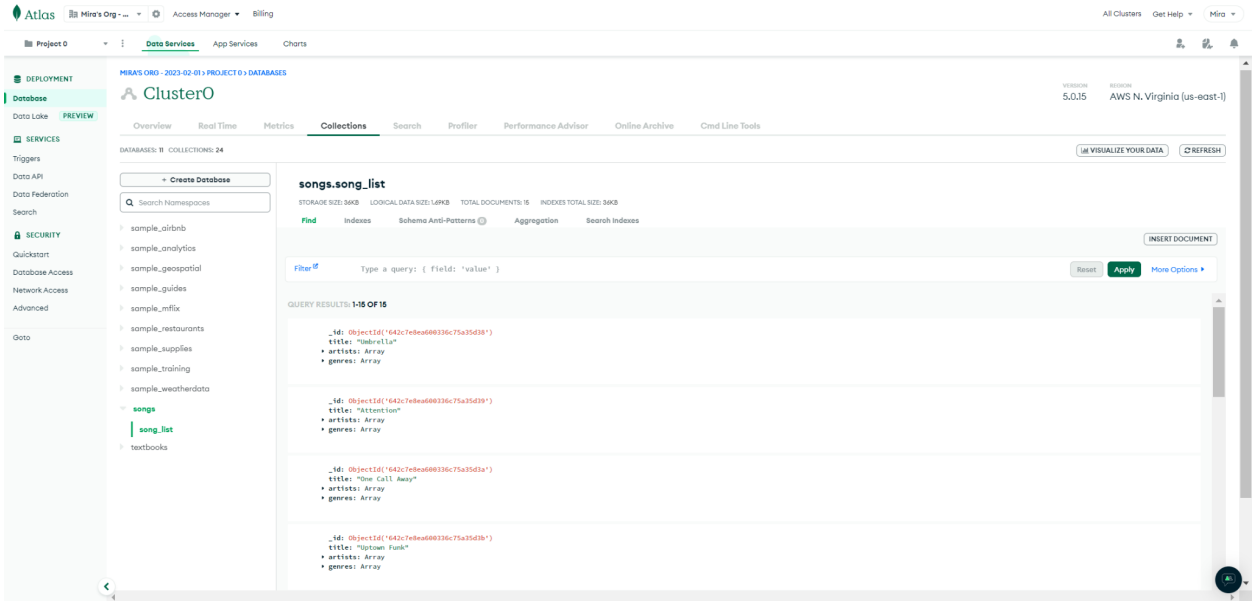
Part 1:



Part 2:

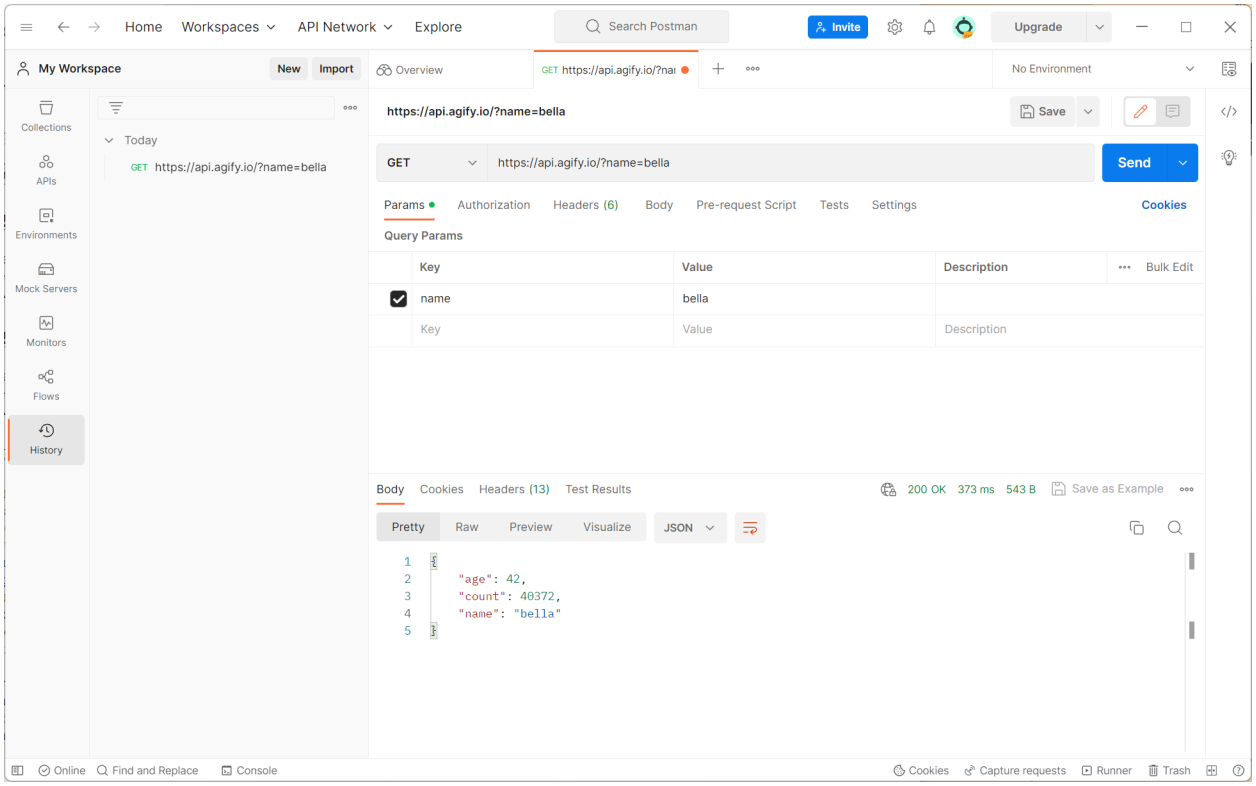
```
Current Mongosh Log ID: 642ca4b20b936c88a371e859
Connecting to:          mongodb+srv://<credentials>@cluster0.gww8kzq.mongodb.net/?appName=mongosh+1.8.0
Using MongoDB:          5.0.15 (API Version 1)
Using Mongosh:          1.8.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Atlas atlas-hbdkil-shard-0 [primary] test> use songs
switched to db songs
Atlas atlas-hbdkil-shard-0 [primary] songs> show collections
song_list
Atlas atlas-hbdkil-shard-0 [primary] songs> db.song_list.find()
[
  {
    _id: ObjectId("642c7e8ea600336c75a35d38"),
    title: 'Umbrella',
    artists: [ 'Rihanna', 'Jay-Z' ],
    genres: [ 'Pop', 'Hip-Hop' ]
  },
  {
    _id: ObjectId("642c7e8ea600336c75a35d39"),
    title: 'Attention',
    artists: [ 'Charlie Puth' ],
    genres: [ 'Pop' ]
  },
  {
    _id: ObjectId("642c7e8ea600336c75a35d3a"),
    title: 'One Call Away',
    artists: [ 'Charlie Puth' ],
    genres: [ 'Pop' ]
  },
```

```
{
  _id: ObjectId("642c7e8ea600336c75a35d3b"),
  title: 'Uptown Funk',
  artists: [ 'Bruno Mars', 'Mark Ronson' ],
  genres: [ 'Pop' ]
},
{
  _id: ObjectId("642c7e8ea600336c75a35d3c"),
  title: 'Symphony',
  artists: [ 'Clean Bandit', 'Zara Larsson' ],
  genres: [ 'Pop' ]
},
{
  _id: ObjectId("642c7e8ea600336c75a35d3d"),
  title: 'Hey Soul Sister',
  artists: [ 'Train' ],
  genres: [ 'Pop' ]
},
{
  _id: ObjectId("642c7e8ea600336c75a35d3e"),
  title: 'Bohemian Rhapsody',
  artists: [ 'Queen' ],
  genres: [ 'Rock' ]
},
{
  _id: ObjectId("642c7e8ea600336c75a35d3f"),
  title: "Hips Don't Lie",
  artists: [ 'Shakira', 'Wyclef Jean' ],
  genres: [ 'Pop', 'Latin' ]
},
{
  _id: ObjectId("642c7e8ea600336c75a35d40"),
  title: 'Defying Gravity',
  artists: [ 'Kristin Chenoweth', 'Idina Menzel' ],
  genres: [ 'Broadway' ]
},
{
  _id: ObjectId("642c7e8ea600336c75a35d41"),
  title: 'Chammak Challo',
  artists: [ 'Akon', 'Hamsika Ayer' ],
  genres: [ 'Bollywood' ]
},
{
  _id: ObjectId("642c7e8ea600336c75a35d42"),
  title: 'HIP',
  artists: [ 'MAMAMOO' ],
  genres: [ 'K-pop' ]
},
{
  _id: ObjectId("642c7e8ea600336c75a35d43"),
  title: '(Crank That) Soulja Boy',
  artists: [ 'Soulja Boy' ],
  genres: [ 'Hip-Hop' ]
},
{
```

```
    _id: ObjectId("642c7e8ea600336c75a35d44"),
    title: 'Satisfied',
    artists: [ 'Renée Elise Goldsberry' ],
    genres: [ 'Broadway' ]
  },
  {
    _id: ObjectId("642c7e8ea600336c75a35d45"),
    title: 'Purple Rain',
    artists: [ 'Prince' ],
    genres: [ 'Rock', 'Soul' ]
  },
  {
    _id: ObjectId("642c7e8ea600336c75a35d46"),
    title: 'Despacito',
    artists: [ 'Luis Fonsi', 'Daddy Yankee' ],
    genres: [ 'Latin', 'Pop' ]
  }
]
Atlas atlas-hbdkil-shard-0 [primary] songs> db.song_list.find({genres:"Pop"},{title:1,_id:0})
[
  { title: 'Umbrella' },
  { title: 'Attention' },
  { title: 'One Call Away' },
  { title: 'Uptown Funk' },
  { title: 'Symphony' },
  { title: 'Hey Soul Sister' },
  { title: "Hips Don't Lie" },
  { title: 'Despacito' }
]
Atlas atlas-hbdkil-shard-0 [primary] songs>
```

Questions:

1. Identify the tools used to work with
    a. an SQL database
       PHP, MySQL,
    b. a NoSQL database.
       MongoDB, Node,
2. Which database do you prefer working with so far- SQL or NoSQL? Explain your answer.
   I prefer NoSQL because it is easier to understand when I don't have to think about relationships between tables.



Code:

```
<!doctype html>
<html>
<head>
    <title>Names API</title>
    <meta charset="utf-8"/>
    <style>
        li {font-weight: bold;}

    </style>
    <script src="https://code.jquery.com/jquery-3.6.4.js"
integrity="sha256-a9jBBRygX1Bh5lt8GZjXDzyOB+bWve9EiO7tROUtj/E=" crossorigin="anonymous"></script>
    <script>
    function getAge() {

        data_ajax = document.getElementById("data_ajax");
        data_fetch = document.getElementById("data_fetch");
        name = document.getElementById("name").value;
        if (name=="") {
            alert("Enter a name.")
```

```
        return false;
    }

    data_ajax.innerHTML = "Loading...";
    data_fetch.innerHTML = "Loading...";

    // AJAX
    req = new XMLHttpRequest();
    req.open("GET", "https://api.agify.io/?name="+name, true)

    req.onreadystatechange = function() {
        if (req.readyState == 4) {
            json_str = req.responseText;
            json_obj = JSON.parse(json_str);

            if (req.status == 200) {
                data_ajax.innerHTML = disp_json(json_obj);
            }
            else {
                error_msg = "Error " + req.status + ": " + json_obj['error'];
                data_ajax.innerHTML = error_msg;
            }
        }
    }
    req.send();

    // FETCH
    fetch("https://api.agify.io/?name="+name)
    .then((response) => response.json())
    .then((data) => {data_fetch.innerHTML = disp_json(data);})
    .catch(data_fetch.innerHTML = "An error occurred")


    // DISPLAY DATA
    function disp_json(json_obj) {
        disp_html = "Name: " + json_obj['name'] + "<br />";
        disp_html += "Occurrences: " + json_obj['count'] + "<br />";
        disp_html += "Predicted Age: " + json_obj['age'] + "<br />";

        return disp_html;
    }
```

```
    return false;
    }
    </script>
</head>


<body>
    <h1>Agify.io</h1>
    <form method="" onsubmit="getAge(); return false;" id="form1">
        <label for="txtNameId">Enter a name to Agify</label> <br />
        <input type="text" id="name" placeholder="Bella">
        <br>
        <input type="submit" value="Get Response">
    </form>

    <h2>Getting data using AJAX</h2>
    <div id="data_ajax"></div>
    <h2>Getting data using fetch() function</h2>
    <div id="data_fetch"></div>
    <h2>About:</h2>
    <div id="about">
        <ol>
            <li> Describe the API you selected and what it does.</li>
            I chose the Agify.io API. It takes a name as input and returns the predicted age of people
with that name.
            <li>Cite the website where you found it.</li>
            I found this API on <a
href="https://mixedanalytics.com/blog/list-actually-free-open-no-auth-needed-apis/">https://mixedanalyt
ics.com/blog/list-actually-free-open-no-auth-needed-apis/</a>.
            <li>Describe the options you used for the API request.</li>
            I used the option to query a name. I also used the errors returned by the API to display
error messages. There weren't any other options available.
            <li>Give two applications where this API would be helpful.</li>
            This would be helpful in social research for finding out the average age of commenters on a
website. It could also be helpful for creative applications like making sure the names of fictional
characters match their ages.
        </ol>
    </div>
</body>
</html>
```