CSCI 5410: Assignment 1

Part B

1. Flowcharts:

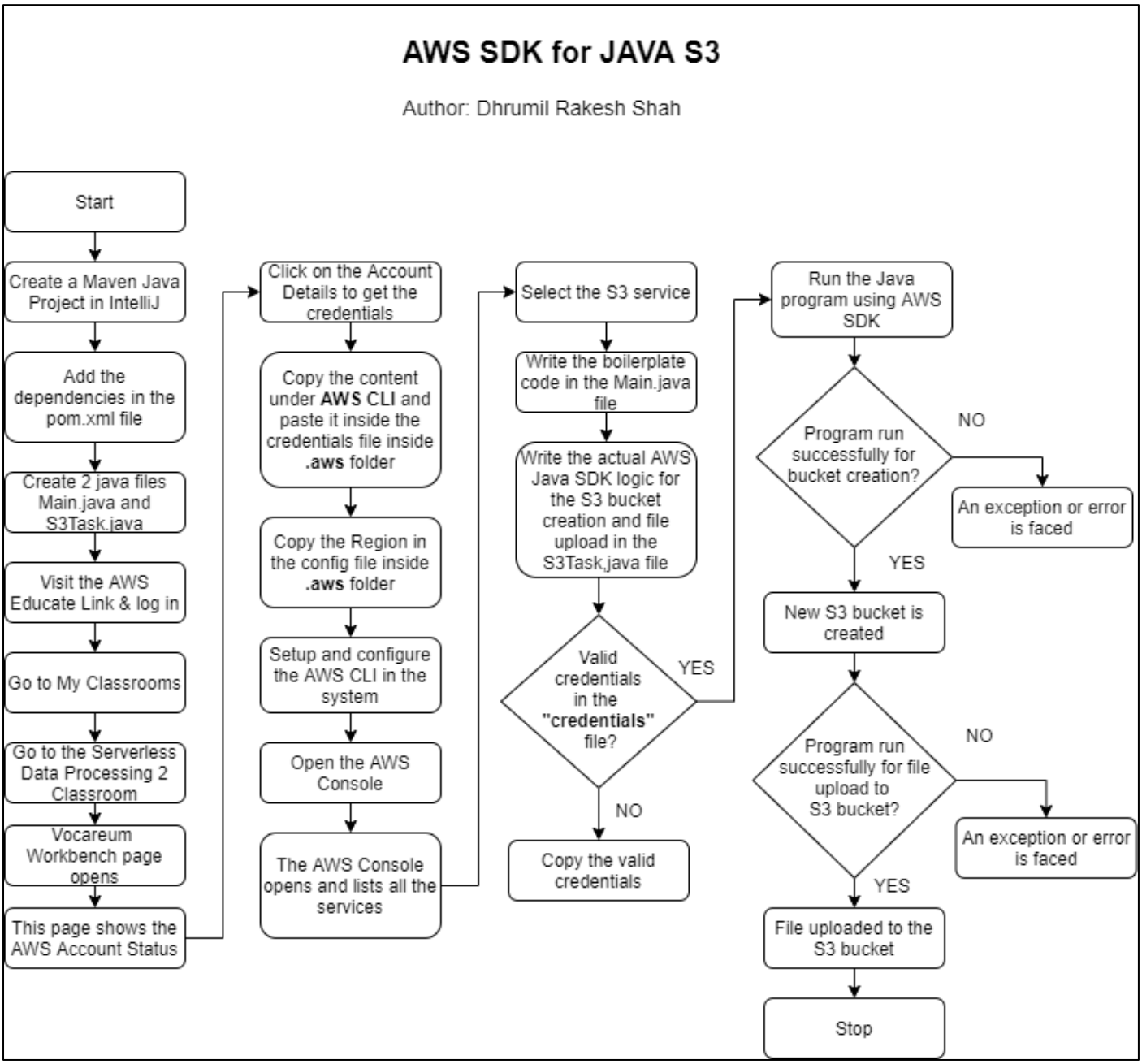

Figure 1. AWS SDK for Java Flowchart

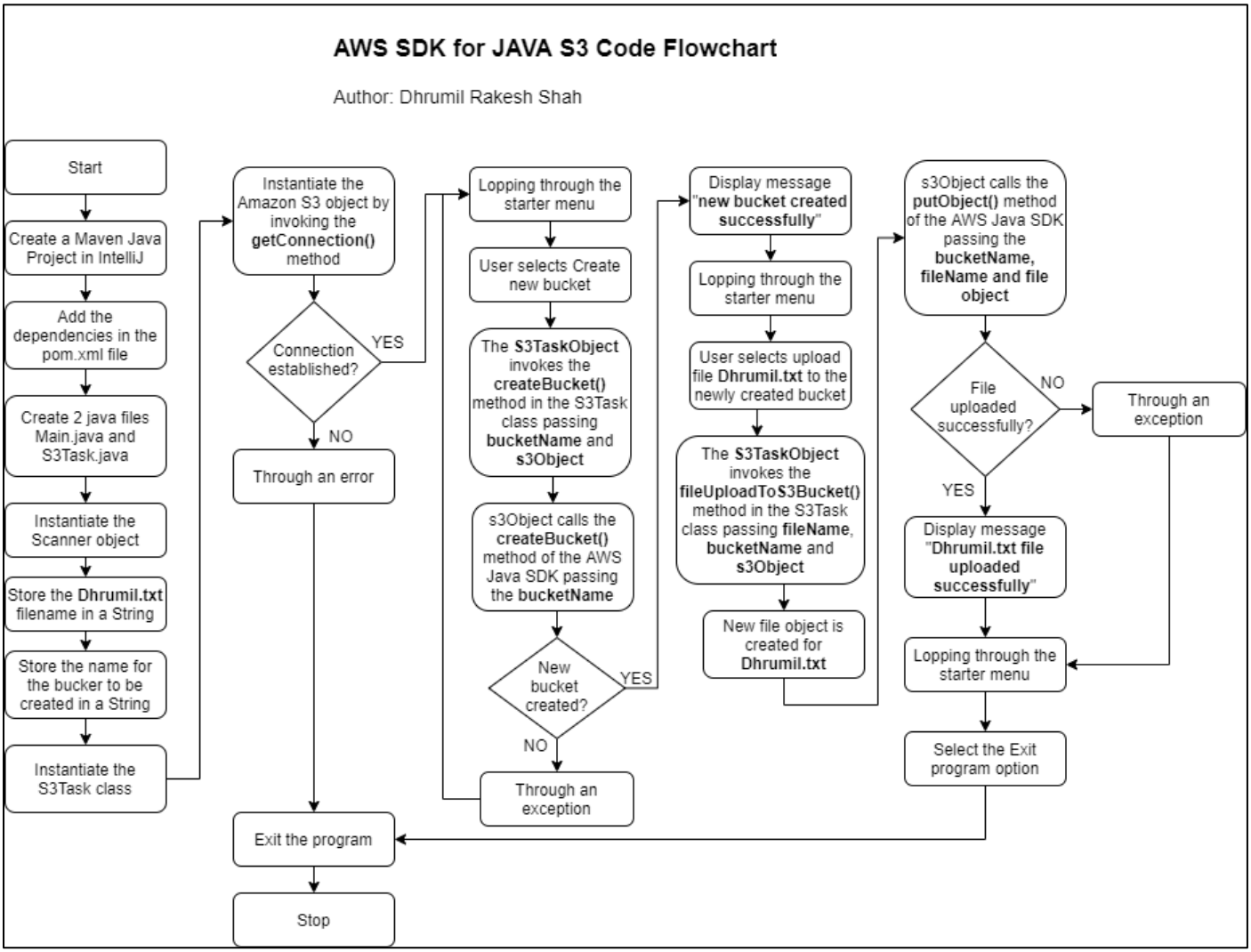

Figure 2: AWS SDK for Java - Code Flowchart

2. Observation of the AWS SDK for Java:

The AWS SDK for Java provides a Java API for various services provided by AWS. So, the idea behind AWS SDK is that we can quickly build a Java application that can interact and perform various operations on Amazon S3, Amazon EC2, DynamoDB and many more. It also supports API lifecycle considerations such as credential management, retries, data marshalling, and even serialization. There are two versions for the AWS SDK for Java, the 1.x and 2.x versions. The AWS SDK for Java is like a major upgrade of the 1.x version as it is built of Java 8+ adding several important features. My observation of AWS SDK for Java was mainly the ease that is provided by AWS to simply connect to use its services. There are so many inbuilt classes and methods that a developer can use to manipulate the services. There are still some features which are not yet supported by the AWS SDK. The very first and the most important step is to establish a connection with AWS. As there are two versions, if not taken care of can cause version mismatch causing connection errors. So, one thing I noticed is that it is preferable to decide upon a version and then stick with that version. Using the inbuilt methods is quite easy like I used the **createBucket ()** method in which I just need to pass the bucketName and the bucket gets created in S3. Similarly, if I want to upload a file to that bucket or to any other existing bucket then there is a function **putObject ()** where bucket name, file name and the file that needs to be uploaded needs to be provided. AWS has made using its SDKs quite seamless and simple making it easy to integrate with the application.

3. Screenshots of all the steps performed:

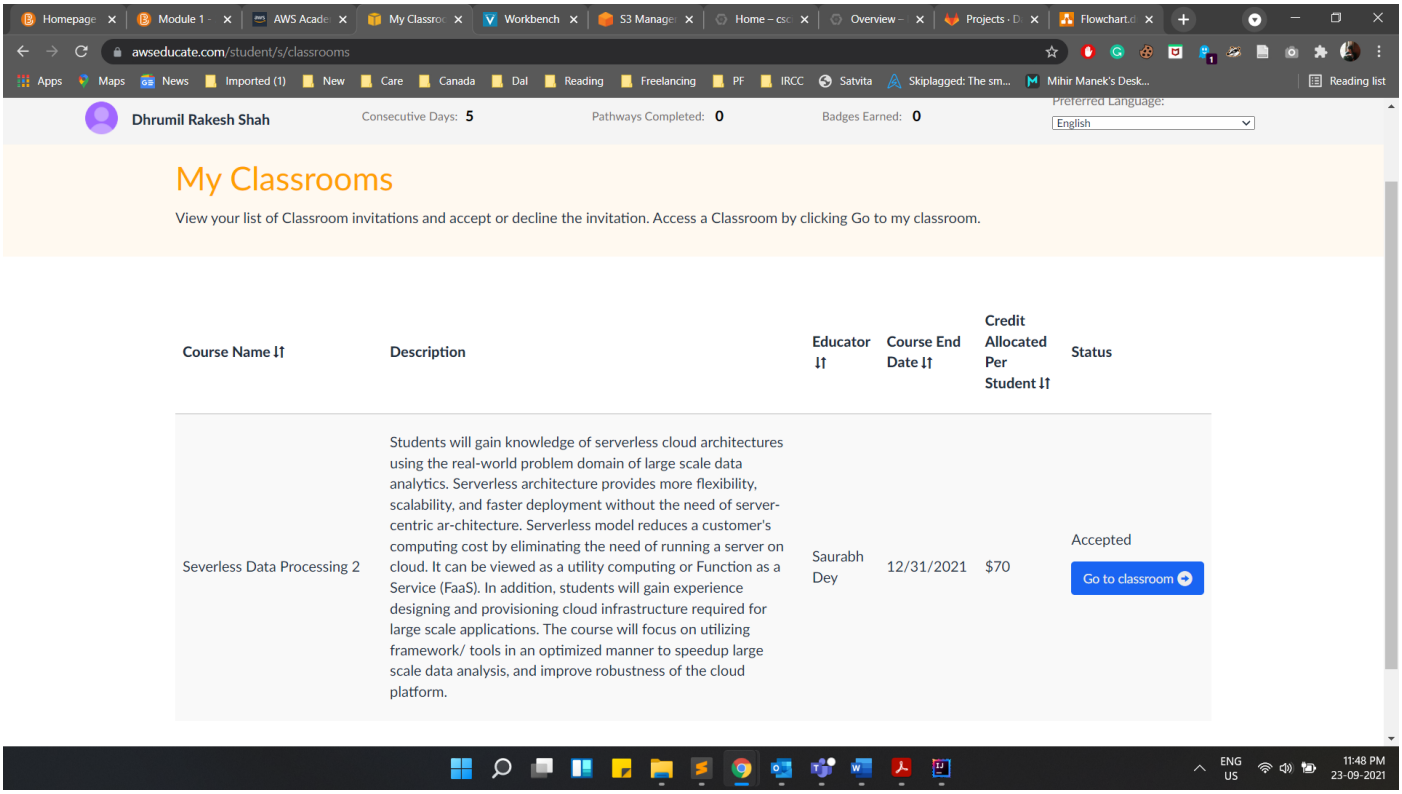

Figure 3: AWS Educate *My Classrooms* page

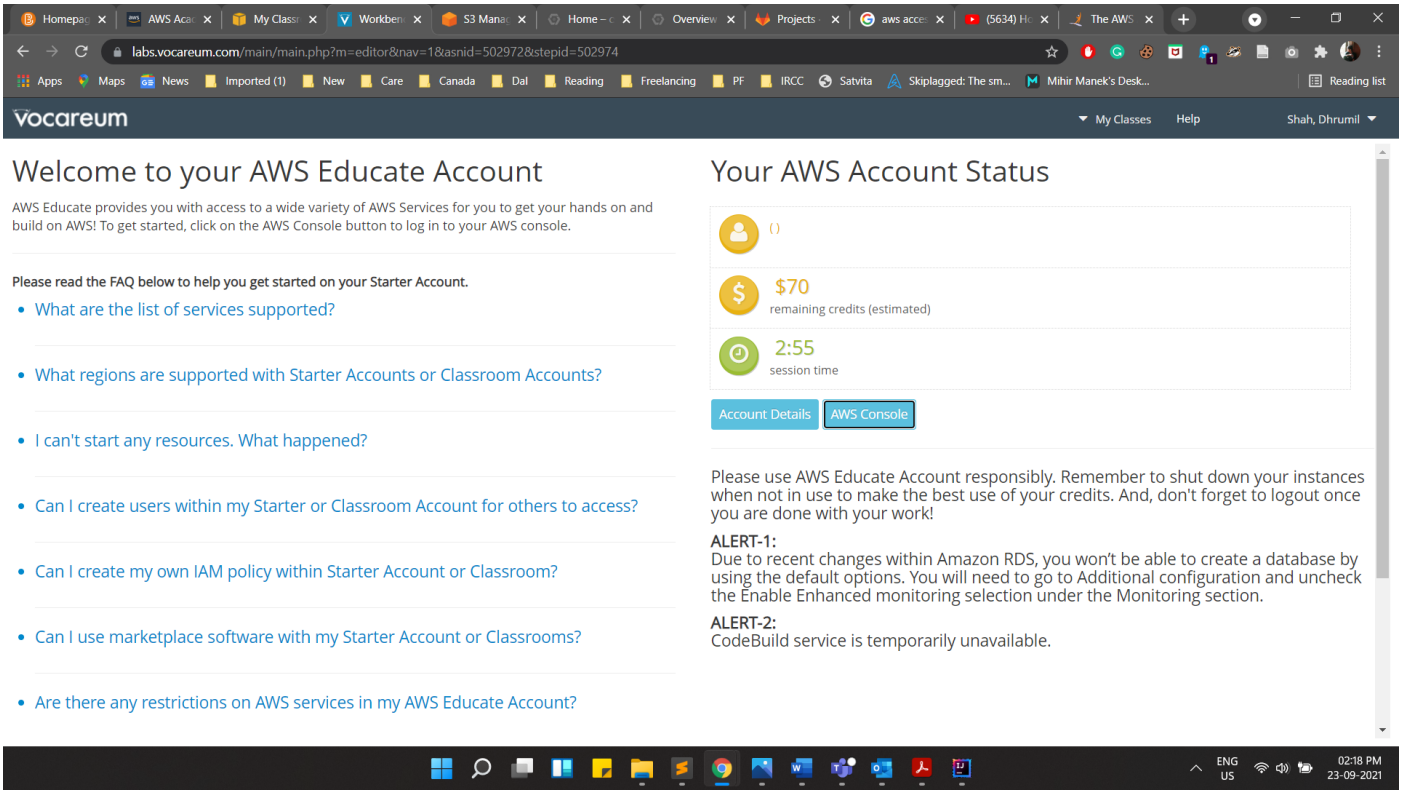

Figure 4: Vocareum Workbench page

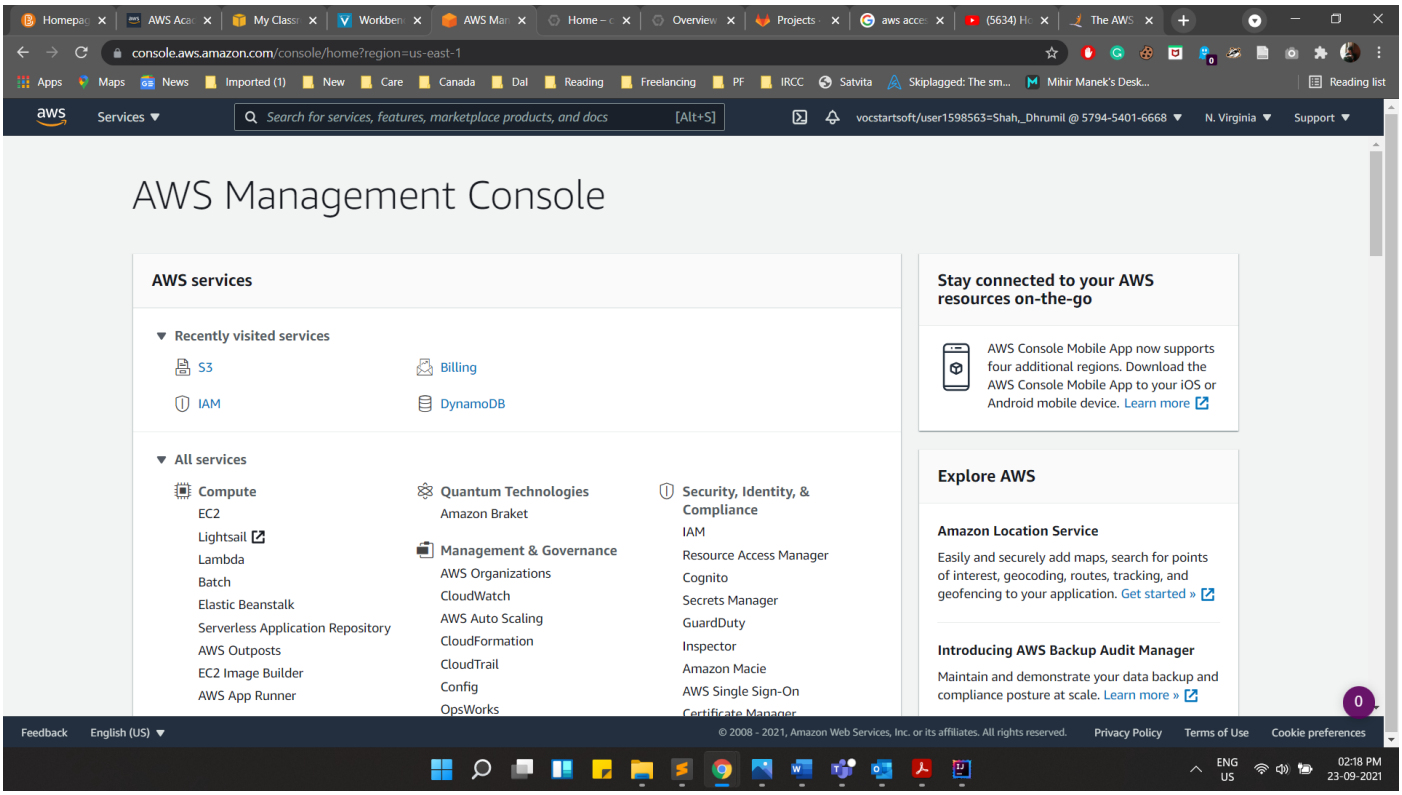

Figure 5: AWS Management Console

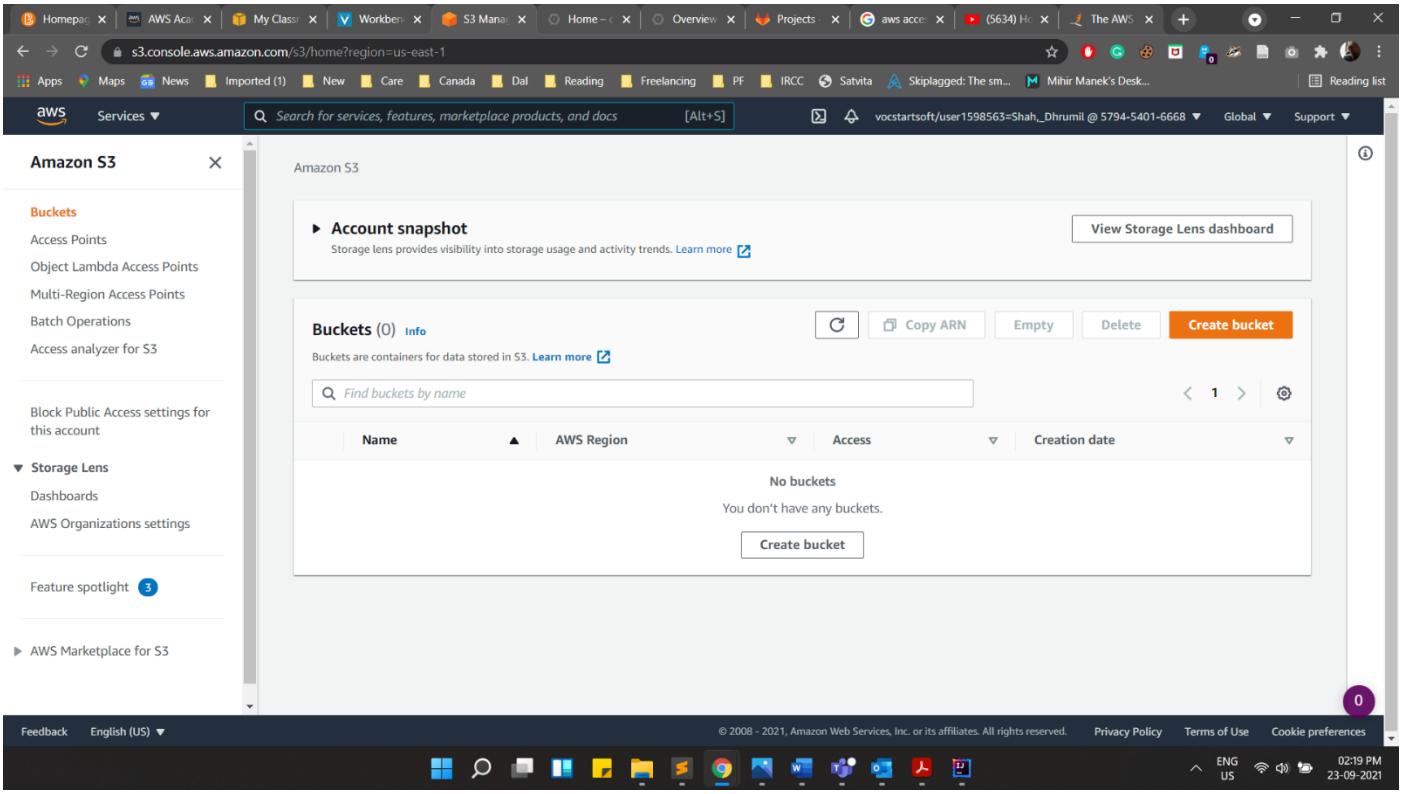

Figure 6: AWS S3 service page

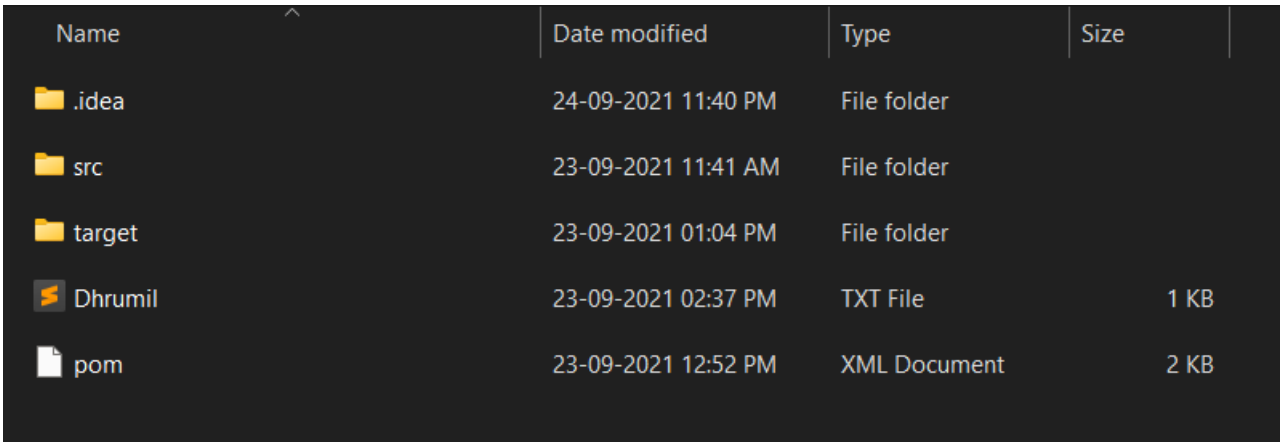

Figure 7: New *Dhrumil.txt* file created

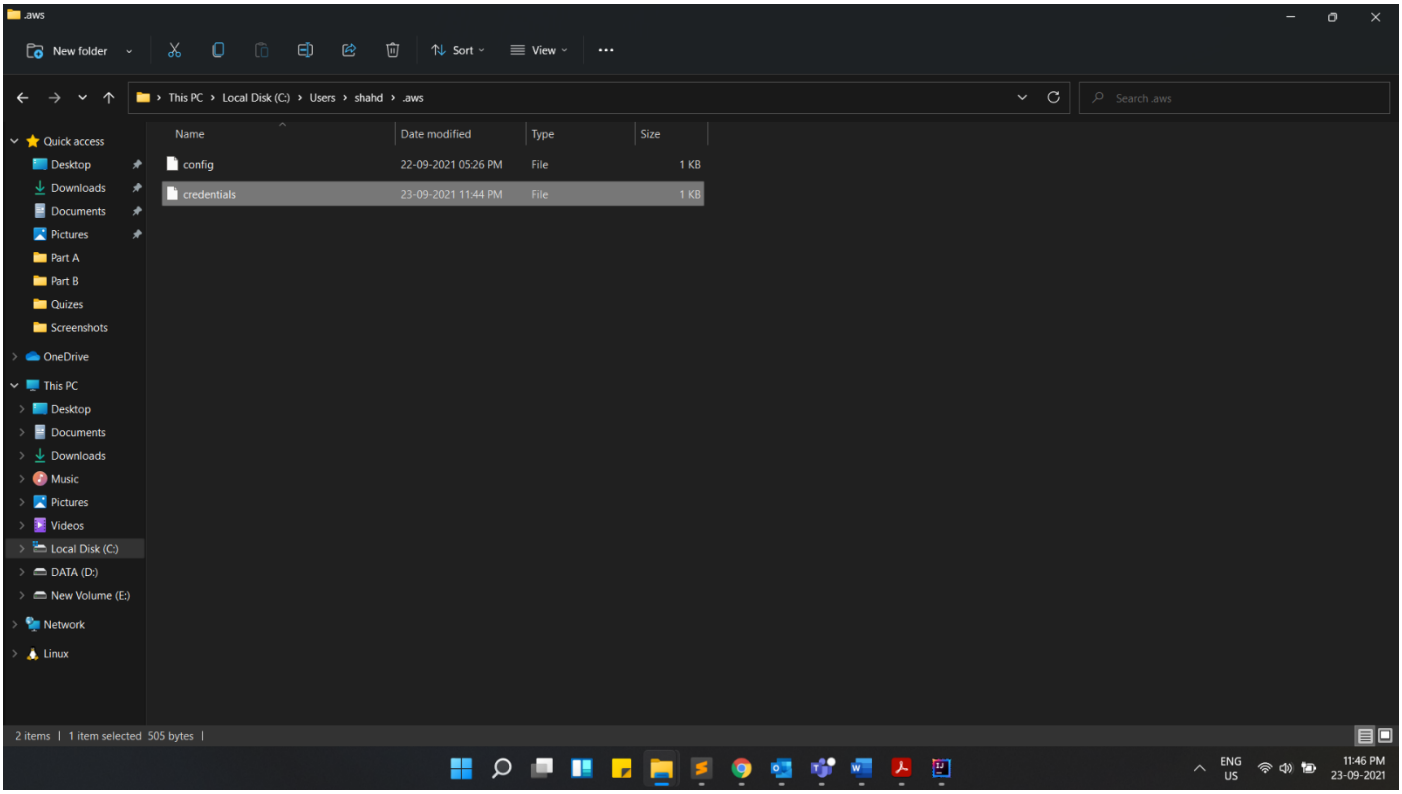

Figure 8: *credentials* file inside *.aws* folder

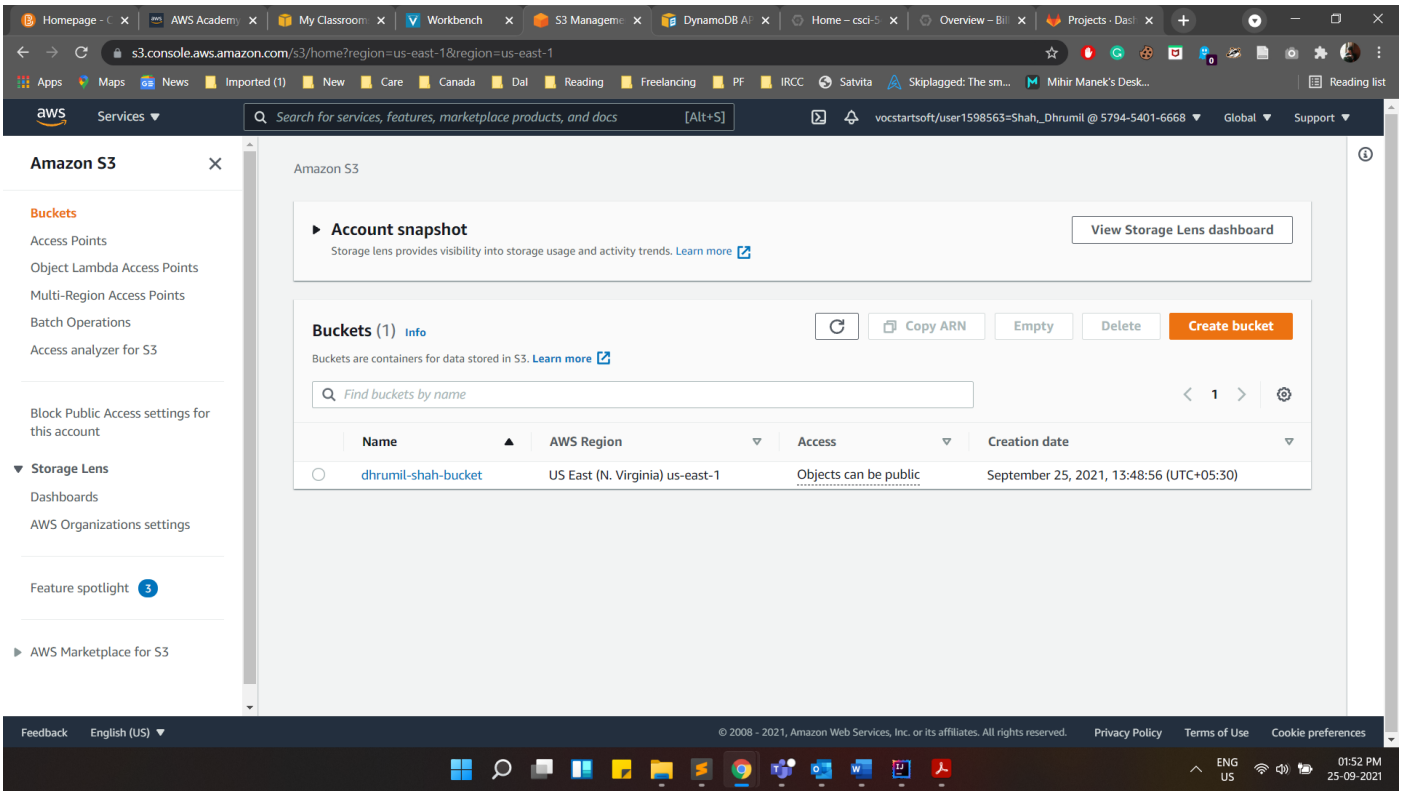

Figure 9: New S3 bucket *dhrumil-shah-bucket* created

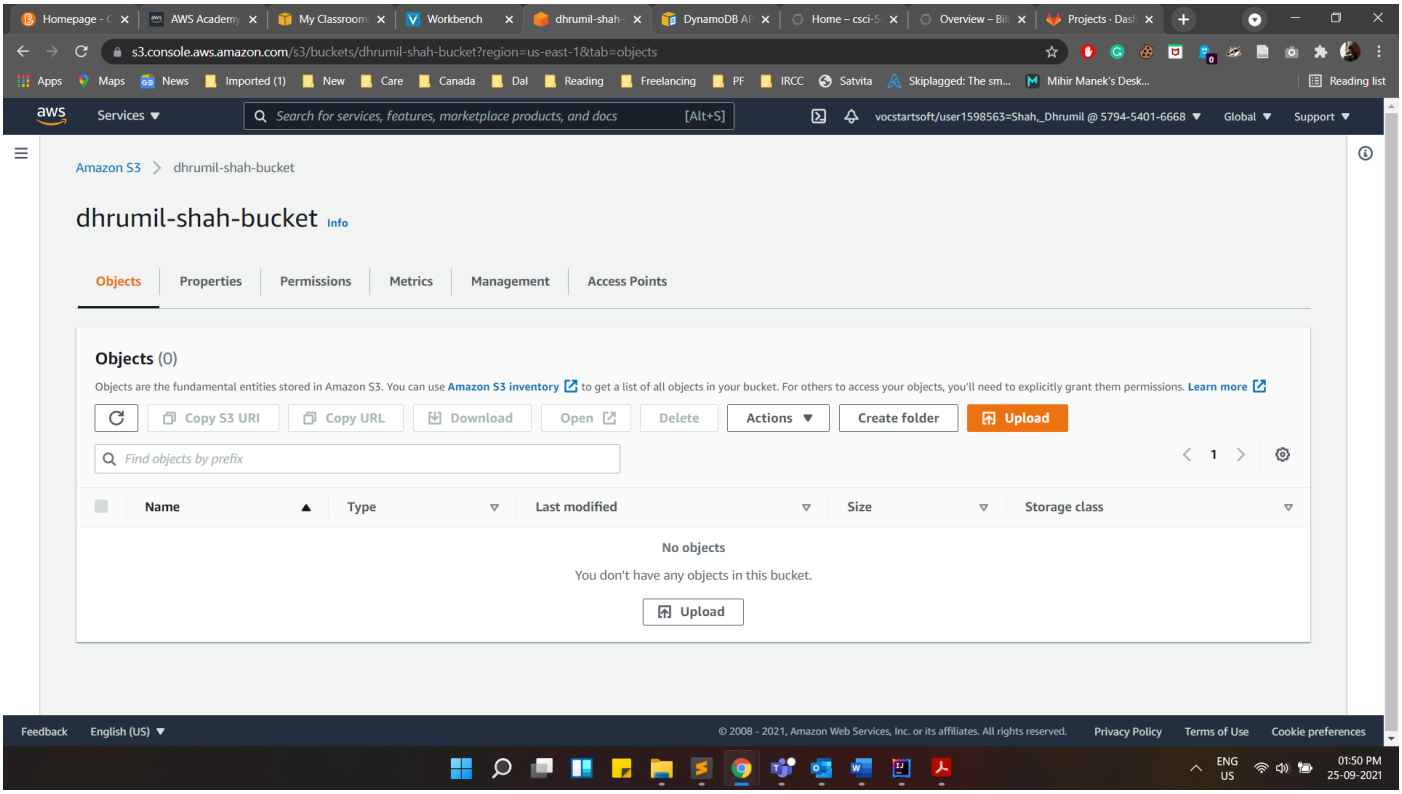

Figure 10: Empty bucket

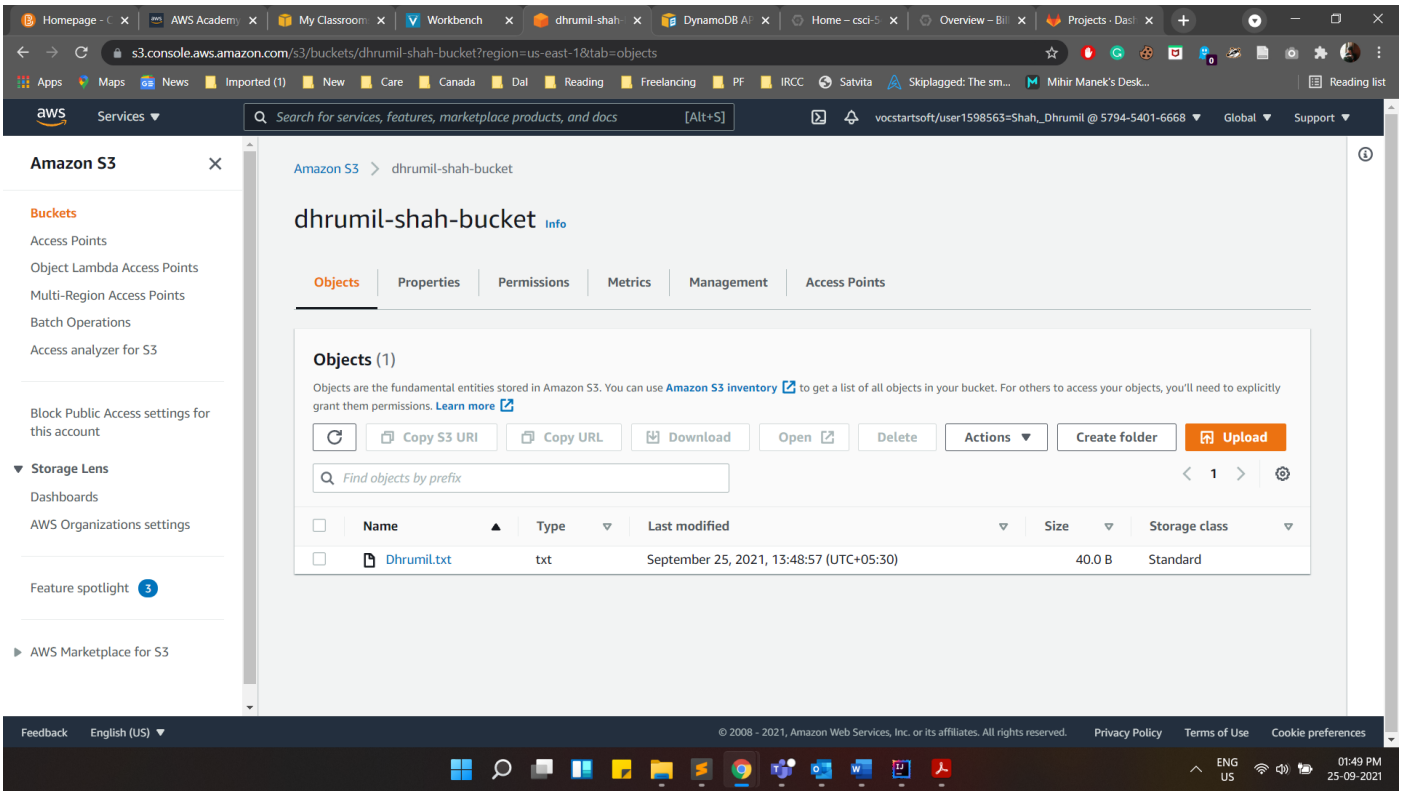

Figure 11: File *Dhrumil.txt* uploaded to bucket *dhrumil-shah-bucket*

4. GitLab Repository Link:

https://git.cs.dal.ca/drshah/dhrumilrakeshshah_csci5410.git

5. Program Script:

Main.java:

```java
import com.amazonaws.services.s3.AmazonS3;

import java.util.Scanner;

/**
 * Author: Dhrumil Rakesh Shah
 * Version: 1.0
 * Class: The Main class containing the boilerplate code of AWS SDK for Java
 */
public class Main {

  // The driver method
  public static void main(String[] args) {

    // Instantiating the Scanner object to read user input
    Scanner sc = new Scanner(System.in);

    // Declaring & Initializing the choice to 0 that stores the
    // choice made by the user
    int choice = 0;

    // Storing the fileName of the file created
    String fileName = "Dhrumil.txt";

    // Storing the bucket name in a String
    String bucketName = "dhrumil-shah-bucket";

    // Creating a new S3Task class object
    S3Task s3TaskObject = new S3Task();

    // Calling the getConnection method in the S3Task class
    // Storing the connection object in s3Object
    AmazonS3 s3Object = s3TaskObject.getConnection();

    // Looping through the menu
    while (choice != -1) {
      System.out.println("Choose any of the below tasks.");
      System.out.println("1. Create a new bucket.");
      System.out.println("2. Upload a file to the bucket.");
      System.out.println("3. Exit.");

      // Reading the user entered choice
      choice = sc.nextInt();

      // Switching through the different choices
      switch (choice) {
        // Creating a new bucket
        case 1:
          s3TaskObject.createBucket(bucketName, s3Object);
          break;
        case 2:
          // Uploading the file to the created bucket
          s3TaskObject.fileUploadToS3Bucket(fileName, s3Object, bucketName);
          break;
```

```
        case 3:
          // Exiting the application
          System.exit(0);
        default:
          // Default switch case
          System.out.println("Enter a valid option.");
          break;
      }
    }
  }
}
```

S3Task.java:

```
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicSessionCredentials;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;

import java.io.File;

/**
 * Author: Dhrumil Rakesh Shah
 * Version: 1.0
 * Class: The AWS SDK for Java helper class
 */
public class S3Task {

  // The method to get connection object of the AWS account
  public AmazonS3 getConnection() {

    // Storing the AWS credentials to establish the connection
    BasicSessionCredentials sessionCredentials = new BasicSessionCredentials(
        "ASIAYN2RGTSOE5Q4C5V4",
        "sRZ9yFppgxacwQKwD73vCMtwParngli8mffeRg6n",
        "FwoGZXIvYXdzEDYaDPkDdnVcKp2hbQrumCK9Afy4RrM7l6rVnVqRbkI" +
            "Gmy9ejYdcfUfPZA8lBAZIhT02zIZ9YkQYSNUOnmCYZUTHXBJjt9w1PdkNrXtK" +
            "0PXj0fcjOPsPLg8g91mJ+cgpb+N/McC+XL94Vl/pDkws0xEo1fZzUjZ4VLSXXA" +
            "z86jjU2LDpaSNZ10tPEwXhrx8dHU2BClyKaScAI6EfV1tX+gd3JrcnWANwxB3C" +
            "QhmGna8Mxv4M1f2L4lE1Nq/kM7f3iSo/4jjsRqtitkKc537bBiix7caKBjItF38" +
            "tpoHmhJolOXA1qZEg9IBegO9ReYuyvU66KD7KR3ijp8rNDtuk82W4LQZ6");

    // Establishing connection with the Amazon S3 service
    // and storing it in the s3Object
    AmazonS3 s3Object = AmazonS3ClientBuilder.standard().withCredentials
            (new AWSStaticCredentialsProvider(sessionCredentials))
        .withRegion(Regions.US_EAST_1)
        .build();

/*
    // Establishing connection with the Amazon S3 service
    // and storing it in the s3Object
    AmazonS3 s3Object = AmazonS3ClientBuilder.standard()
        .withRegion(Regions.US_EAST_1)
        .build();
*/

    // Returning the s3Object
    return s3Object;
  }
```

```
  // The public method to upload the file to a S3 bucket
  public void fileUploadToS3Bucket(String fileName,
                                   AmazonS3 s3Object,
                                   String bucketName) {
    try {
      // Creating a new file with the passed fileName
      File file = new File(fileName);

      // Uploading the file to the S3 bucket
      s3Object.putObject(bucketName, fileName, file);

      // Printing to the console
      System.out.format("The file '%s' is pushed to the '%s' bucket \n",
          fileName, bucketName);

      // Catching the exception
    } catch (Exception e) {
      System.err.println(e.getMessage());
    }
  }

  // The public method to create a new S3 bucket
  public void createBucket(String bucketName, AmazonS3 s3Object) {
    try {
      // Creating a new S3 bucket using the s3Object
      s3Object.createBucket(bucketName);

      // Printing to the console
      System.out.format("A new bucket '%s' is created.\n", bucketName);

      // Catching the exception
    } catch (Exception e) {
      System.err.println(e.getMessage());
    }
  }
}
```

References

[1] "draw.io," JGraph, [Online]. Available: https://app.diagrams.net/.

[2] A. W. Services, "AWS SDK for Java Documentation," Amazon, 2021. [Online]. Available: https://docs.aws.amazon.com/sdk-for-java/index.html.