Assignment-4

Name: Uma Lakshmi Sai Gude

Roll No: 20NN1A1218

E-mail: umalakshmisai24@gmail.com

Ph no: 9391448905

College: Vignan’s Nirula Institute of Science and Technology for Women.

Index.js

```
express=require('express')
const mongoose=require('mongoose');
const Product=require('./models/product.model.js');
const app = express()

app.use(express.json());




//reading all products
app.get('/',function(req,res){
         res.send("hello from the node apiupdate");
     });

     app.get('/api/products',async(req,res)=>{

      try{
         const products=await Product.find({});
         res.status(200).json(products);

         }catch(error){
            res.status(500).json({message:error.message});

         }
     });

      //read api but by only one product
     app.get('/api/product/:id',async(req,res)=>{ try{
            const{id}=req.params;
            const product=await Product.findById(id);
            res.status(200).json( product );

      }catch(error){
         res.status(500).json({message:error.message});
      }
     });


     //create api
     app.post('/api/products',async(req,res)=>{
     try{
        const product=await Product.create(req.body);
        res.status(200).json(product);
     }catch(error){
      res.status(500).json({message:error.message});
     }
     });



         //update a product
         app.put('/api/product/:id',async(req,res)=>{ try {
               const{id}=req.params;

                const product=await Product.findByIdAndUpdate(id,
req.body);
                if(!product){
                  return res.status(404).json({message:"Product not found"});
                }

                 Const updated Product=await Product.findById(id);
                  res.status(200).json(updated Product);



            }catch(error){
                res.status(500).json({message:error.message});
            }
         });


         //delete a product

         app.delete("/api/product/:id",async(req,res)=>{
            try{
               const{id}=req.params;
              const product=await Product.findByIdAndDelete(id); if
              (!product){
               return res.status(404).json({message:"Product not found"});
              }

              res.status(200).json({message:"Product deleted successfully"});


            }catch(error){
                res.status(500).json({message:error.message});
            }
         })
     //here first i connected db and then listened to the port

     mongoose.connect("mongodb+srv://akashvaddi333:K5m18vy6fB6aU7K2@cluster0.h
p9gamr.mongodb.net/Node-API?retryWrites=true&w=majority&appName=Cluster0")
  .then(()=>{console.log('Connected!'); app.listen(3000,
  () =>{
    console.log('server is running on port 3000')
 });

});
```

Package.Json

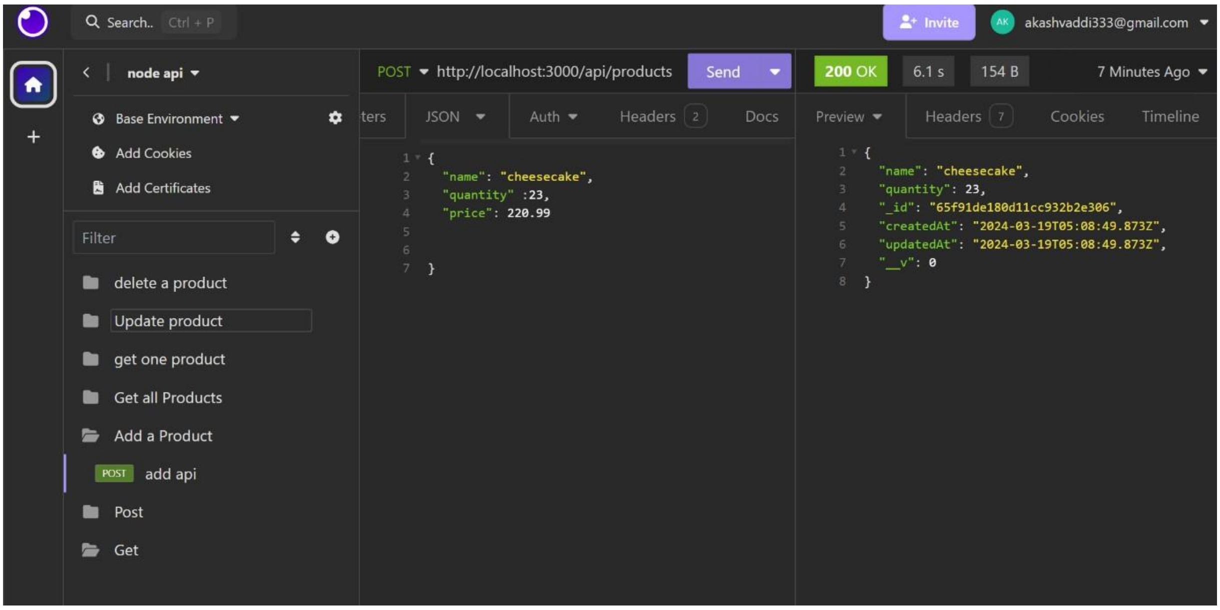
```
{
  "name":"aka-qpi",
  "version":"1.0.0",
  "description":"",
  "main":"index.js",
  "scripts": {
    "test":"echo\"Error:notes t specified\"&&exit1", "serve":
    "node index.js",
    "dev":"nodemonindex.js"
  },
  "keywords":[],
  "author":"",
  "license":"ISC",
  "dependencies":{
    "express":"^4.18.3",
    "mongodb":"^6.5.0",
    "mongoose":"^8.2.2"
  },
  "devDependencies": {
    "nodemon":"^3.1.0"
  }
}
```

Product.model.js

```
Const mongoose=require('mongoose');


Const ProductSchema=mongoose.Schema(
    {
        name:{
                type:String,
                required:[true,"proto"],


        },
        quantity:{
            type:Number,
            required:true,
            default:0
        },

        image:{
            type:String,
            required:false
        },


    },

    {
            timestamps:true,
    }
);



const Product=mongoose.model("Product",ProductSchema);
module.exports= Product;
```

- ***CRUD operations***

Create api:

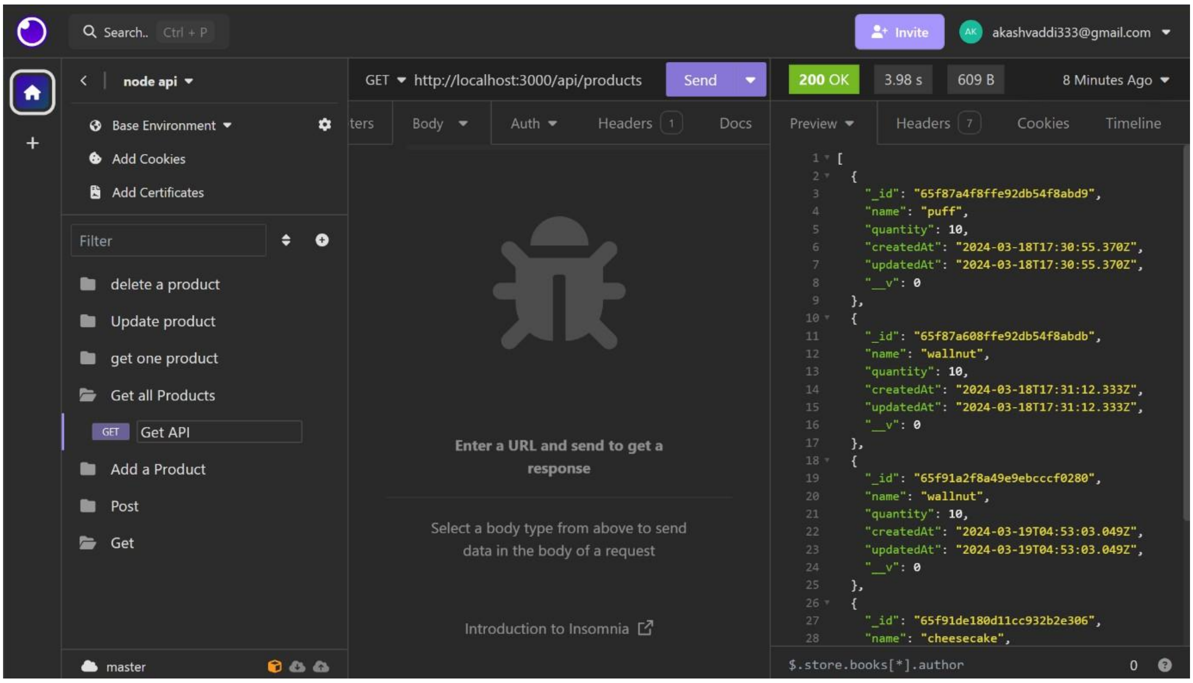


POST http://localhost:3000/api/products

```
{
  "name": "cheesecake",
  "quantity" :23,
  "price": 220.99


}
```

```
{
  "name": "cheesecake",
  "quantity": 23,
  "_id": "65f91de180d11cc932b2e306",
  "createdAt": "2024-03-19T05:08:49.873Z",
  "updatedAt": "2024-03-19T05:08:49.873Z",
  "__v": 0
}
```

Read Api:



GET http://localhost:3000/api/products

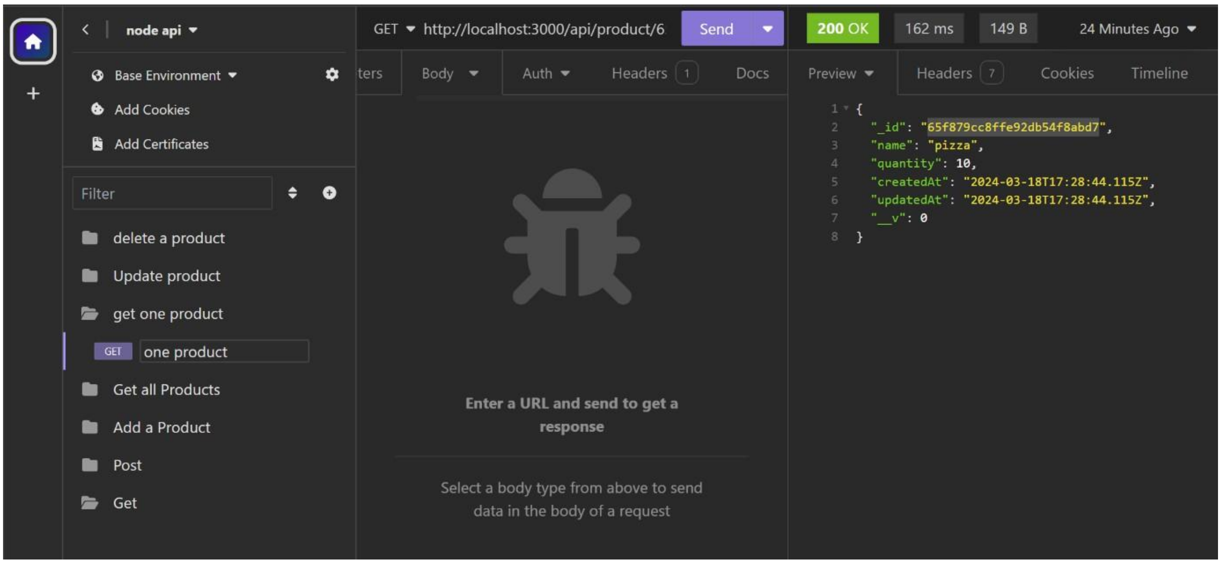
Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

```
[
  {
    "_id": "65f87a4f8ffe92db54f8abd9",
    "name": "puff",
    "quantity": 10,
    "createdAt": "2024-03-18T17:30:55.370Z",
    "updatedAt": "2024-03-18T17:30:55.370Z",
    "__v": 0
  },
  {
    "_id": "65f87a608ffe92db54f8abdb",
    "name": "wallnut",
    "quantity": 10,
    "createdAt": "2024-03-18T17:31:12.333Z",
    "updatedAt": "2024-03-18T17:31:12.333Z",
    "__v": 0
  },
  {
    "_id": "65f91a2f8a49e9ebcccf0280",
    "name": "wallnut",
    "quantity": 10,
    "createdAt": "2024-03-19T04:53:03.049Z",
    "updatedAt": "2024-03-19T04:53:03.049Z",
    "__v": 0
  },
  {
    "_id": "65f91de180d11cc932b2e306",
    "name": "cheesecake",
```

$.store.books[*].author 0

Readone Api:



GET http://localhost:3000/api/product/6

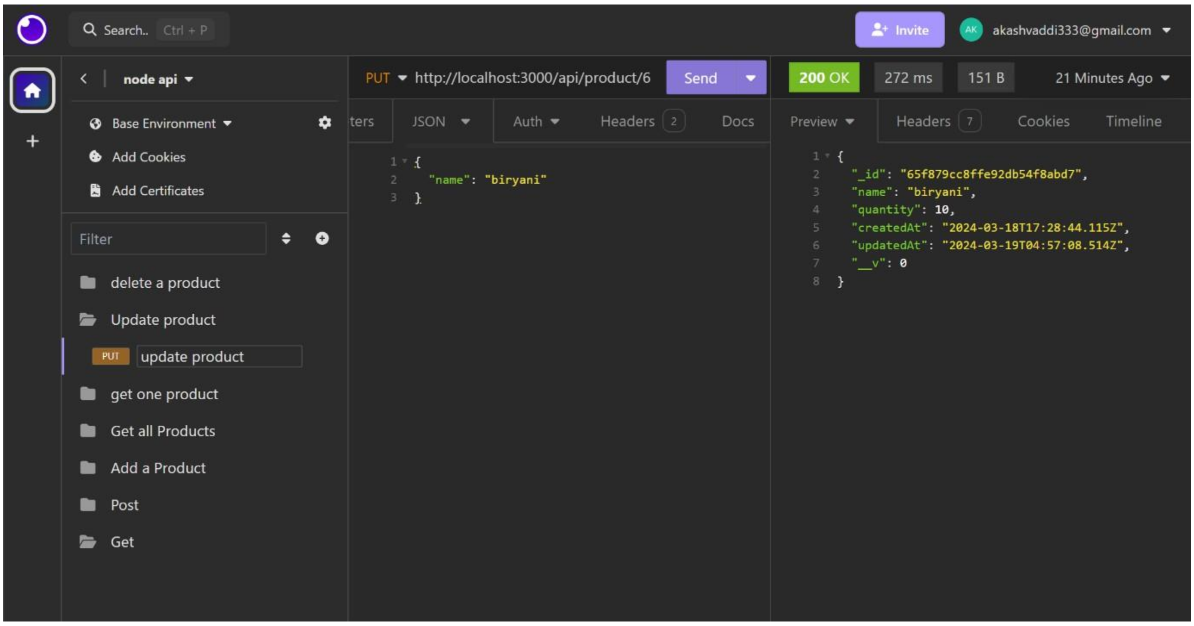
200 OK 162 ms 149 B 24 Minutes Ago

```
{
  "_id": "65f879cc8ffe92db54f8abd7",
  "name": "pizza",
  "quantity": 10,
  "createdAt": "2024-03-18T17:28:44.115Z",
  "updatedAt": "2024-03-18T17:28:44.115Z",
  "__v": 0
}
```

Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

Update Api:



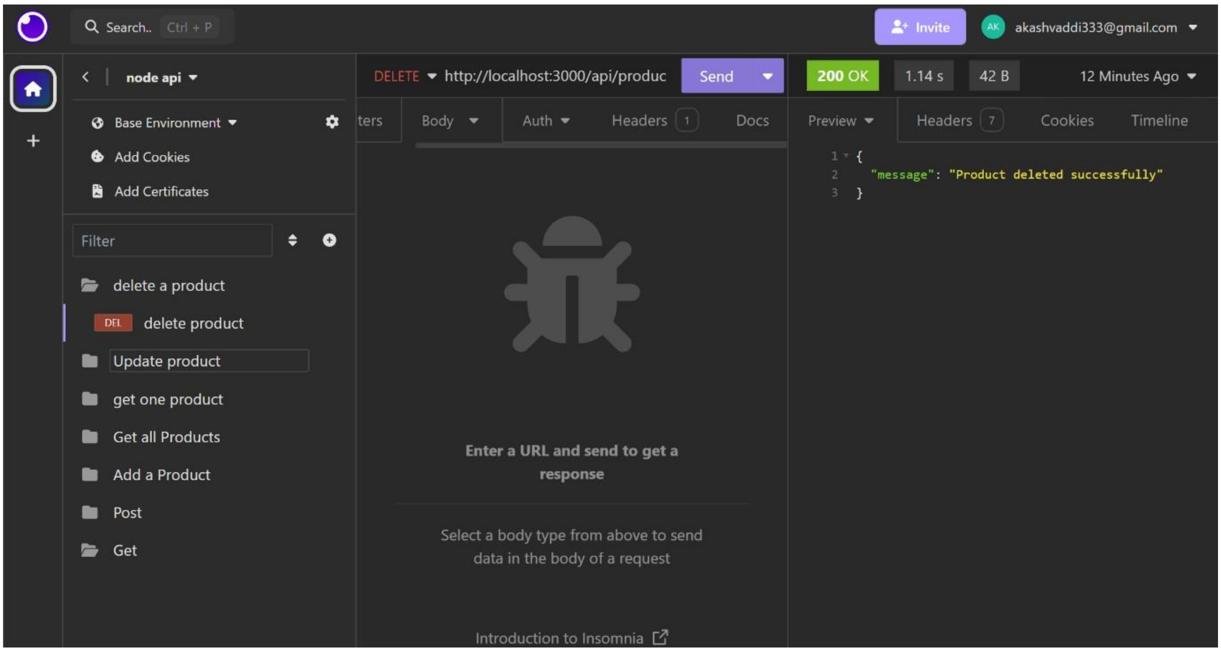
PUT http://localhost:3000/api/product/6

```
{
  "name": "biryani"
}
```

200 OK 272 ms 151 B 21 Minutes Ago

```
{
  "_id": "65f879cc8ffe92db54f8abd7",
  "name": "biryani",
  "quantity": 10,
  "createdAt": "2024-03-18T17:28:44.115Z",
  "updatedAt": "2024-03-19T04:57:08.514Z",
  "__v": 0
}
```

Delete Api:



DELETE http://localhost:3000/api/produc Send

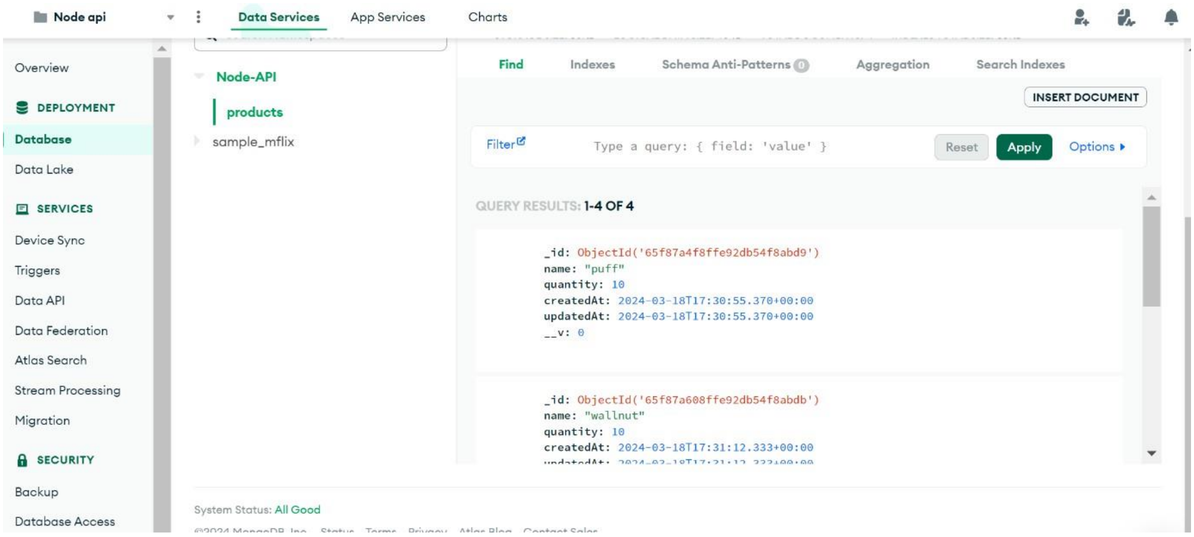
200 OK 1.14 s 42 B 12 Minutes Ago

```
{
  "message": "Product deleted successfully"
}
```

Enter a URL and send to get a response

Select a body type from above to send data in the body of a request

MongoDB: (final view)



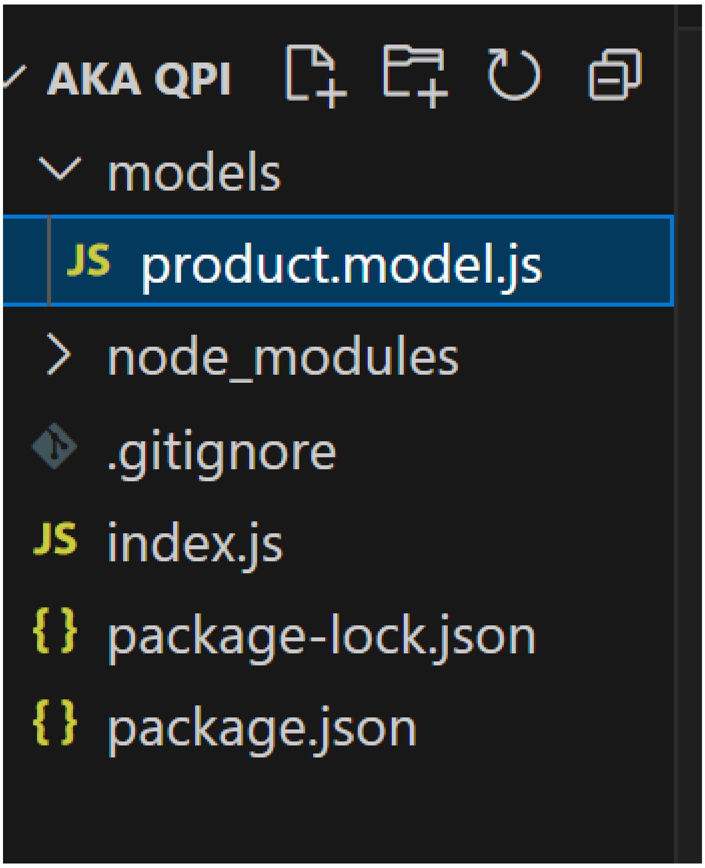
Type a query: { field: 'value' }

QUERY RESULTS: 1-4 OF 4

```
_id: ObjectId('65f87a4f8ffe92db54f8abd9')
name: "puff"
quantity: 10
createdAt: 2024-03-18T17:30:55.370+00:00
updatedAt: 2024-03-18T17:30:55.370+00:00
__v: 0
```

```
_id: ObjectId('65f87a608ffe92db54f8abdb')
name: "wallnut"
quantity: 10
createdAt: 2024-03-18T17:31:12.333+00:00
```

AKAQPI:



AKA QPI

models

product.model.js

node_modules

.gitignore

index.js

package-lock.json

package.json