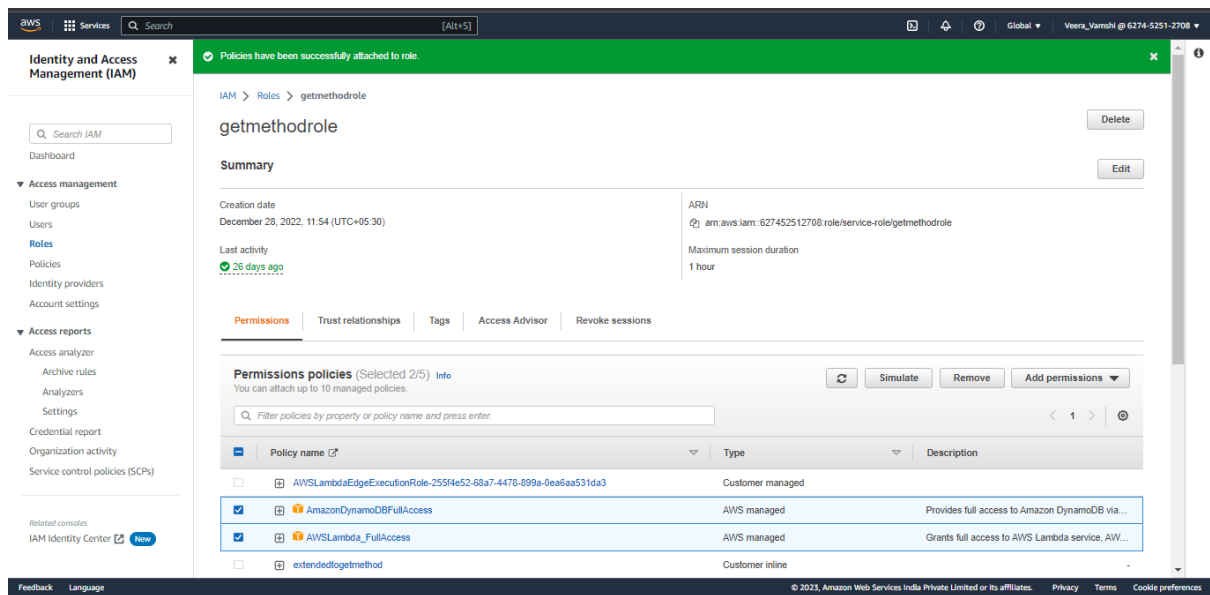# CSC PRACTICAL – 08

Name: K.Nitin reddy

ID no: 2000030510

Sec: 13

Create role:

Create and Integrate Lambda to API:

Code:

```
use strict' const AWS = require('aws-sdk');

 AWS.config.update({ region: "us-east-1"});

 exports.handler = async(event, context) => { const db = new AWS.DynamoDB(

{ apiVersion: "2012-10-08"});

const documentClient = new AWS.DynamoDB.DocumentClient({ region: "us-east1"});

const params = { TableName: "Student", Key: { id: "9876" } }

try { const data = await documentClient.get(params).promise();

 console.log(data);

}

catch(err) {console.log(err); } }
```
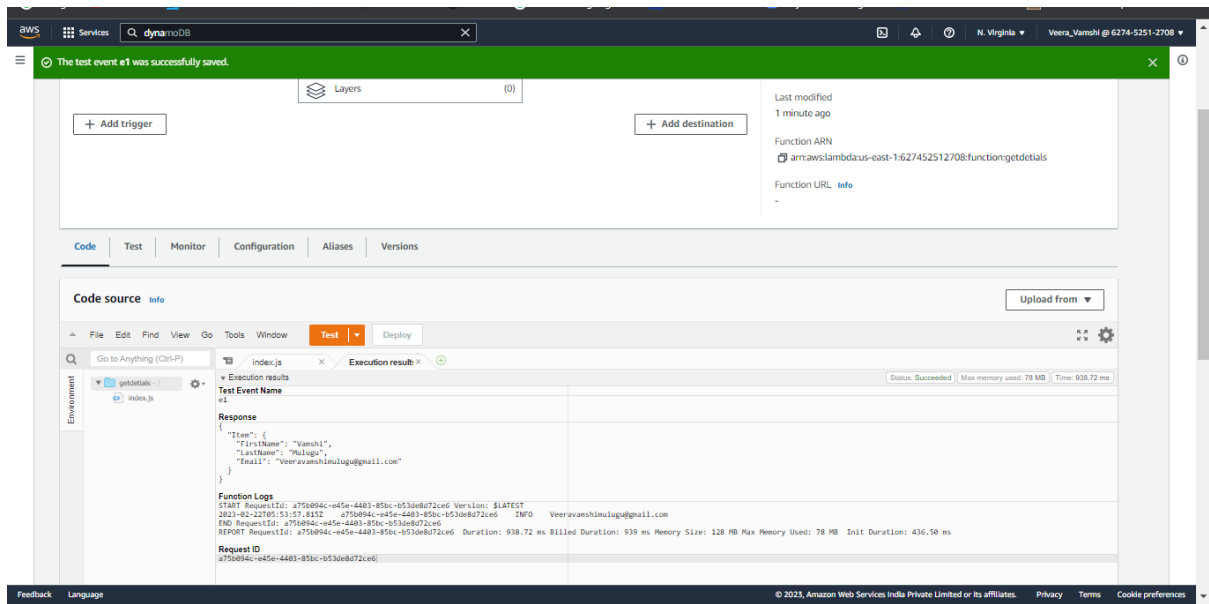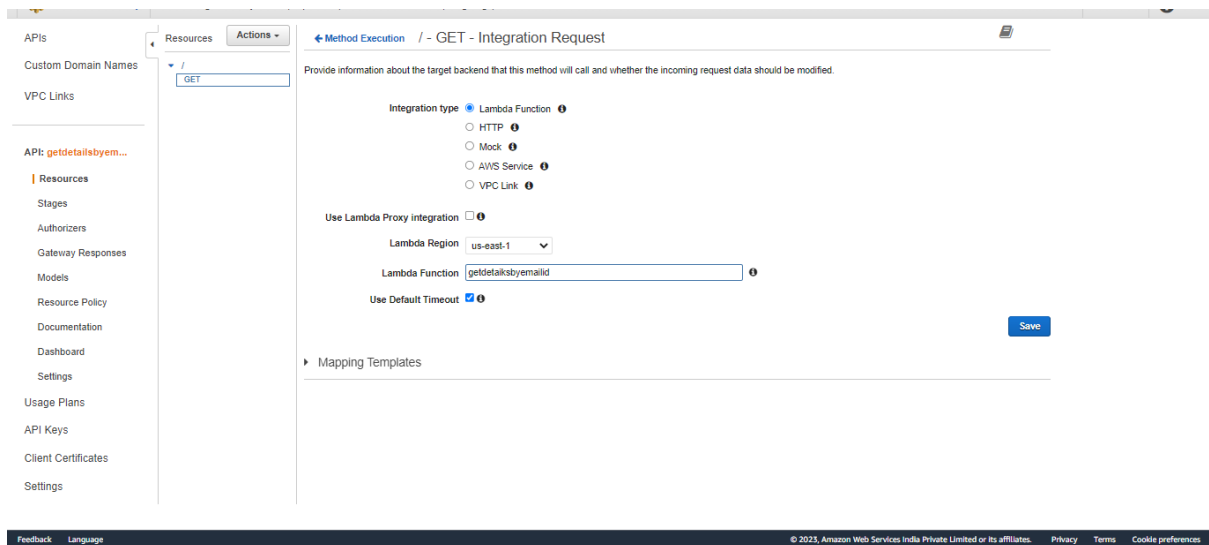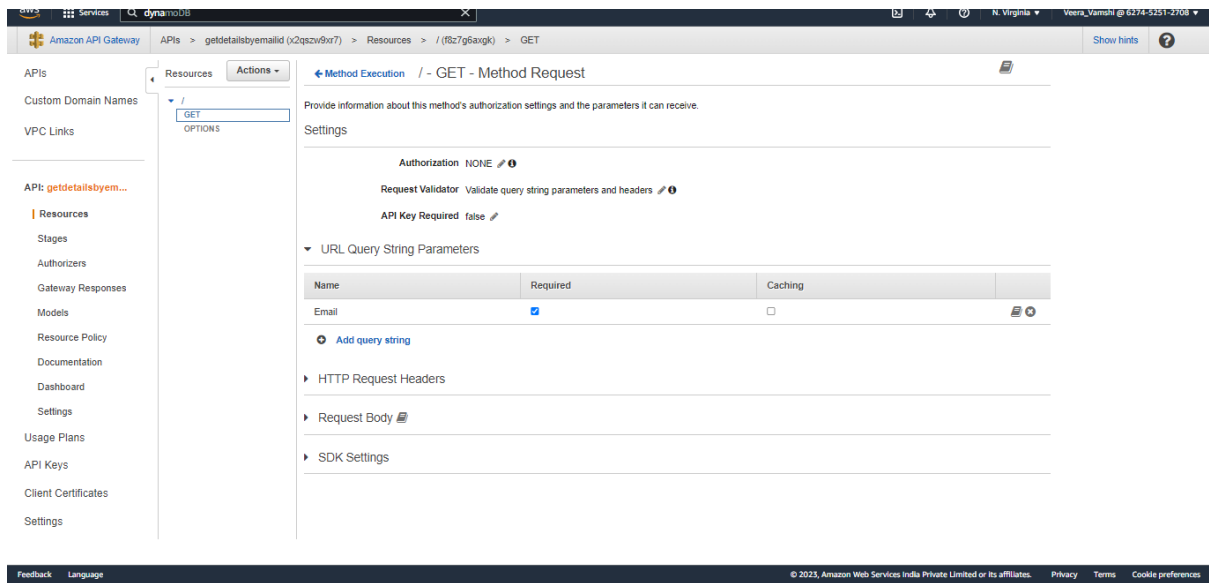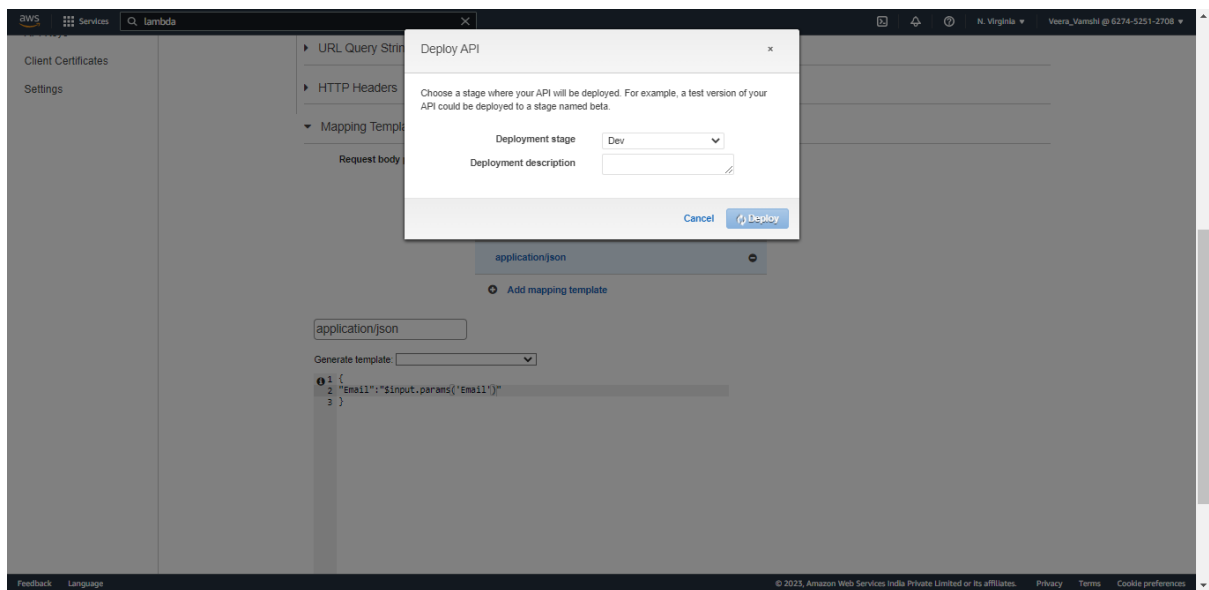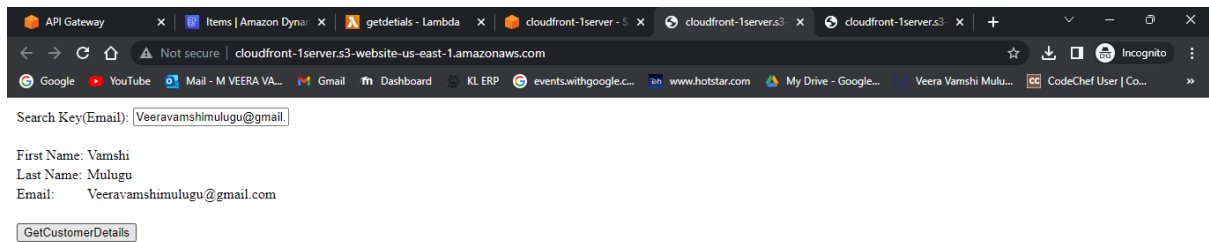
Test it:

Integrate with API Gateway:

Similarly, deploy API:

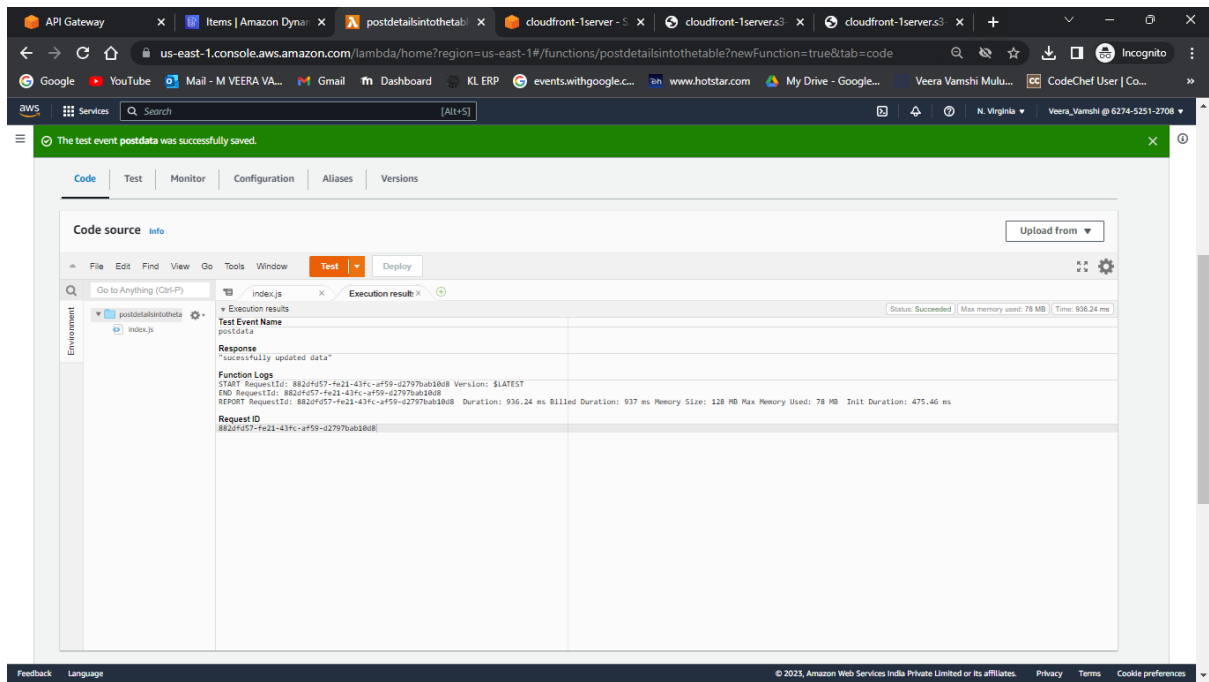**Adding files into the s3 and hosting it and data is being retrieved from the database based on email:**

Search Key(Email): Veeravamshimulugu@gmail.

First Name: Vamshi
Last Name: Mulugu
Email: Veeravamshimulugu@gmail.com

GetCustomerDetails

Now for Posting data into the dynamo DB:

create a lambda fn and write the code as follows:


```javascript
var docClient = new AWS.DynamoDB.DocumentClient();

exports.handler = (event, context, callback) => {
    var tableName = "CustomerDetails";

    var params = {
        TableName : tableName,
        Item:{
            "Email" : event.Email,
            "FirstName" : event.FirstName,
            "LastName" : event.LastName,
            }
};
docClient.put(params, function(err,data){
    if(err){
        callback(err);
    }
    else{
        callback(null,"sucessfully updated data");
    }
})
};
```
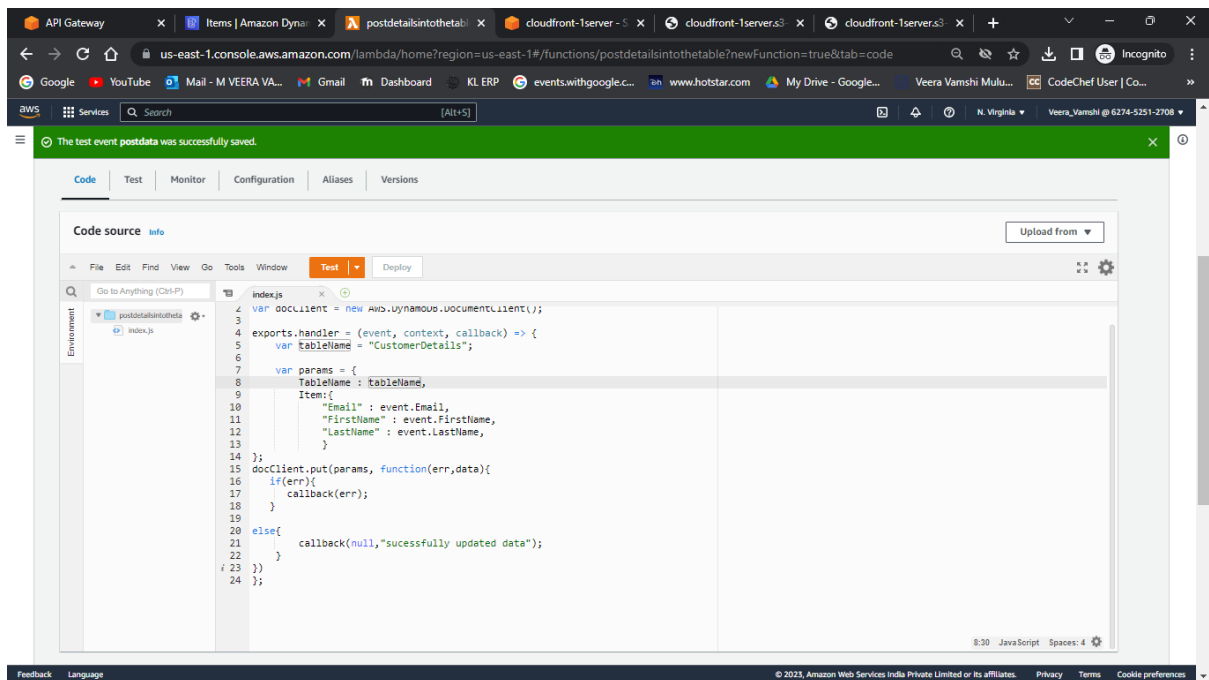
Tested:

Output:
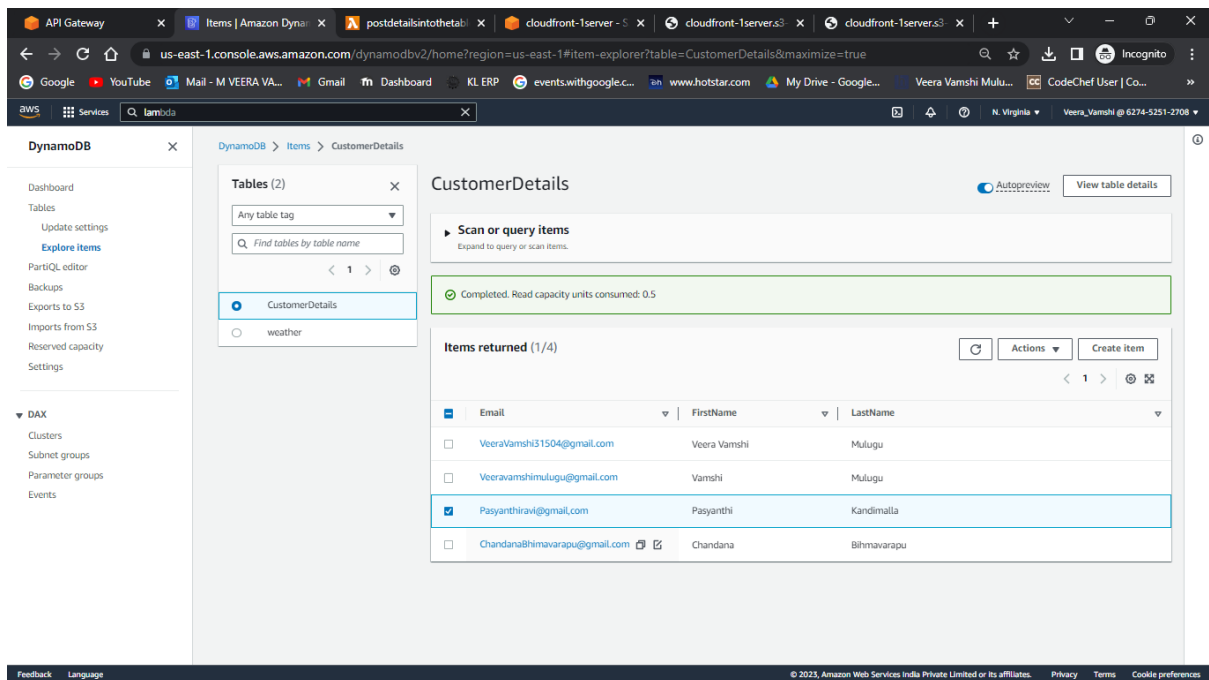
Integrate lambda with post method:
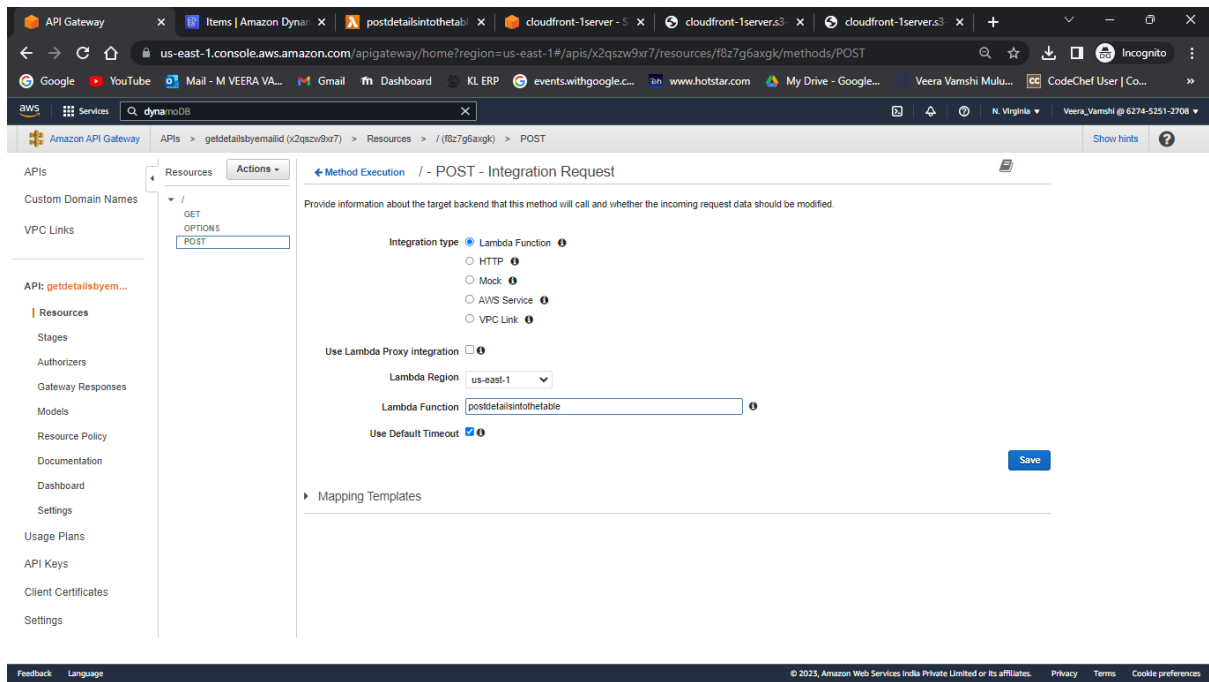
Enable CORS and deploy it and test on postman:
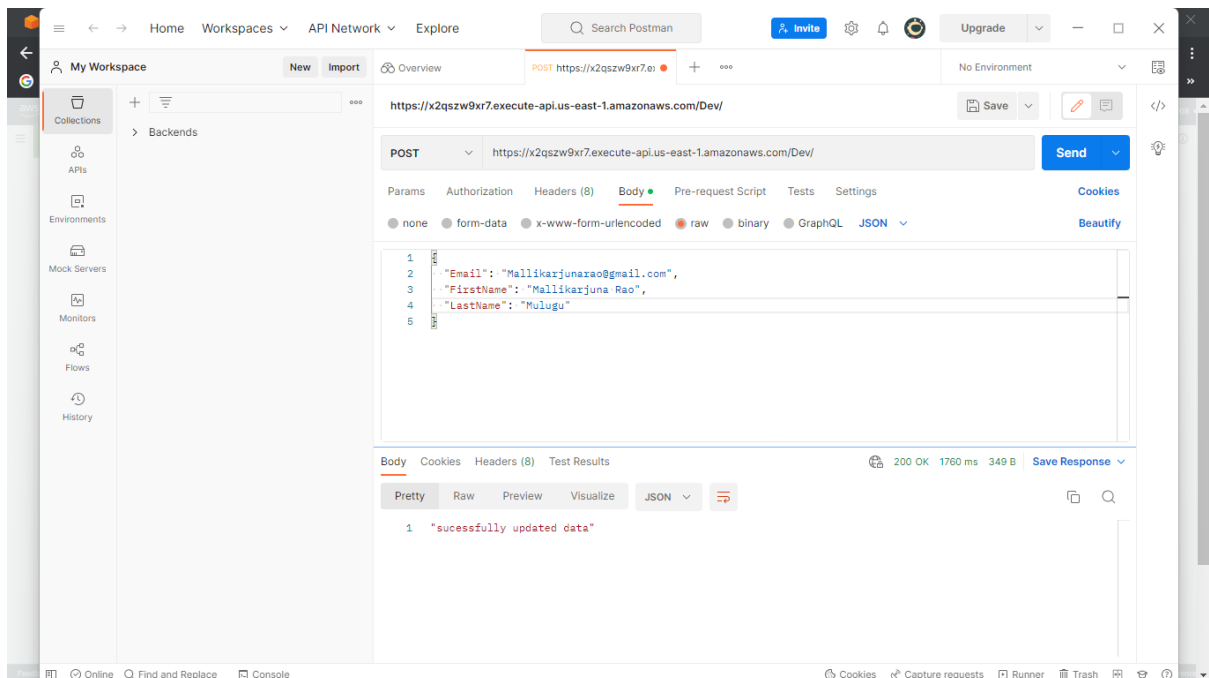
Data is added into the database:

Hosting a website: Create a bucket with publicly accessible and add the following files:

Hosting and sending data:

Email: Bharathi@gmail.com

FirstName: Bharathi

LastName: Mulugu

Save

Email:

FirstName:

LastName:

sucessfully updated data

Save

Output: added into the database: