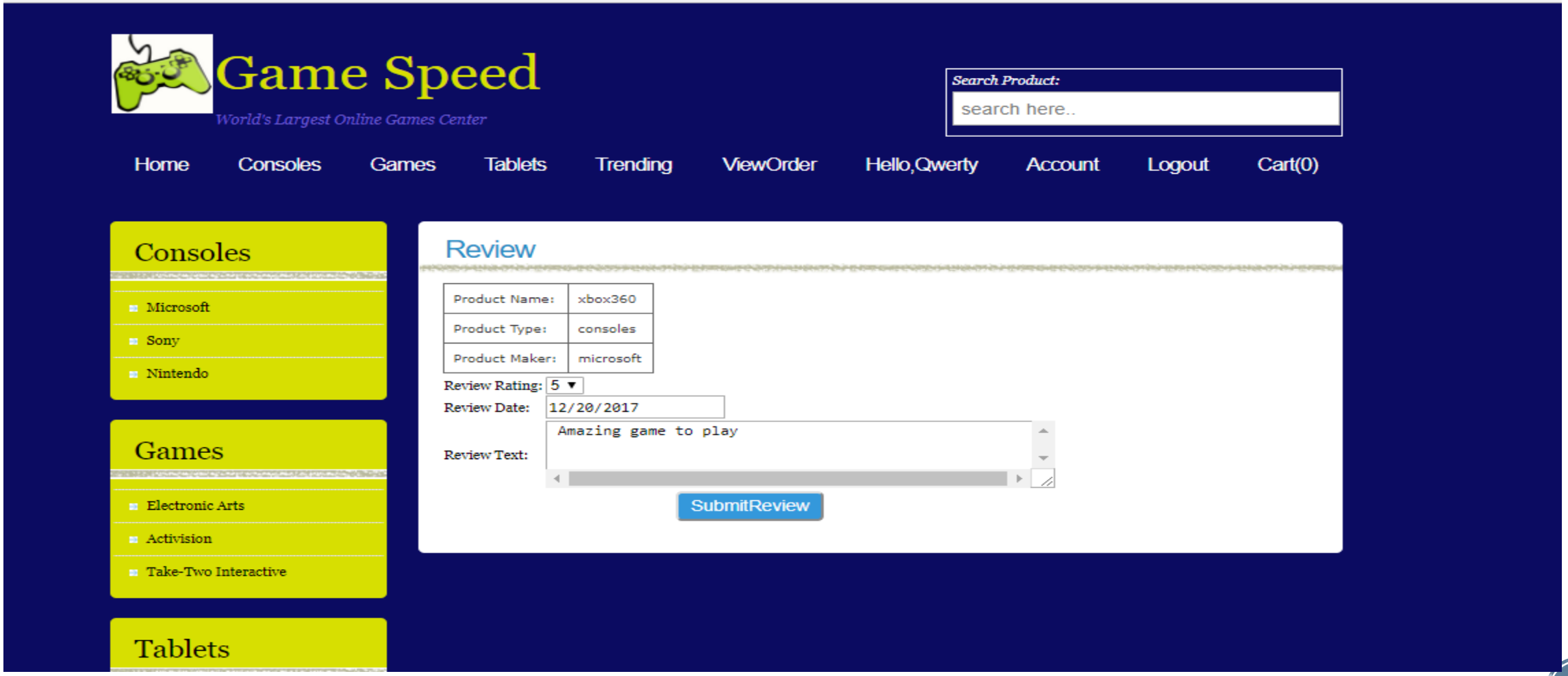
6. Example – Write Review:

On clicking the SubmitReview button user will get response that reviews for product is stored in database



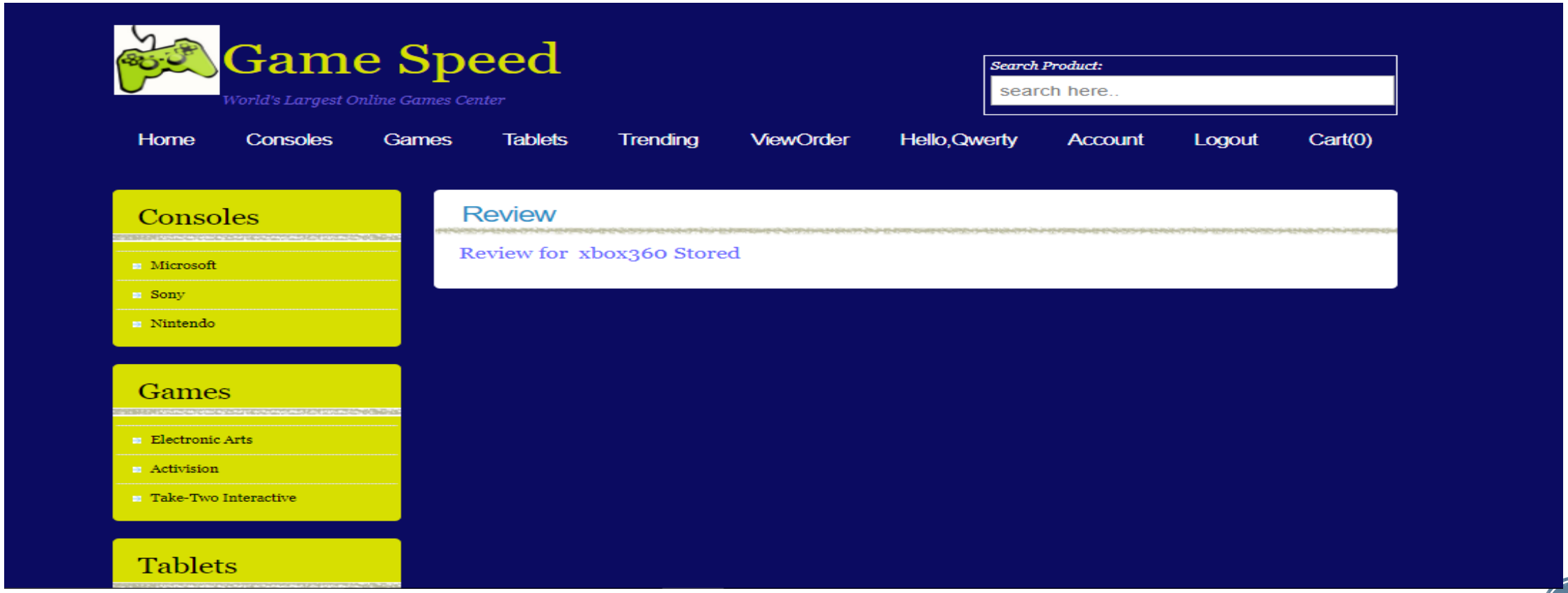
6. Example – Write Review:

Submitting one more Review for product



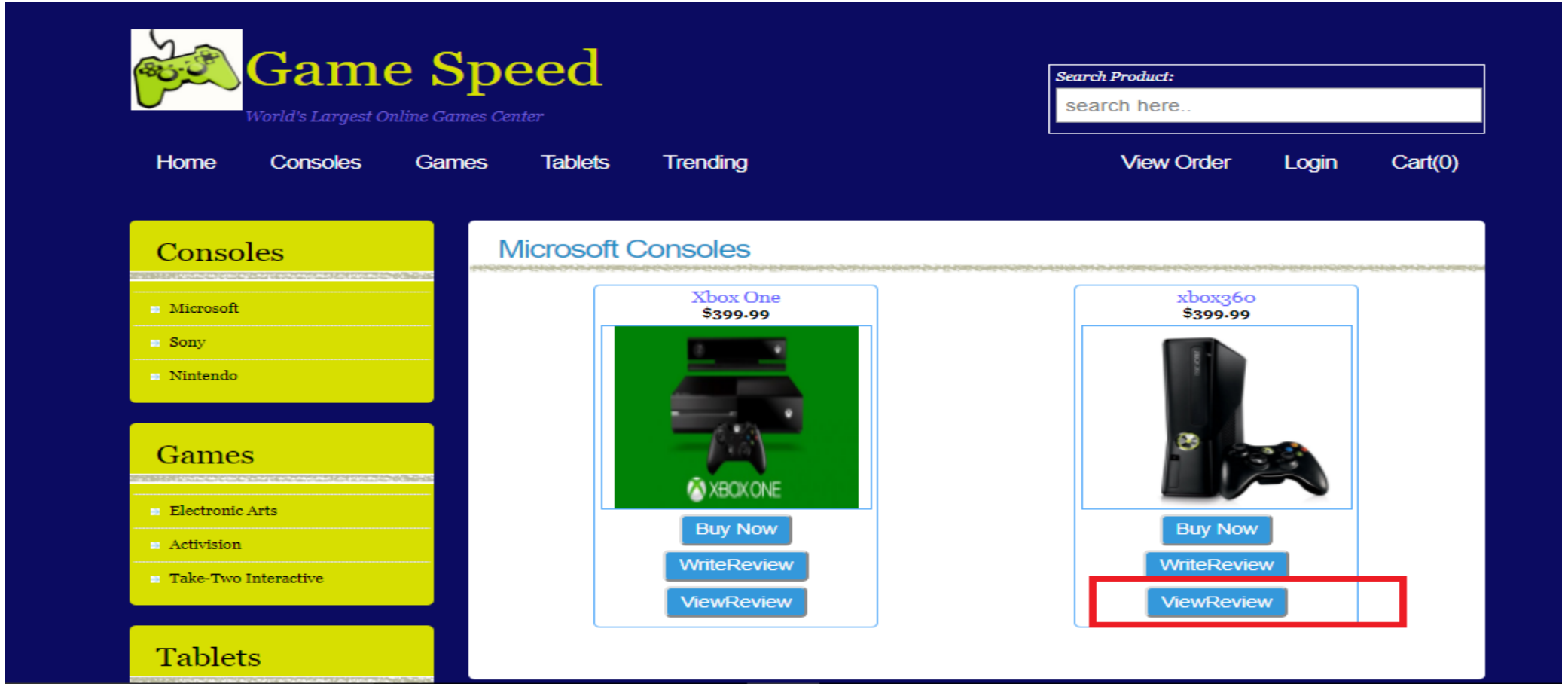
6. Example – Write Review:

On clicking the SubmitReview button user will get response that reviews for product is stored in database



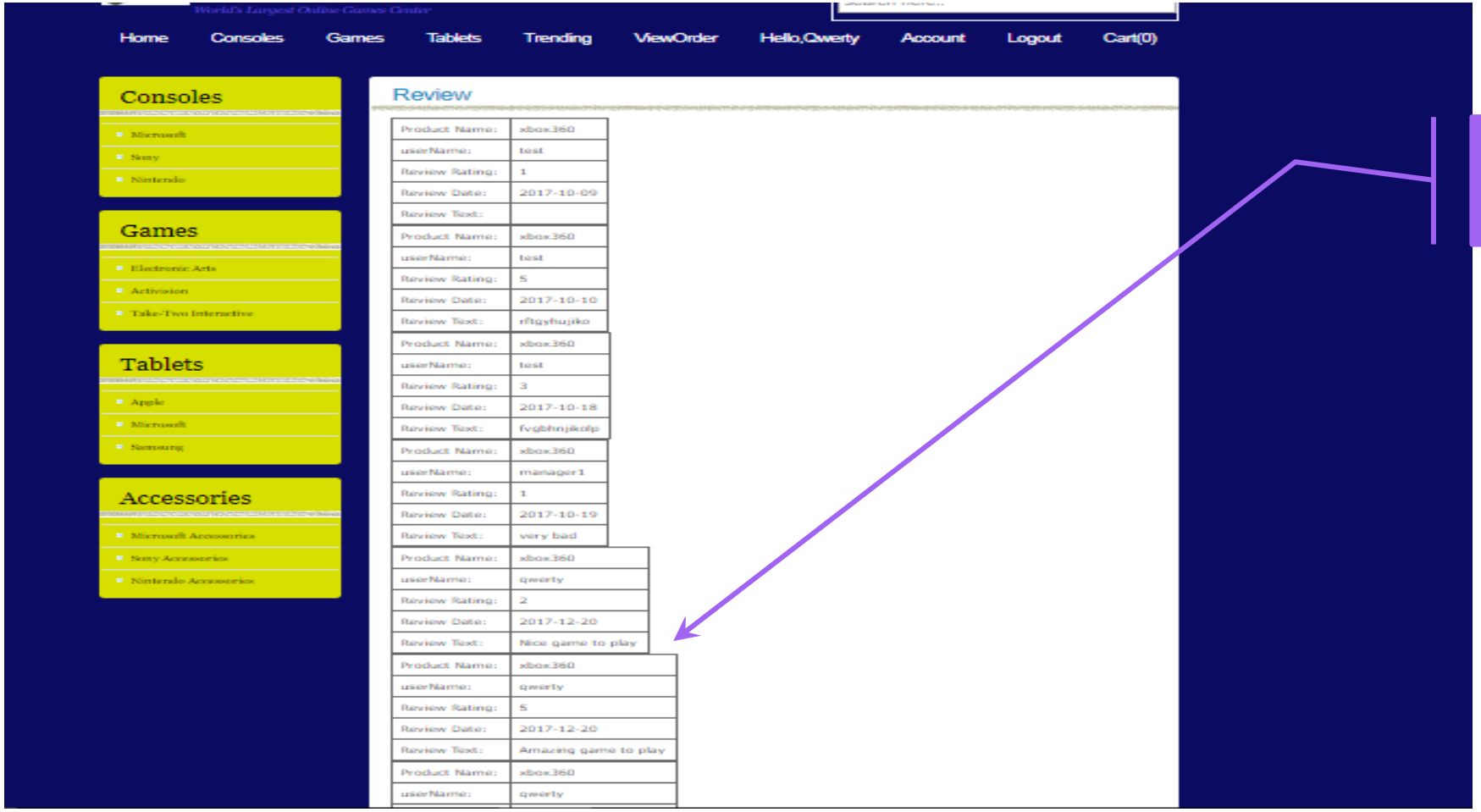
6. Example – View Review:

You can view the review submitted by clicking on ViewReview button on products page



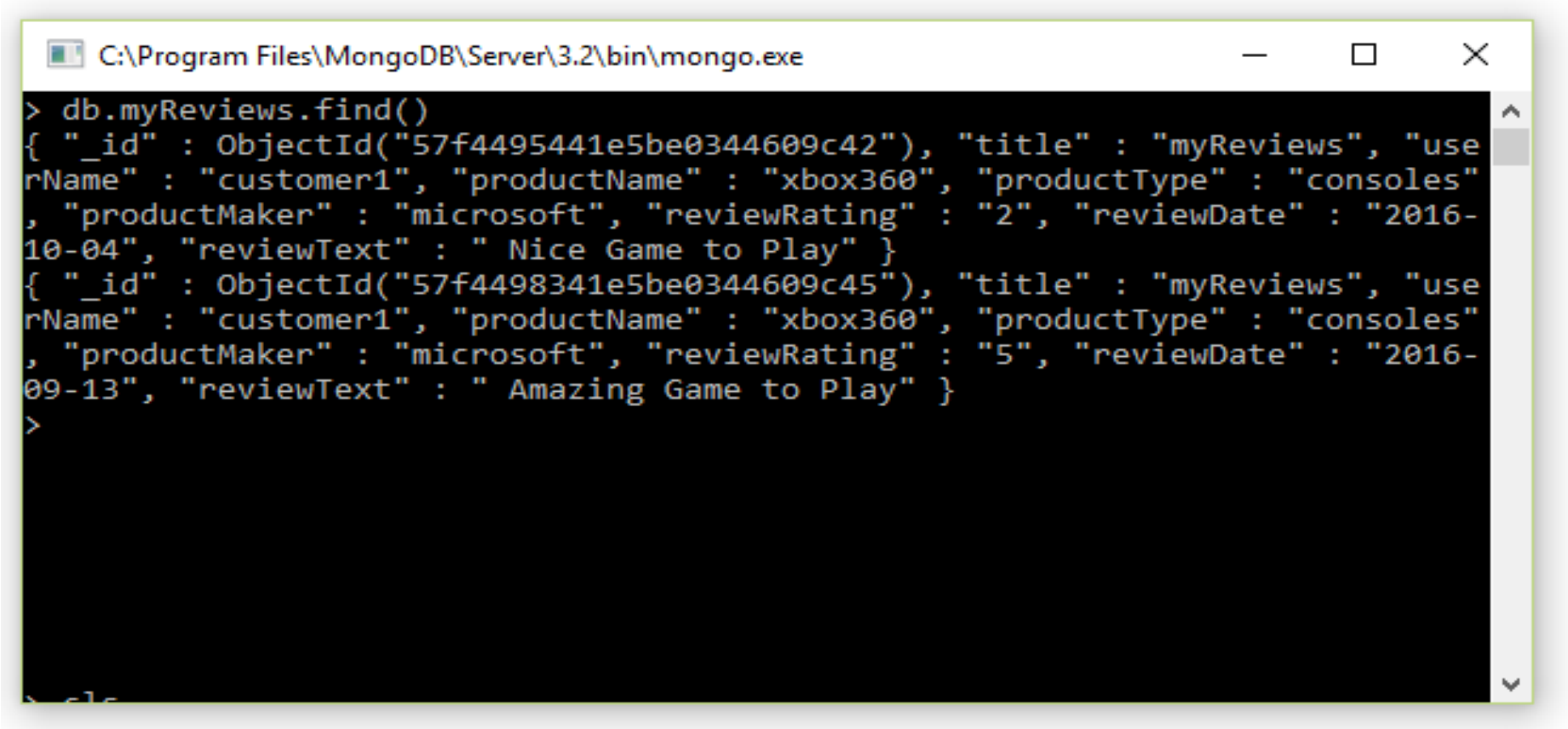
6. Example – View Review:

All the reviews for the product will be retrieved from mongo db and displayed in web page



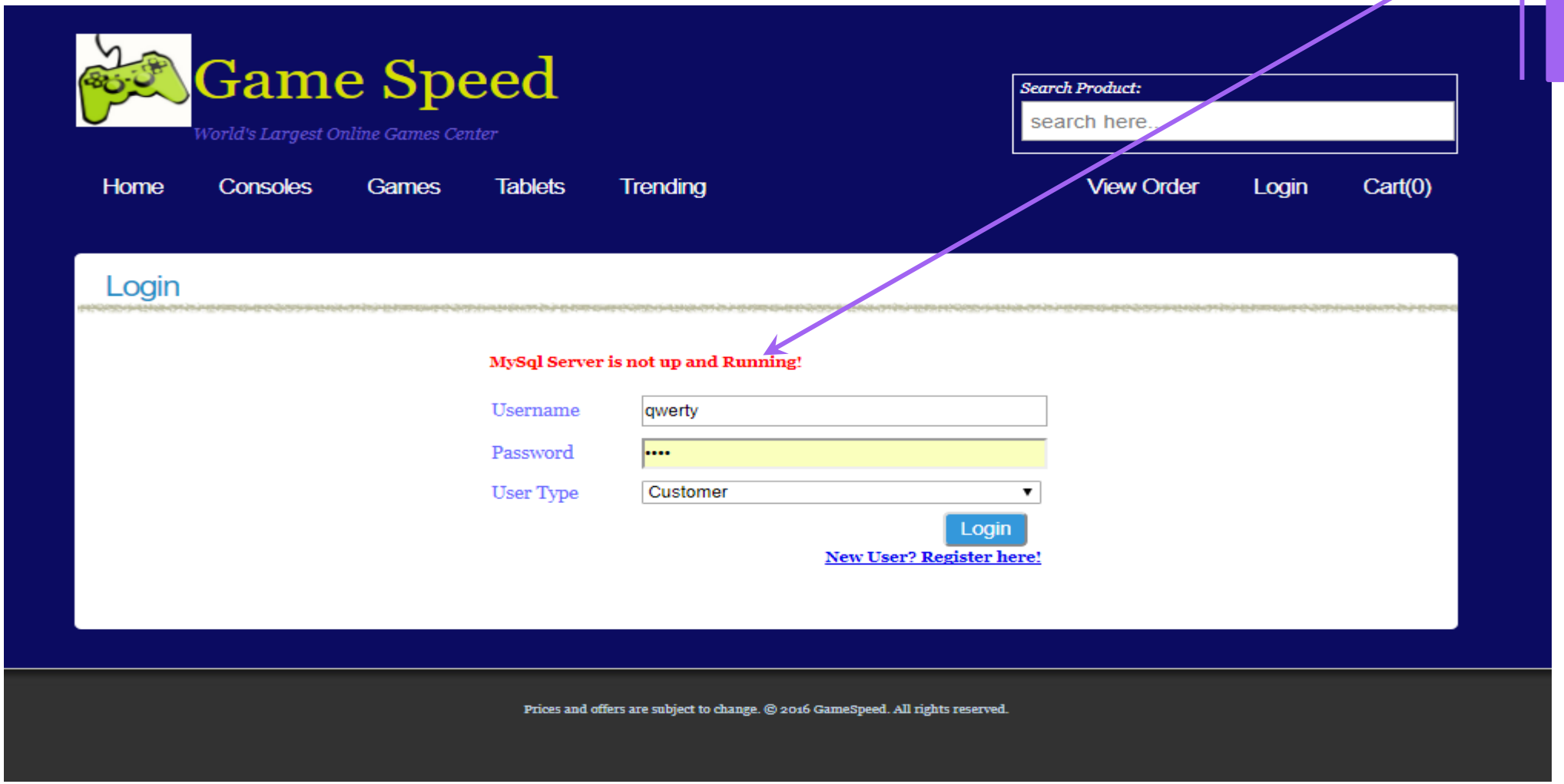
6. Example – View Review:

Check in the mongo shell if the myReviews collection is created inside example database and data for two reviews is stored in it



6. Example - Server Not Running For Registration:

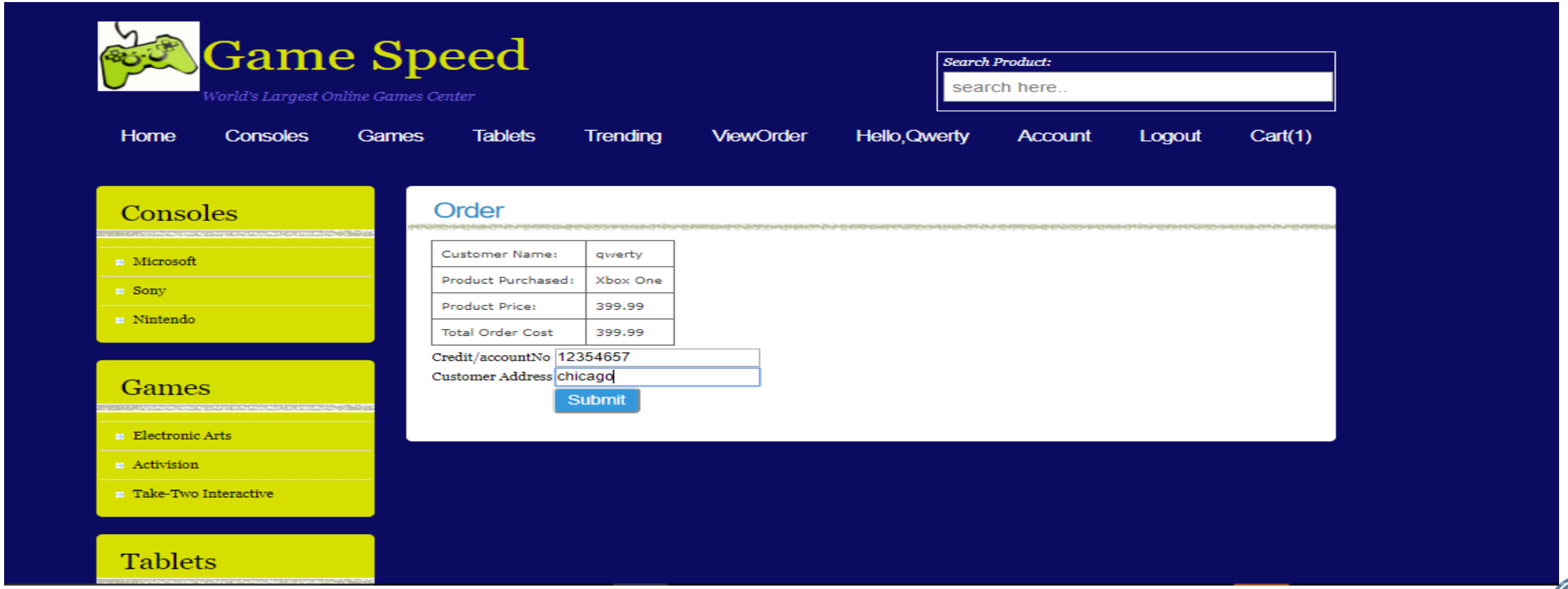
Trying to Register when server is not up and running



If mySql server not running gives an error message

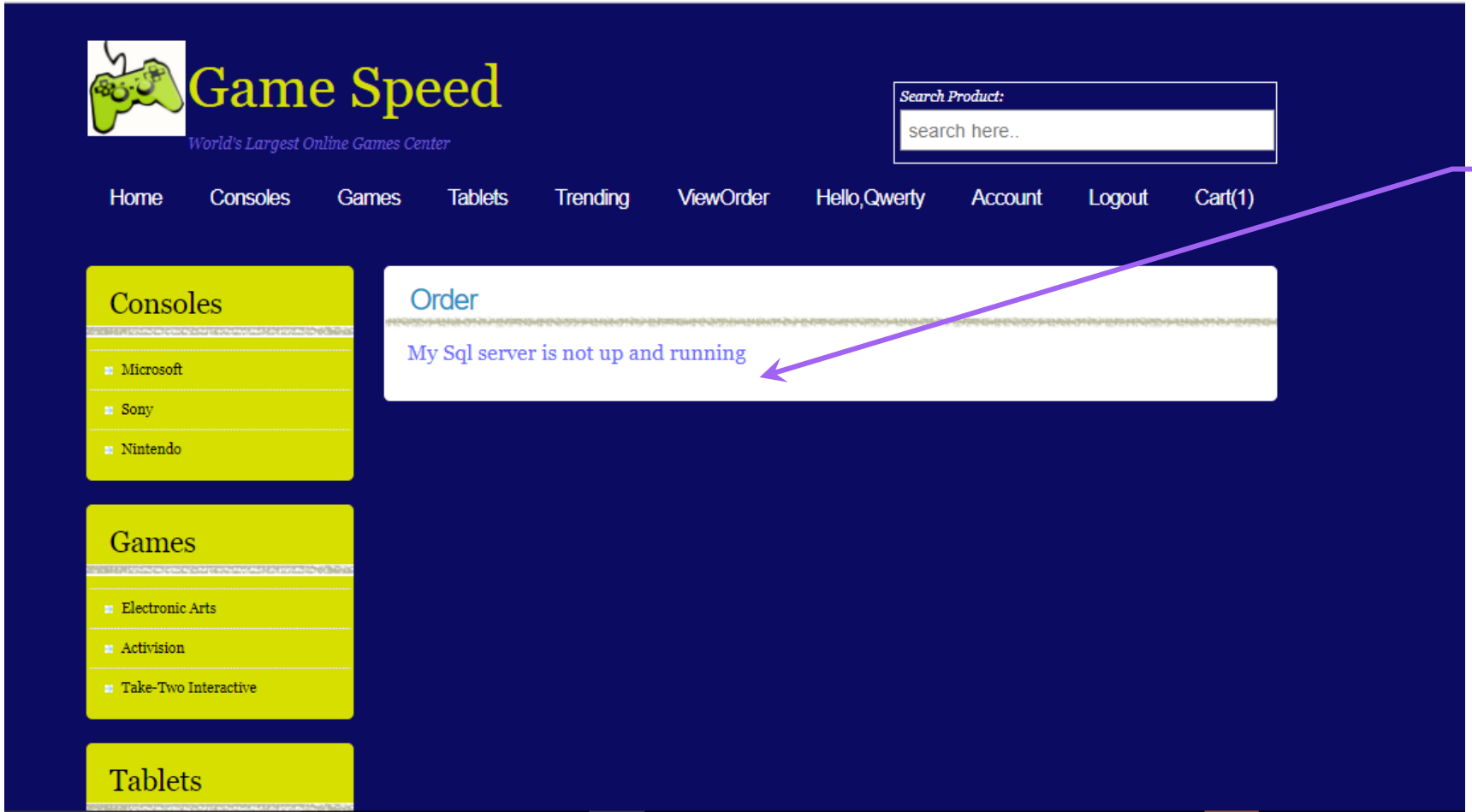
6. Example - Server Not Running For Orders:

Trying to Place order when server is not up and running



6. Example - Server Not Running For Orders:

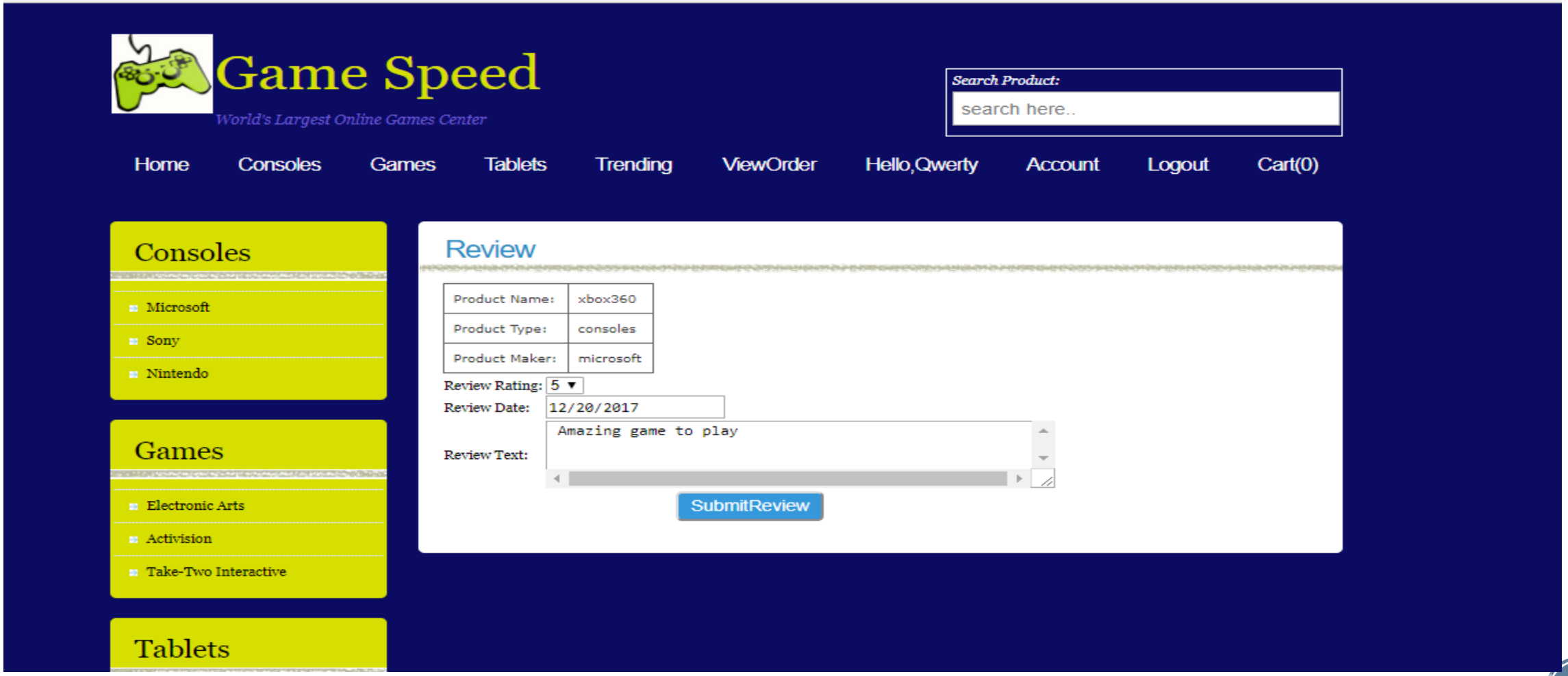
Trying to Place order when server is not up and running



If MySql server not running gives an error message

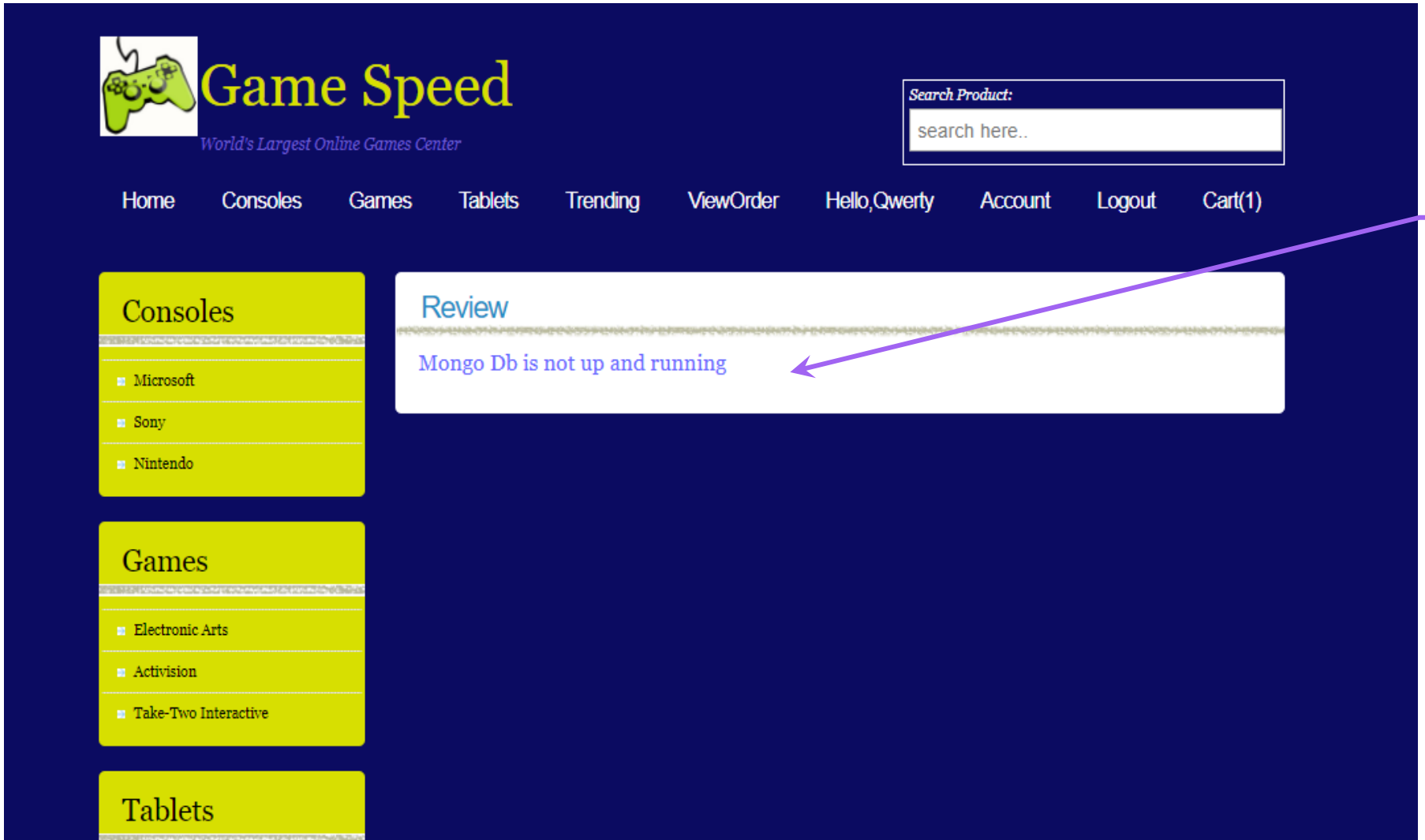
6. Example – Write Review when MongoDb Server not running:

Trying to submit review for Product



6. Example – Write Review when MongoDb Server not running:

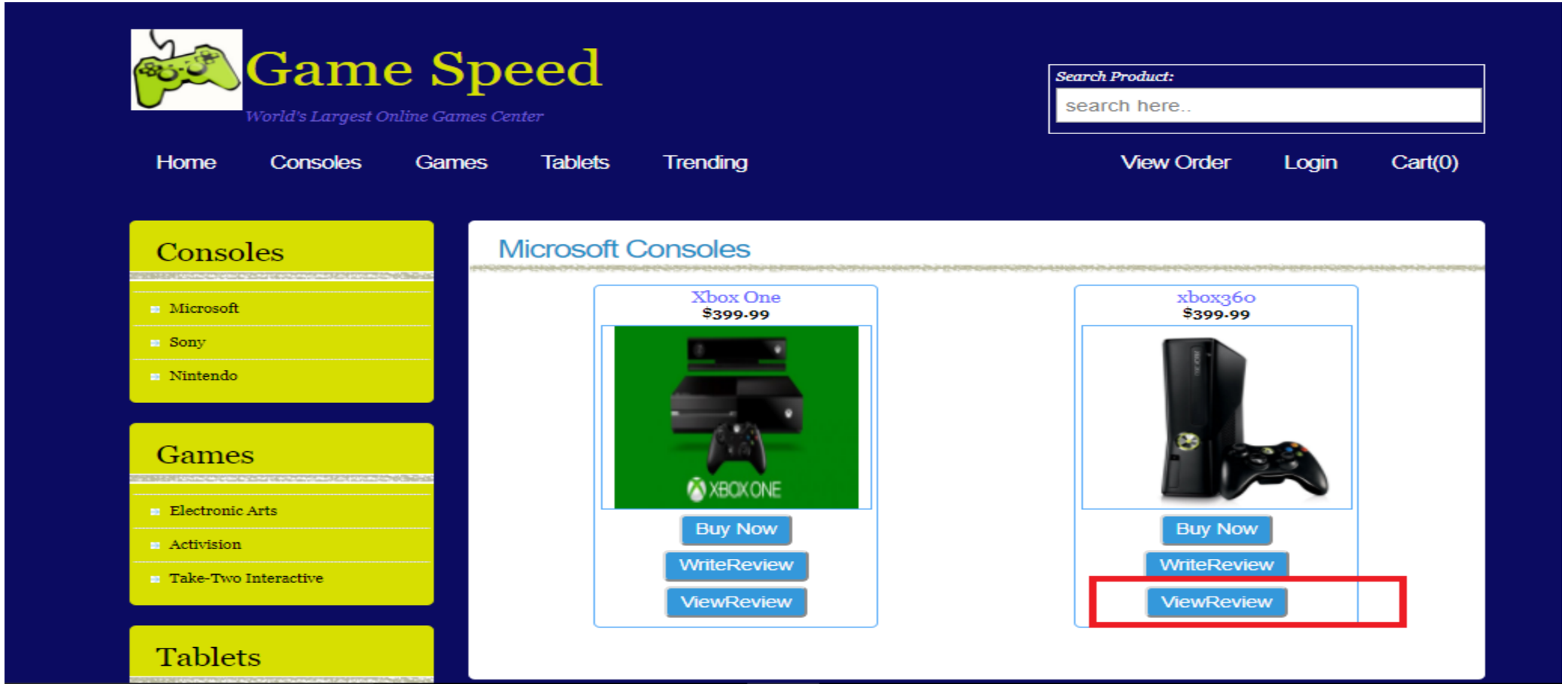
Trying to submit review for Product



If MongoDb server not running gives an error message

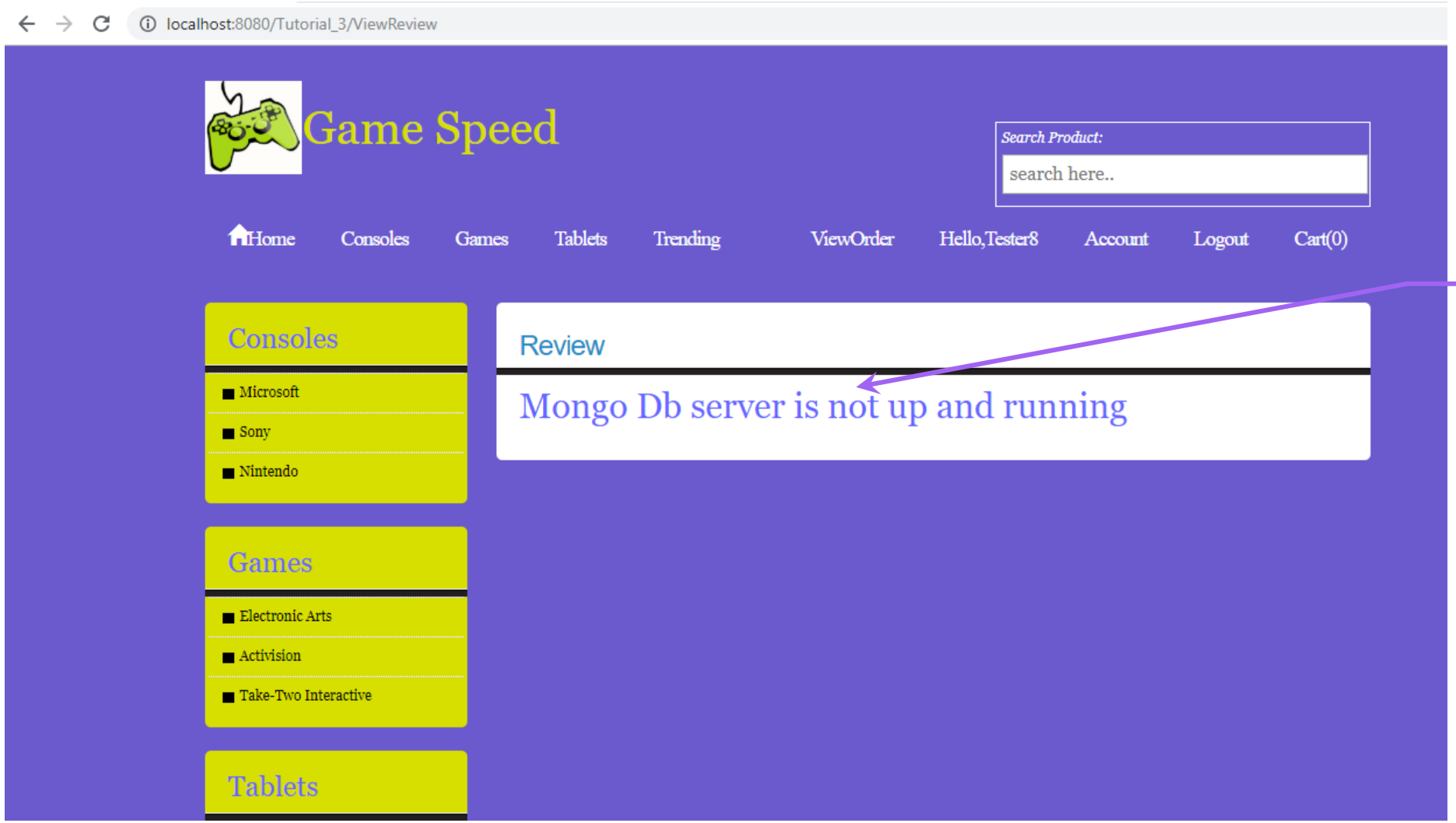
6. Example – View Review when MongoDb Server not running:

Trying to view review for Product



6. Example – View Review when MongoDb Server not running:

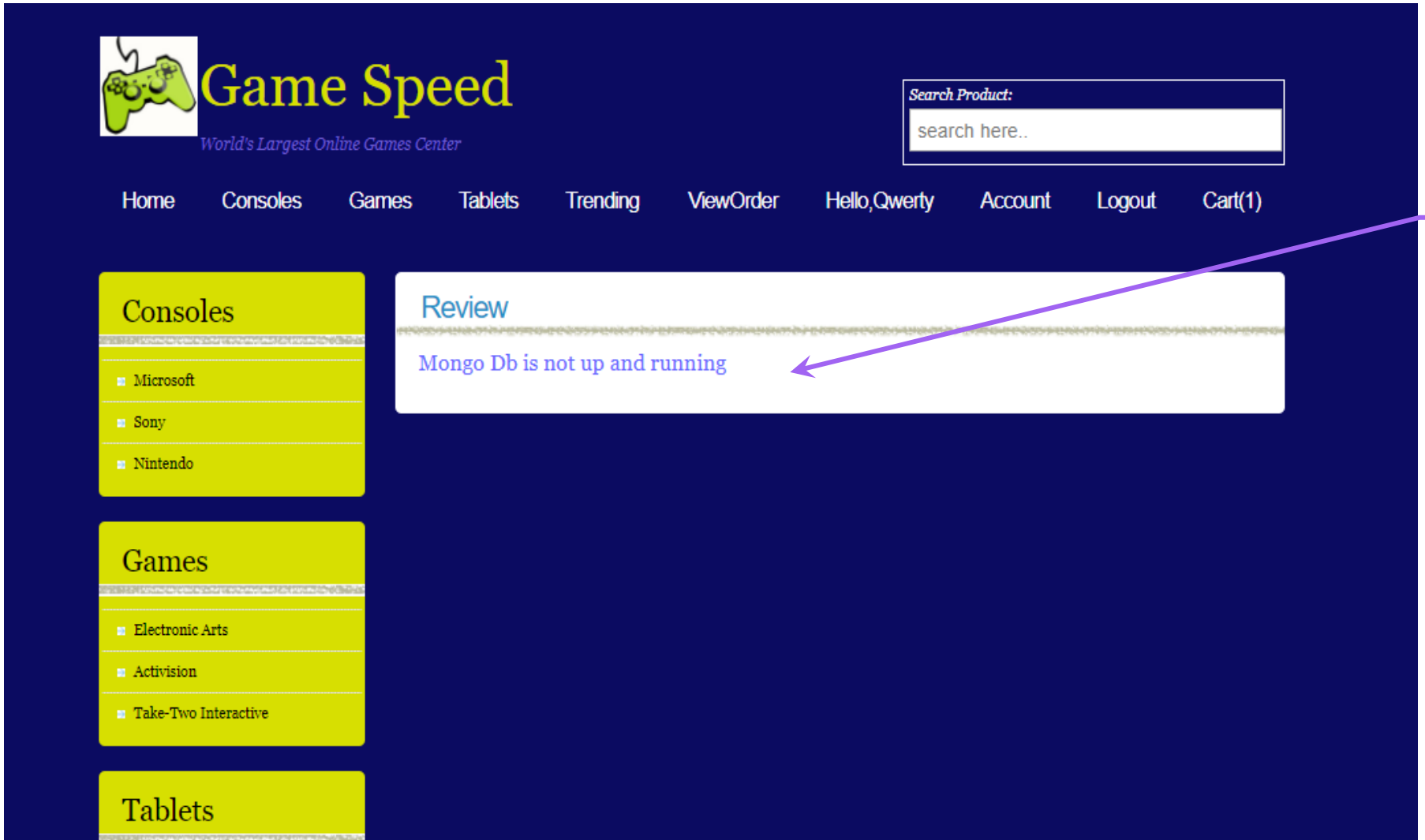
Trying to view review for Product



If MongoDb server not running gives an error message

6. Example – Write Review when MongoDb Server not running:

Trying to submit review for Product



If MongoDb server not running gives an error message

7. Code Snippet

Walkthrough to get connect to Database from Servlet

MongoDBDataStoreUtilities class to connect Database from Servlet

```
public class MongoDBDataStoreUtilities
{
static DBCollection myReviews;
public static void getConnection()
{
MongoClient mongo;
mongo = new MongoClient("localhost", 27017);

DB db = mongo.getDB(“CustomerReviews");
 myReviews= db.getCollection("myReviews");
}
```

Connecting to CustomerReviews database

Getting Reviews data to DbCollection object

Walkthrough for Storing Reviews Code Snippet

Walkthrough for Storing reviews

```
public void storeReview(String productname,String producttype,String reviewrating,String reviewdate,String
reviewtext)
{           HashMap<String, ArrayList<Review>> reviews= new HashMap<String, ArrayList<Review>>();
            try
            {reviews=MongoDBDataStoreUtilities.selectReview();}
            catch(Exception e)
            { }
            if(!reviews.containsKey(productname)){
                  ArrayList<Review> arr = new ArrayList<Review>();
                  reviews.put(productname, arr);
            }
            ArrayList<Review> listReview = reviews.get(productname);
            Review review = new Review(productname,username(),producttype,reviewrating,reviewdate,reviewtext);
            listReview.add(review);
            try
            {
MongoDBDataStoreUtilities.insertReview(productname,username(),producttype,reviewrating,reviewdate,reviewtext
            }
            catch(Exception e)
            { }
}
```

Calling utility function to select data from database and storing reviews in hashmap

Calling utility function to inserting reviews in database

Utility Function for Selecting Review Data into Hashmap

```
public static HashMap<String, ArrayList<Review>> selectReview()
{
	getConnection();
	HashMap<String, ArrayList<Review>> reviewHashmap=new HashMap<String, ArrayList<Review>>();
	DBCursor cursor = myReviews.find();
	while (cursor.hasNext())
	{
	BasicDBObject obj = (BasicDBObject) cursor.next();
		if(! reviewHashmap.containsKey(obj.getString("productName")))
		  {
			ArrayList<Review> arr = new ArrayList<Review>();
			reviewHashmap.put(obj.getString("productName"), arr);
		  }
		  ArrayList<Review> listReview = reviewHashmap.get(obj.getString("productName"));
		  Review review =new
Review(obj.getString("productName"),obj.getString("userName"),obj.getString("productType"),obj.getStr
ing("reviewRating"),obj.getString("reviewDate"),obj.getString("reviewText"));
				listReview.add(review);
	}
	return  reviewHashmap;
}
```

DBCursor used to store table data obtained from database in servlet

Iterate through Cursor and Store each review into class object

Utility Function for Writing Reviews into Mongo database

```
public static void insertReview(String productname,String username,String
producttype,String reviewrating,String reviewdate,String reviewtext)
{
        getConnection();
        BasicDBObject doc = new BasicDBObject("title", "myReviews").
            append("userName", username).
            append("productName", productname).
            append("productType", producttype).
            append("reviewRating", reviewrating).
            append("reviewDate", reviewdate).
            append("reviewText", reviewtext);
        myReviews.insert(doc);



}
```

Creating a BasicObject to insert data into database

Specifying each column to insert value

DbCollection.insert() Will insert data into database

Questions?