

Assignment 1

Part C

Meet Patel (B00899516)

Dalhousie University

Subject

**CSCI 5410 (Serverless Data
Processing)**

Professor

Dr. Saurabh Dey

Link to the GitLab repository: https://git.cs.dal.ca/patel13/csci5410_b00899516_meet_patel.git

Link to the package: https://git.cs.dal.ca/patel13/csci5410_b00899516_meet_patel/-/tree/main/Assignment-1/a1_code/src/main/java/part_c_aws_s3

Screenshots of the DynamoDB service

The screenshot displays the AWS Management Console interface for creating a new DynamoDB table. The page title is 'Create table'. The 'Table details' section includes a 'Table name' field with the value 'Parks_NovaScotia' and a 'Partition key' section with a 'Name' field set to 'String'. Below this, there is a 'Sort key - optional' section with a 'Name' field set to 'String'. The 'Settings' section shows 'Default settings' selected. At the bottom, the 'Default settings' section lists two tags: 'owner' with value 'Meet Patel' and 'email' with value 'mt631537@dal.ca'. The 'Create table' button is highlighted in orange.

Table details info
DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name
This will be used to identify your table.
Parks_NovaScotia
Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.
Name String
1 to 255 characters and case sensitive.

Sort key - optional
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.
Enter the sort key name String
1 to 255 characters and case sensitive.

Settings
☒ Default settings
The fastest way to create your table. You can modify these settings now or after your table has been created.
☐ Customize settings
Use these advanced features to make DynamoDB work better for your needs.

Default settings
Looking for language selection? Find it in the new Unified Settings.

Key Value - optional
owner Meet Patel Remove
email mt631537@dal.ca Remove
Add new tag
You can add 48 more tags.

Cancel Create table

Figure 1: Screenshot of the *Parks_NovaScotia* table structure setup

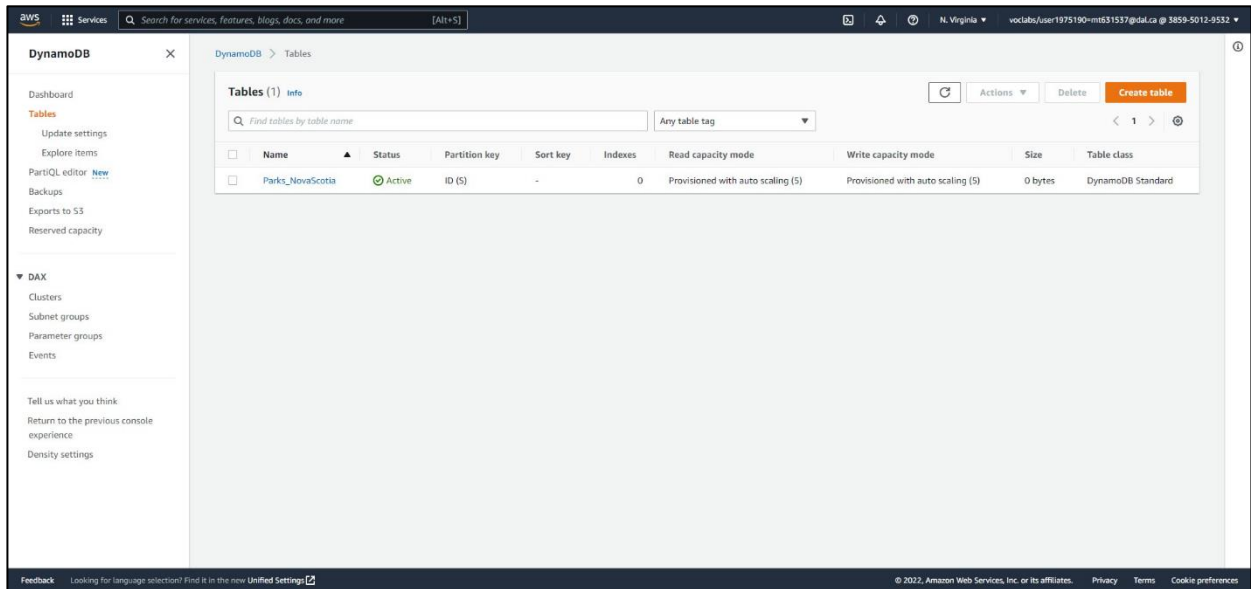


Figure 2: Screenshot shows the creation of *Parks_NovaScotia* table on DynamoDB

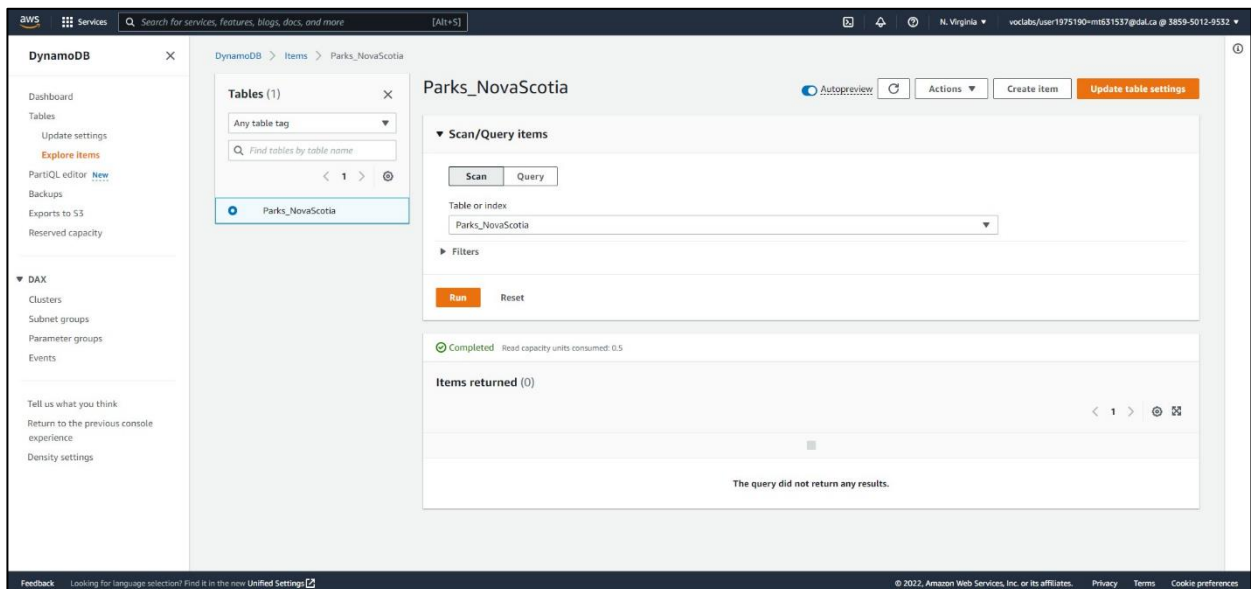


Figure 3: Screenshot shows the empty *Parks_NovaScotia* table on DynamoDb

CSCI 5410: Assignment #1 (Part C) | Meet Patel B00899516

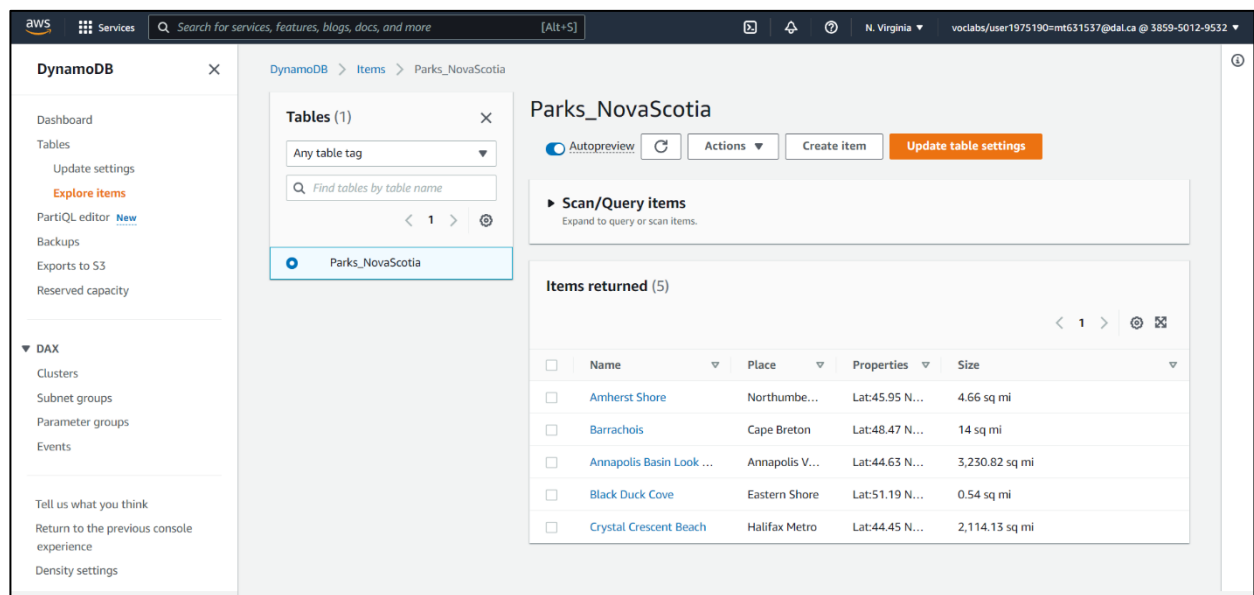


Figure 4: Screenshot shows the inserted items in the **Parks_NovaScotia** table

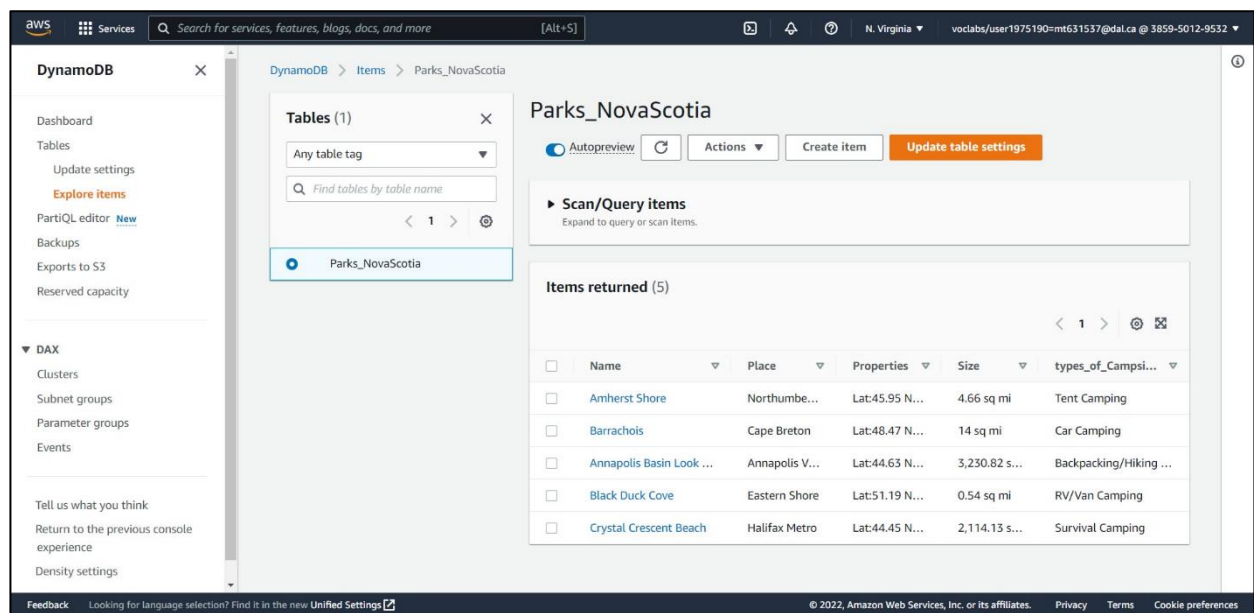


Figure 5: Screenshot shows the successful update of the item by adding `type_of_campsite` to every item in the table using java code. **Note: The parks which I have considered here have only one camp site**

Code Output

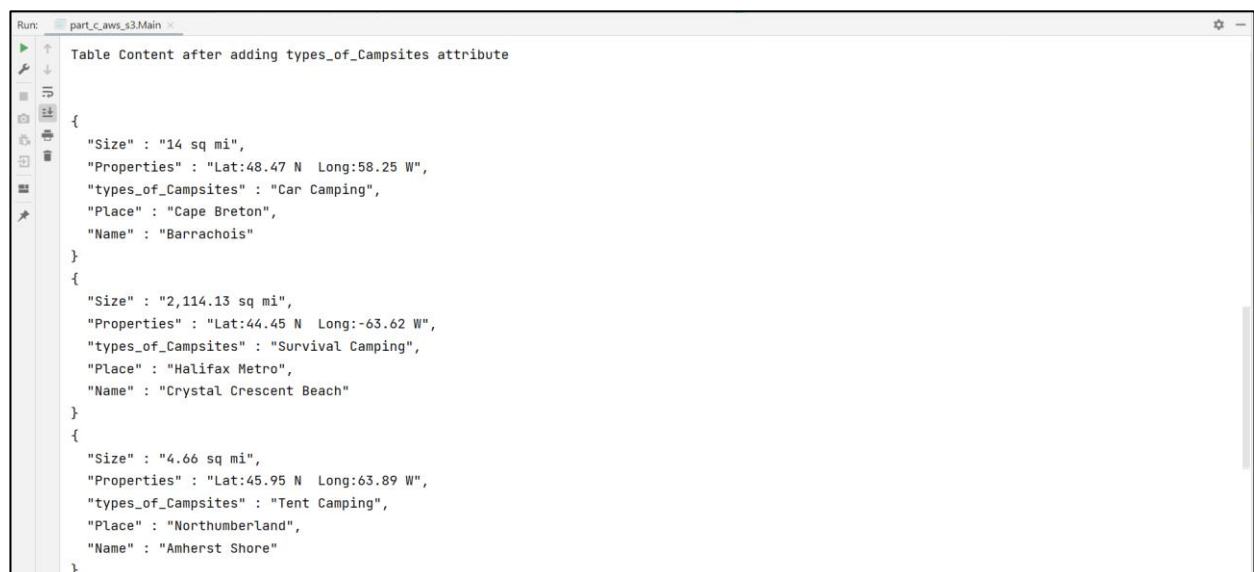


The screenshot shows a code editor window titled 'part_c_aws_s3.Main'. The code displays the content of a table before adding the 'types_of_Campsites' attribute. The table contains three rows of data, each represented as a JSON object with fields for Size, Properties, Place, and Name.

```
Table Content before adding types_of_Campsites attribute

{
  "Size" : "14 sq mi",
  "Properties" : "Lat:48.47 N Long:58.25 W",
  "Place" : "Cape Breton",
  "Name" : "Barrachois"
}
{
  "Size" : "2,114.13 sq mi",
  "Properties" : "Lat:44.45 N Long:-63.62 W",
  "Place" : "Halifax Metro",
  "Name" : "Crystal Crescent Beach"
}
{
  "Size" : "4.66 sq mi",
  "Properties" : "Lat:45.95 N Long:63.89 W",
  "Place" : "Northumberland",
  "Name" : "Amherst Shore"
}
```

Figure 6: Screenshot shows table content before adding types_of_Campsite Item



The screenshot shows the same code editor window as Figure 6, but now the 'types_of_Campsites' attribute has been added to each row of the table. The JSON objects now include an additional field for the type of campsite.

```
Table Content after adding types_of_Campsites attribute

{
  "Size" : "14 sq mi",
  "Properties" : "Lat:48.47 N Long:58.25 W",
  "types_of_Campsites" : "Car Camping",
  "Place" : "Cape Breton",
  "Name" : "Barrachois"
}
{
  "Size" : "2,114.13 sq mi",
  "Properties" : "Lat:44.45 N Long:-63.62 W",
  "types_of_Campsites" : "Survival Camping",
  "Place" : "Halifax Metro",
  "Name" : "Crystal Crescent Beach"
}
{
  "Size" : "4.66 sq mi",
  "Properties" : "Lat:45.95 N Long:63.89 W",
  "types_of_Campsites" : "Tent Camping",
  "Place" : "Northumberland",
  "Name" : "Amherst Shore"
}
```

Figure 7: Screenshot shows table content after adding types_of_Campsite Item

Program Script

AWSConnection.java

This java class is used to establish connection with the [Amazon S3 service](#)

```
package part_c_aws_s3;

import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicSessionCredentials;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.dynamodbv2.AmazonDynamoDB;
import com.amazonaws.services.dynamodbv2.AmazonDynamoDBClientBuilder;
import com.amazonaws.services.dynamodbv2.document.DynamoDB;

public class DynamoDBConnection {
    private static final String AWS_ACCESS_KEY_ID = "ASIAVTXDLTV6ASJ67LBM";
    private static final String AWS_SECRET_ACCESS_KEY = "qNkMmdezImxgu2pldEar7HRDt966aFxbKSUwuFw";
    private static final String AWS_SESSION_TOKEN = "FwoGZXIvYXdzEN3////////wEaDDIQCJb6TB8qf8439CLA" +
        "AbXtHAVE5Ltr0H0CNL2l+TaBEMsd9BeWcJnvoW9ldiAz7pBA7uPQG8ox6U/0M7QLPA3BT+/rgVnCf88jW3TU2z1ft" +
        "jUmMqpOoWlg3AkeE9JGjkDlqME99CuHrIcT22y+48UcFwGLP7EsyS5oujaeaiC/Rgq+5Du1MI/LRCrLROk9Ym58dE" +
        "s9vYDPyJEatrE2d1Dlsz93LUiAGiXg/BM5CHfKmhJoOlsBorh6/bBJRgMpYRGCRVKh79rsyXEZYydlqCiAgcGUBjI" +
        "tvPCuqdgA4Jl0cQ5pBqBm8yjpKbwMW/VE1LxvUb07MUUZLl1cJ3oF40Gqs5S3";

    public DynamoDB createDynamoDBClientBuilder() {
        BasicSessionCredentials basicSessionCredentials = new BasicSessionCredentials(AWS_ACCESS_KEY_ID,
            AWS_SECRET_ACCESS_KEY, AWS_SESSION_TOKEN);

        AmazonDynamoDB awsDynamoDB =
            AmazonDynamoDBClientBuilder.standard().withCredentials
                (new AWSStaticCredentialsProvider(basicSessionCredentials))
                .withRegion(Regions.US_EAST_1)
                .build();

        DynamoDB dynamoDB = new DynamoDB(awsDynamoDB);
        return dynamoDB;
    }
}
```

AddAttribute.java

This java class is used to add the [type_of_campsite](#) item to the table

```
package part_c_aws_s3;

import com.amazonaws.services.dynamodbv2.document.*;
import com.amazonaws.services.dynamodbv2.document.spec.UpdateItemSpec;
import com.amazonaws.services.dynamodbv2.document.utils.NameMap;
import com.amazonaws.services.dynamodbv2.document.utils.ValueMap;
import com.amazonaws.services.dynamodbv2.model.ReturnValue;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class AddAttribute {
    DynamoDBConnection dynamoDBConnection;
    List<Map<String, Object>> items;

    public AddAttribute() {
        dynamoDBConnection = new DynamoDBConnection();
        items = new ArrayList<>();
    }

    public void createNewAttribute(String tableName) {
        Table table = dynamoDBConnection.createDynamoDBClientBuilder().getTable(tableName);

        try {
            for (Item dynamoDBItem: table.scan()){
                Map<String, Object> map = new HashMap<>();
                map.put("Name", dynamoDBItem.asMap().get("Name"));
                addTypeOfCampsites(dynamoDBItem, map);
                items.add(map);
            }
            updateAttributeToItem(table);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

    }
}
private void addTypeOfCampsites(Item dynamoDBItem, Map<String, Object> map) {
    if (dynamoDBItem.asMap().get("Name").toString().toLowerCase().contains(
        "Amherst Shore".toLowerCase())) {
        map.put("types_of_Campsites", "Tent Camping");
    } else if (dynamoDBItem.asMap().get("Name").toString().toLowerCase().contains(
        "Annapolis Basin Look Off".toLowerCase())) {
        map.put("types_of_Campsites", "Backpacking/Hiking Camping");
    } else if (dynamoDBItem.asMap().get("Name").toString().toLowerCase().contains(
        "Barrachois".toLowerCase())) {
        map.put("types_of_Campsites", "Car Camping");
    } else if (dynamoDBItem.asMap().get("Name").toString().toLowerCase().contains(
        "Black Duck Cove".toLowerCase())) {
        map.put("types_of_Campsites", "RV/Van Camping");
    } else if (dynamoDBItem.asMap().get("Name").toString().toLowerCase().contains(
        "Crystal Crescent Beach".toLowerCase())) {
        map.put("types_of_Campsites", "Survival Camping");
    } else {
        map.put("types_of_Campsites", "");
    }
}
private void updateAttributeToItem(Table table) {
    for (Map<String, Object> map : items) {
        if (map.get("types_of_Campsites").toString().length() > 1) {
            UpdateItemSpec updateItemSpec =
                new UpdateItemSpec().withPrimaryKey("Name",
                    map.get("Name").toString())
                    .withUpdateExpression("set #keyAttribute = :valueAttribute")
                    .withNameMap(new NameMap()
                        .with("#keyAttribute", "types_of_Campsites"))
                    .withValueMap(new ValueMap()
                        .withString(":valueAttribute", map.get("types_of_Campsites")
                            .toString()).withReturnValues(ReturnValue.ALL_NEW);
            table.updateItem(updateItemSpec);
        }
    }
}
}

```

RetrieveData.java

This java class is used to retrieve the dynamo table content. The Output is shown in the figure 6 & 7.

```

package part_c_aws_s3;

import com.amazonaws.services.dynamodbv2.document.*;
import com.amazonaws.services.dynamodbv2.document.spec.UpdateItemSpec;
import com.amazonaws.services.dynamodbv2.document.utils.NameMap;
import com.amazonaws.services.dynamodbv2.document.utils.ValueMap;
import com.amazonaws.services.dynamodbv2.model.ReturnValue;

import java.util.List;

public class RetrieveData {
    public void retrieveTableContent(String tableName){
        DynamoDBConnection dynamoDBConnection = new DynamoDBConnection();
        try {
            TableKeysAndAttributes tableKeysAndAttributes = new TableKeysAndAttributes(tableName);
            tableKeysAndAttributes.addHashOnlyPrimaryKeys("Name",
                "Amherst Shore",
                "Annapolis Basin Look Off",
                "Barrachois",
                "Black Duck Cove",
                "Crystal Crescent Beach");
            BatchGetItemOutcome batchGetItemOutcome =
                dynamoDBConnection.createDynamoDBClientBuilder()
                    .batchGetItem(tableKeysAndAttributes);
            for (String name : batchGetItemOutcome.getTableItems().keySet()) {
                List<Item> items = batchGetItemOutcome.getTableItems().get(name);
                for (Item item : items) {
                    System.out.println(item.toJSONPretty());
                }
            }
        }
    }
}

```

Main.java

The Main class containing the boilerplate code of AWS SDK for Java

```
package part_c_aws_s3;

public class Main {
    public static void main(String[] args) {

        DynamoDBConnection dynamoDBConnection = new DynamoDBConnection();
        dynamoDBConnection.createDynamoDBClientBuilder();

        System.out.println("\n");
        System.out.println("Table Content before adding types_of_Campsites attribute");
        System.out.println("\n");
        RetrieveData retrieveBefore = new RetrieveData();
        retrieveBefore.retrieveTableContent("Parks_NovaScotia");

        System.out.println("\n");
        System.out.println("-----");
        System.out.println("\n");

        System.out.println("Table Content after adding types_of_Campsites attribute");
        System.out.println("\n");
        AddAttribute addAttribute = new AddAttribute();
        addAttribute.createNewAttribute("Parks_NovaScotia");
        RetrieveData retrieveAfter = new RetrieveData();
        retrieveAfter.retrieveTableContent("Parks_NovaScotia");

    }
}
```


References

- [1] A. W. Services, "AWS SDK for Java Documentation," *Amazon*, 2022. [Online]. Available: <https://docs.aws.amazon.com/sdk-for-java/index.html> [Accessed: May 26, 2022].
- [2] Amazon and AWS, "Cloud Services - Amazon Web Services (AWS)," *Amazon*, [Online]. Available: <https://aws.amazon.com/>. [Accessed: May 26, 2022].
- [3] Amazon and AWS, "DynamoDB Examples Using the AWS SDK for Java," [Online]. Available: <https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/examplesdynamodb.html>. [Accessed: May 26, 2022]
- [4] "Home | Nova Scotia Parks," *Parks.novascotia.ca*, 2022. [Online]. Available: <https://parks.novascotia.ca> [Accessed: May 26, 2022].