

React + NODE JS(express js) CURD Operation

	note
<pre>const express = require('express'); // express মডিউল আমদানি করা হলো const cors = require('cors'); // ক্রস-অরিজিন রিসোর্স শেয়ারিং (CORS) সক্ষম করে, যা ব্রাউজারকে একটি ডোমেইন থেকে অন্য ডোমেইনের রিসোর্স অ্যাক্সেস করতে দেয়। const jwt = require('jsonwebtoken'); // jsonwebtoken মডিউল আমদানি করা হলো, যা JSON ওয়েব টোকেন ব্যবহারের জন্য const cookieParser = require('cookie-parser'); // cookie-parser মডিউল আমদানি করা হলো, যা কুকি হ্যান্ডলিং এর জন্য const { MongoClient, ServerApiVersion, ObjectId } = require('mongodb'); // MongoDB মডিউল এবং ক্লাসগুলি আমদানি করা হলো require('dotenv').config(); // dotenv মডিউল আমদানি করা হলো, যা .env ফাইল থেকে পরিবেশ ভেরিয়েবল লোড করে const app = express(); // express অ্যাপ্লিকেশন তৈরি করা হলো const port = process.env.PORT 5000; // পরিবেশ ভেরিয়েবল থেকে পোর্ট নম্বর সেট করা হলো, যদি না পাওয়া যায় তাহলে ৫০০০ সেট করা হবে // middleware app.use(cors({ origin: ["http://localhost:5173", // স্থানীয় ডেভেলপমেন্ট সার্ভারের জন্য অনুমোদিত "https://b9a11-client-side.web.app", // হোস্টেড ক্লায়েন্ট সাইটের জন্য অনুমোদিত "https://b9a11-client-side.firebaseio.com" // ফায়ারবেস অ্যাপের জন্য অনুমোদিত], credentials: true, // ক্রস-অরিজিন অনুরোধগুলিতে কুকি এবং প্রমাণপত্র অন্তর্ভুক্ত করতে সক্ষম করে })); app.use(express.json()); // এক্সপ্রেস অ্যাপ্লিকেশনকে JSON ফর্ম্যাটে ডেটা পার্স করতে সক্ষম করে। এইটা ছাড়া data font end থেকে backend এ যায় না। app.use(cookieParser()); // কুকি পার্সিং সক্ষম করা হলো</pre>	Basic setup of Express js

	<pre>// একটি রুট ডিফাইন করা হল, যা হোম পেজে অনুরোধ করলে একটি 'Hello World!' মেসেজ পাঠাবে</pre>	
--	--	--

```
app.get('/', (req, res) => {  
  res.send('Hello World!');  
});
```

```
// সার্ভারটি ৩০০০ পোর্টে শোনা শুরু করবে
```

```
app.listen(3000, () => {  
  console.log('Server is running on http://localhost:3000');  
});
```

ADD/POST

<pre>const handleAdd = (e) => { e.preventDefault(); const form = e.target;</pre> <p>Note: সব ফিল্ড থেকে ডাটাগুলো রিসিভ করা হচ্ছে</p> <pre>const productName = form.productName.value; const productBrand = form.productBrand.value; const productImageURL = form.productImageURL.value; const queryTitle = form.queryTitle.value; const boycottingReasonDetails = form.boycottingReasonDetails.value; const userEmail = user.email; const name = user.displayName; const image = user.photoURL; const dateTime = formattedDate; const recommendationCount = 0;</pre> <p>ডাটা রেডি সার্ভারের কাছে পাঠানোর জন্য</p> <pre>const addQueries = { productName, productBrand, productImageURL, queryTitle, boycottingReasonDetails, userEmail, name, image,</pre>	<pre>async function run() { try { const db = client.db("quickThinkDB"); // "quickThinkDB" ডাটাবেসের সাথে সংযোগ স্থাপন করা হলো // দুইটি সংগ্রহ (collection) তৈরি করা হলো const addQueriesCollection = db.collection("addQueriesCollection"); const recommendationCollection = db.collection("recommendationCollection");</pre>	
---	--	--

```
    dateTime,  
    recommendationCount  
  }  
}
```

```
fetch("https://b9a11-server-side-sakhawat-  
code-info.vercel.app/addQueryData", {  
  method: "POST", // POST অনুরোধের  
  মাধ্যম  
  headers: {  
    "Content-Type": "application/json" //  
    অনুরোধের হেডার যা কনটেন্ট টাইপ JSON  
    হিসাবে সেট করে  
  },  
  body: JSON.stringify(addQueries), //  
  অনুরোধের শরীর যা JSON ফরম্যাটে প্রেরণ  
  করা হচ্ছে  
})  
  .then(res => res.json()) // প্রতিক্রিয়াটি  
  JSON ফরম্যাটে রূপান্তরিত করা হচ্ছে  
  .then(data => {  
    if (data.insertedId) { // যদি ডেটায়  
    insertedId থাকে, যা নির্দেশ করে ডেটা  
    সফলভাবে সংরক্ষিত হয়েছে  
      do something alert or .....  
    }  
  
    form.reset(); // ফর্ম রিসেট করা হবে  
  });
```

উপরে নিচে basic setup

```
// post query data  
app.post('/addQueryData', async (req, res) => {  
  const receiveQueryData = req.body; //  
  অনুরোধের শরীর থেকে ডেটা প্রাপ্ত করা হচ্ছে  
  const result = await  
  addQueriesCollection.insertOne(receiveQueryData);  
  // ডেটা সংগ্রহে একত্রিত করা হচ্ছে  
  res.send(result); // ফলাফল ক্লায়েন্টকে পাঠানো  
  হচ্ছে  
})
```

```
console.log("Pinged your deployment. You  
successfully connected to MongoDB!");  
} finally {  
  // নিশ্চিত করে যে ক্লায়েন্ট বন্ধ হবে যখন আপনার  
  কাজ শেষ হবে বা ত্রুটি ঘটবে  
}  
}  
run().catch(console.dir); // ফাংশনটি চালানো হচ্ছে  
এবং ত
```

<p>Note: ei <code>handleAdd</code> দিয়ে উপরে ফাংশনে ডাটা সেভ করতে হবে।</p> <pre> <form onSubmit={handleAdd}> <input type="text" className="shadow-sm bg-gray-50 border border-gray-300 text-gray-900 sm:text-sm rounded-lg focus:ring-cyan-600 focus:border-cyan-600 block w-full p-2.5" placeholder="Apple 27" name="productName" required /> */} </form> </pre>		
<p>Individual data get or find korar upay</p>		
	<pre> // send myQueries Data app.get('/myQueries/:byEmail', async (req, res) => { const emailreceive = req.params.byEmail; </pre>	<p>userEmail এইটা database এ আছে।</p>

const { user } = UseAuth(); // UseAuth হুক থেকে ব্যবহারকারীর তথ্য প্রাপ্ত করা হচ্ছে
const [myQueriesData, setMyQueriesData] = useState([]); // myQueriesData রাজ্য ঘোষণা করা হচ্ছে এবং সেট করার জন্য setMyQueriesData ফাংশন তৈরি করা হচ্ছে

```
useEffect(() => {  
  fetch('https://b9a11-server-side-sakhawat-code-info.vercel.app/myQueries/${user.email}')  
  // ব্যবহারকারীর ইমেইল দিয়ে ডেটা ফেচ করা হচ্ছে  
  .then(res => res.json()) // প্রতিক্রিয়াটি JSON ফরম্যাটে রূপান্তরিত করা হচ্ছে  
  .then(data => {  
    const sortByDate =  
    [].concat(data).sort((a, b) => b.dateTime - a.dateTime); // ডেটাগুলি সময় অনুযায়ী সাজানো হচ্ছে  
    setMyQueriesData(sortByDate); // সাজানো ডেটাগুলি myQueriesData রাজ্যে সংরক্ষণ করা হচ্ছে  
  });  
}, [user.email]); // user.email পরিবর্তন হলে useEffect হুকটি পুনরায় চালু হবে
```

```
const cursor = addQueriesCollection.find({ userEmail: emailreceive }); // addQueriesCollection থেকে ব্যবহারকারীর ইমেইল অনুযায়ী ডেটা নামিয়ে আনতে কিউসর তৈরি করা হলো  
const result = await cursor.toArray(); // কিউসর থেকে ডেটাগুলি অ্যারেতে রূপান্তর করা হলো  
res.send(result); // ফলাফল ক্লায়েন্টকে পাঠানো হলো  
});
```

→ Data base থেকে single data but single user এর All data Get করে নিয়ে আসবো

myQueriesData.map

<p>Search field data get or find korar upay</p>		
<p>const allQueryData = useLoaderData(); // লোডার ডেটা ব্যবহার করে সব প্রশ্নের ডেটা প্রাপ্ত করা হচ্ছে</p> <p>const [allQueries, setAllQueries] = useState([]); // allQueries রাজ্য ঘোষণা করা হচ্ছে এবং সেট করার জন্য setAllQueries ফাংশন তৈরি করা হচ্ছে</p> <p>useEffect(() => { setAllQueries(allQueryData); // allQueryData রাজ্যকে allQueries রাজ্যে সংরক্ষণ করা হচ্ছে }, [allQueryData]); // allQueryData পরিবর্তিত হলে useEffect হুকটি পুনরায় চালু হবে</p> <p>const handleSearchField = (e) => { e.preventDefault(); // ফর্মের ডিফল্ট সাবমিট বিহেভিয়ার বন্ধ করা হচ্ছে const form = e.target; const searchData = form.searchData.value; // সার্চ ফিল্ডের ভ্যালু প্রাপ্ত করা হচ্ছে searchDataFetch(searchData); // সার্চ ডেটা ফেচ ফাংশন কল করা হচ্ছে };</p> <p>const searchDataFetch = async (searchData) => { const data = await fetch(`https://b9a11-server-side-sakhawat-code-info.vercel.app/allQueryDataBy?productName=\${searchData}`); // সার্ভার থেকে সার্চ ডেটা ফেচ করা হচ্ছে</p>	<pre>// send search Data app.get('/allQueryDataBy', async (req, res) => { const querie = req.query.productName; const cursor = addQueriesCollection.find({ productName: { \$regex: querie, \$options: 'i' } }).sort({ \$natural: -1 }); const result = await cursor.toArray(); res.send(result); })</pre>	

<pre>const res = await data.json(); // প্রতিক্রিয়াটি JSON ফরম্যাটে রূপান্তরিত করা হচ্ছে setAllQueries(res); // প্রাপ্ত ডেটা allQueries রাজ্যে সংরক্ষণ করা হচ্ছে };</pre>		
---	--	--