# Ajax Fundamentals - With Project

## Lets run by building a project

### CRUD:

**Create**
**Read**
**Update**
**Delete**

Download **milligram.css**
Download **axios.min.js**

### File structure:

```
|-css
  |-milligram.css
|-js
  |-axios.min.js
|-create.html
|-index.html
|-update.html
```

1. **index.html** - মূল HTML ফাইল, যেখানে টেবিল তৈরি হবে।
2. **milligram.css** - সাধারণ ডিজাইনের জন্য CSS ফাইল।
3. **axios.min.js** - HTTP রিকোয়েস্ট পাঠানোর জন্য লাইব্রেরি।

### APIs:

https://crud.teamrabbil.com/api/v1/ReadProduct
https://crud.teamrabbil.com/api/v1/DeleteProduct/${id}
https://crud.teamrabbil.com/api/v1/CreateProduct
https://crud.teamrabbil.com/api/v1/ReadProductByID/${id}
https://crud.teamrabbil.com/api/v1/UpdateProduct/${productID}

**index.html** setup:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

```
    <link rel="stylesheet" href="./css/milligram.css" />
    <title>Axios</title>
  </head>

  <body>

    <script src="./js/axios.min.js"></script>
  </body>
</html>
```

---

### CRUD : Read

https://crud.teamrabbil.com/api/v1/ReadProduct

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="./css/milligram.css" />
    <title>Axios</title>
  </head>

  <body>
    <div class="container">
      <div class="row">
        <table>
          <thead>
            <tr>
              <th>Product Name</th>
              <th>Product Code</th>
              <th>Unit Price</th>
              <th>Quantity</th>
              <th>Total Price</th>
              <th>Delete</th>
              <th>Update</th>
            </tr>
          </thead>
          <tbody id="productList"></tbody>
        </table>
      </div>
    </div>


    <script src="./js/axios.min.js"></script>
```

```
    <script>

   // Read Operation:

      async function getProducts() {
        const URL = "https://crud.teamrabbil.com/api/v1/ReadProduct"; //
picking the URL first from the API

        const response = await axios.get(URL); // storing the response in a
variable, we are calling the api url, with axios, so we wrote axios.get()
function. For this all the data will be stored in response variable in
JavaScript. You can check the response, in the console.

        const data = response.data["data"]; // All the products are stored
in 'data' object. So first we have to enter to the 'data' object, then we
will get all the product list from the 'data' object. By storing the
products data in data variable, we will get all the data from the response.

        // now we will send these data to the tbody element with loop, using
forEach.
        data.forEach((product) => {
          document.getElementById("productList").innerHTML += `
            <tr>
              <td>${product["ProductName"]}</td>
              <td>${product["ProductCode"]}</td>
              <td>${product["UnitPrice"]}</td>
              <td>${product["Qty"]}</td>
              <td>${product["TotalPrice"]}</td>
              <td><button>Delete</button></td>
              <td><button>Update</button></td>
            </tr>
          `;
        }); // with this we will get the items one by one and store them in
'product' variable.
      }

      getProducts();
    </script>
  </body>
</html>
```

## Explanation:

এই কোডটি API থেকে ডেটা নিয়ে টেবিলে দেখাবে।

### Axios ব্যবহার করা

```
// পণ্য লোড করার জন্য ফাংশন
async function getProducts() {
```

```
    const URL = "https://crud.teamrabbil.com/api/v1/ReadProduct"; // API ঠিকানা
    const response = await axios.get(URL); // API থেকে ডেটা নিয়ে আসা
    const data = response.data["data"]; // ডেটার ভিতর "data" নামে একটি অংশ আছে, যেখানে পণ্যের তালিকা আছে।

    // প্রতিটি পণ্যের জন্য টেবিলের একটি করে সারি যোগ করা
    data.forEach((product) => {
document.getElementById("productList").innerHTML += `
        <tr> <td>${product["ProductName"]}</td>
        <td>${product["ProductCode"]}</td>
        <td>${product["UnitPrice"]}</td>
        <td>${product["Qty"]}</td>
        <td>${product["TotalPrice"]}</td>
        <td><button>ডিলিট</button></td>
        <td><button>আপডেট</button></td>
        </tr>
        `;
        });
    } // ফাংশনটি কল করা, যাতে পণ্যগুলো লোড হয় getProducts();
```

**ধাপে ধাপে ব্যাখ্যা :**

**HTML টেবিল**

- `<table>` : এখানে পণ্যের তথ্য দেখানোর জন্য একটি টেবিল তৈরি করা হয়েছে।
- `<tbody id="productList">` : টেবিলের এই অংশে আমরা API থেকে পাওয়া ডেটা দেখাবো।

**API থেকে ডেটা আনা**

- `axios.get()` ফাংশনটি দিয়ে API-তে রিকোয়েস্ট পাঠানো হয়।
- API থেকে ডেটা রেসপন্স হিসেবে আসে, যা আমরা `response.data["data"]` থেকে পাই।

**ডেটা দেখানো**

- `forEach()` : এই ফাংশন দিয়ে প্রতিটি পণ্যের জন্য একটি করে টেবিলের সারি ( `<tr>` ) তৈরি করা হয়।
- `innerHTML` : টেবিলের মধ্যে নতুন সারি যোগ করার জন্য এটি ব্যবহার করা হয়েছে।

**Button**

- প্রতিটি সারিতে **ডিলিট** ও **আপডেট** বোতাম রাখা হয়েছে, যা পরবর্তীতে কাজ করবে।

**API সম্পর্কে**

- **ঠিকানা**: `https://crud.teamrabbil.com/api/v1/ReadProduct`
- **মেথড**: GET
- **ফলাফল**: JSON ফরম্যাটে পণ্যের তথ্য।

## CRUD : Delete

https://crud.teamrabbil.com/api/v1/DeleteProduct/${id}

```
<!DOCTYPE html>
<html lang="en">

  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="./css/milligram.css" />
    <title>Axios</title>
  </head>

  <body>

    <div class="container">
      <div class="row">
        <table>
          <thead>
            <tr>
              <th>Product Name</th>
              <th>Product Code</th>
              <th>Unit Price</th>
              <th>Quantity</th>
              <th>Total Price</th>
              <th>Delete</th>
              <th>Update</th>
            </tr>
          </thead>

          <tbody id="productList"></tbody>
        </table>
      </div>
    </div>

    <script src="./js/axios.min.js"></script>

    <script>

      // Read
      async function getProducts() {

        const URL = "https://crud.teamrabbil.com/api/v1/ReadProduct"; //
picking the URL first from the API

        const response = await axios.get(URL); // storing the response in a
```

```
variable, we are calling the api url, with axios, so we wrote axios.get()
function. For this all the data will be stored in response variable in
JavaScript. You can check the response, in the console.

        const data = response.data["data"]; // All the products are stored
in 'data' object. So first we have to enter to the 'data' object, then we
will get all the product list from the 'data' object. By storing the
products data in data variable, we will get all the data from the response.

        // now we will send these data to the tbody element with loop, using
forEach.

        data.forEach((product) => {

          document.getElementById("productList").innerHTML += `

            <tr>
              <td>${product["ProductName"]}</td>
              <td>${product["ProductCode"]}</td>
              <td>${product["UnitPrice"]}</td>
              <td>${product["Qty"]}</td>
              <td>${product["TotalPrice"]}</td>
              <td><button
onclick="deleteProduct('${product["_id"]}')">Delete</button></td>
              <td><button>Update</button></td>
            </tr>
          `;
        }); // with this we will get the items one by one and store them in
'product' variable.
      }

      getProducts();




      // Delete

      async function deleteProduct(id) {

        const URL =
`https://crud.teamrabbil.com/api/v1/DeleteProduct/${id}`; // picking the URL
first from the API

        const response = await axios.get(URL); // storing the response in a
variable

        document.getElementById("productList").innerHTML = ""; // to remove
the previous data from the table
```

```
        await getProducts(); // to get the updated data after deleting
      }

    </script>

  </body>

</html>
```

**Delete Function:**

কোডটি একটি নির্দিষ্ট পণ্য **ডিলিট** করার জন্য লেখা হয়েছে। যখন টেবিলের `Delete` বোতামটি চাপবেন, এটি API-তে ডিলিট রিকোয়েস্ট পাঠাবে এবং সেই পণ্যটি ডিলিট হয়ে যাবে। এরপর টেবিলটি আপডেট হবে।

**Delete কোড**

```
// Delete ফাংশন
async function deleteProduct(id) {
  const URL = `https://crud.teamrabbil.com/api/v1/DeleteProduct/${id}`;
  // এখানে ডিলিট করার API-এর URL তৈরি করা হয়েছে।
  // এই URL-এ নির্দিষ্ট পণ্যের `id` যোগ করা হয়েছে, কারণ কোন পণ্যটি ডিলিট হবে তা `id` দিয়েই বোঝা যায়।

  const response = await axios.get(URL);
  // এখানে API-তে GET রিকোয়েস্ট পাঠানো হয়েছে।
  // `axios.get(URL)` রিকোয়েস্ট পাঠানোর পরে সার্ভার থেকে ডিলিট হওয়া সম্পর্কে রেসপন্স আসবে।
  // এই রেসপন্সটি আমরা `response` ভেরিয়েবলে সংরক্ষণ করেছি।

  document.getElementById("productList").innerHTML = "";
  // টেবিলের আগের সকল ডেটা মুছে ফেলা হয়েছে।
  // এটা করা হয়েছে যাতে নতুন ডেটা আপডেট করার সময় ডুপ্লিকেট না হয়।

  await getProducts();
  // পণ্যগুলোর আপডেট করা তালিকা দেখানোর জন্য `getProducts()` ফাংশনটি আবার কল করা হয়েছে।
  // এটি নতুন তালিকা টেবিলে দেখাবে, যেখানে ডিলিট হওয়া পণ্যটি থাকবে না।
}
```

**HTML ট্যাগে Delete বোতাম : adding onClick() :**

```
<td><button onclick="deleteProduct('${product["_id"]}')">Delete</button>
</td>
```

**কী হচ্ছে এখানে?**

1. `onclick="deleteProduct('${product["_id"]}')` :
   - এই লাইন বোতামে ক্লিক করার সাথে সাথে **deleteProduct()** ফাংশন কল করে।
   - `product["_id"]` -এর মাধ্যমে নির্দিষ্ট পণ্যের **ID** ফাংশনে পাঠানো হয়, যাতে API বুঝতে পারে কোন পণ্য ডিলিট করতে হবে।
2. **Delete বোতাম**:
   - প্রতিটি টেবিল রোতে (পণ্যের জন্য) একটি করে **Delete** বোতাম যোগ করা হয়েছে।
   - এই বোতামে ক্লিক করলে, সেই পণ্যের **ID** অনুযায়ী API রিকোয়েস্ট যাবে।

### কোডের কাজের ধাপ সহজ ভাষায়

1. **বোতামে ক্লিক করা**:
   - যখন আপনি `Delete` বোতামে ক্লিক করবেন, তখন **deleteProduct()** ফাংশনটি চালু হবে।
2. **API রিকোয়েস্ট পাঠানো**:
   - `deleteProduct()` ফাংশন API-তে রিকোয়েস্ট পাঠাবে, যেখানে পণ্যের **ID**-এর মাধ্যমে সার্ভারকে বলা হবে, "এই পণ্যটি ডিলিট করো।"
3. **টেবিল আপডেট করা**:
   - সার্ভার পণ্যটি ডিলিট করার পর, পুরো টেবিলটি খালি করে নতুন তালিকা দেখানো হয়।
   - `getProducts()` ফাংশনটি আবার কল করে নতুন ডেটা লোড করা হয়।

### বোঝার জন্য সহজ উদাহরণ

### ধরুন টেবিলে ৫টি পণ্য আছে:

1. `Product A`
2. `Product B`
3. `Product C`
4. `Product D`
5. `Product E`

যদি আপনি **Product C**-এর `Delete` বোতামে ক্লিক করেন:

1. **API রিকোয়েস্ট**: API-তে `Product C` -এর ID পাঠানো হবে।
2. **পণ্য ডিলিট**: সার্ভার থেকে `Product C` ডিলিট করা হবে।
3. **টেবিল আপডেট**:
   - আগের টেবিল মুছে নতুন তালিকা দেখানো হবে।
   - **নতুন তালিকা:**
     1. `Product A`
     2. `Product B`
     3. `Product D`
     4. `Product E`

### কেন আগের ডেটা মুছে নতুন ডেটা দেখানো হলো?

আগের ডেটা না মুছে নতুন ডেটা যোগ করলে ডুপ্লিকেট পণ্য দেখা যেতে পারে।

তাই প্রথমে `document.getElementById("productList").innerHTML = "";` দিয়ে টেবিল খালি করা হয়েছে।

---

### CRUD : Create

https://crud.teamrabbil.com/api/v1/CreateProduct

**create.html:**

```
<!DOCTYPE html>
<html lang="en">

  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <link rel="stylesheet" href="./css/milligram.css" />
  </head>

  <body>

    <div class="container">
      <div class="row">
        <div class="column">

          <label>Product Name</label>
          <input id="ProductName" type="text" />

          <label>Product Code</label>
          <input id="ProductCode" type="text" />

          <label>Product Image</label>
          <input id="Img" type="text" />

          <label>Unit Price</label>
          <input id="UnitPrice" type="text" />

          <label>Product Qty</label>
          <input id="Quantity" type="text" />

          <label>Product Total</label>
          <input id="TotalPrice" type="text" />

          <button onclick="createData()">Submit</button>
```

```
        </div>
      </div>
    </div>

    <script src="./js/axios.min.js"></script>

    <script>

      async function createData() {
        const productName = document.getElementById("ProductName").value; //
getting the value from the input
        const productCode = document.getElementById("ProductCode").value; //
getting the value from the input
        const img = document.getElementById("Img").value; // getting the
value from the input
        const unitPrice = document.getElementById("UnitPrice").value; //
getting the value from the input
        const quantity = document.getElementById("Quantity").value; //
getting the value from the input
        const totalPrice = document.getElementById("TotalPrice").value; //
getting the value from the input


        // creating an object, key and value. Key should be exact same as
the database column name
        const object = {
          ProductName: productName,
          ProductCode: productCode,
          Img: img,
          UnitPrice: unitPrice,
          Qty: quantity,
          TotalPrice: totalPrice,
        };


        const url = "https://crud.teamrabbil.com/api/v1/CreateProduct"; //
picking the URL first from the API

        const response = await axios.post(url, object); // storing the
response in a variable

        window.location = "index.html"; // after creating the data, it will
redirect to the index page

      }

    </script>

  </body>
```

```
</html>
```

### কোডের ব্যাখ্যা: Create Operation

এই কোডটি একটি ফর্ম ব্যবহার করে নতুন পণ্য ডাটাবেসে যোগ করার জন্য লেখা হয়েছে। ফর্ম থেকে পণ্যের তথ্য নেওয়া হয়, API-তে পাঠানো হয়, এবং সফলভাবে ডেটা যোগ করার পর ব্যবহারকারীকে মূল পৃষ্ঠায় (index.html) নিয়ে যাওয়া হয়।

### কোডের প্রতিটি অংশের ব্যাখ্যা

### HTML ফর্ম

```
<div class="container">
      <div class="row">
        <div class="column">

          <label>Product Name</label>
          <input id="ProductName" type="text" />

          <label>Product Code</label>
          <input id="ProductCode" type="text" />

          <label>Product Image</label>
          <input id="Img" type="text" />

          <label>Unit Price</label>
          <input id="UnitPrice" type="text" />

          <label>Product Qty</label>
          <input id="Quantity" type="text" />

          <label>Product Total</label>
          <input id="TotalPrice" type="text" />

          <button onclick="createData()">Submit</button>

        </div>
      </div>
    </div>
```

**ব্যাখ্যা:**

1. **ইনপুট ফিল্ড:**
   - প্রতিটি `<input>` ট্যাগ একটি ভিন্ন তথ্য নিতে ব্যবহৃত হয়, যেমন পণ্যের নাম, কোড, ছবি, ইউনিট দাম, পরিমাণ, এবং মোট দাম।
   - প্রতিটি ইনপুট ফিল্ডের `id` দেওয়া হয়েছে, যা জাভাস্ক্রিপ্টে ডেটা নেওয়ার জন্য দরকার।

2. **বোতাম:**
   - `Submit` বোতামে ক্লিক করলে, **createData()** ফাংশনটি কল হবে।

### জাভাস্ক্রিপ্ট ফাংশন

```
      async function createData() {

        const productName = document.getElementById("ProductName").value; //
getting the value from the input
        const productCode = document.getElementById("ProductCode").value; //
getting the value from the input
        const img = document.getElementById("Img").value; // getting the
value from the input
        const unitPrice = document.getElementById("UnitPrice").value; //
getting the value from the input
        const quantity = document.getElementById("Quantity").value; //
getting the value from the input
        const totalPrice = document.getElementById("TotalPrice").value; //
getting the value from the input
```

**ব্যাখ্যা:**

- **ডেটা সংগ্রহ:**
  প্রতিটি ইনপুট ফিল্ড থেকে `value` সংগ্রহ করা হচ্ছে।
  উদাহরণস্বরূপ:
  - `ProductName` ইনপুট ফিল্ডের মান নেওয়া হচ্ছে এবং `productName` ভেরিয়েবলে রাখা হচ্ছে।
  - একইভাবে, বাকি ফিল্ডগুলোর ডেটা সংশ্লিষ্ট ভেরিয়েবলে রাখা হচ্ছে।

javascript

```
        // creating an object, key and value. Key should be exact same as
the database column name
        const object = {
          ProductName: productName,
          ProductCode: productCode,
          Img: img,
          UnitPrice: unitPrice,
          Qty: quantity,
          TotalPrice: totalPrice,
        };
```

**ব্যাখ্যা:**

- **অবজেক্ট তৈরি করা:**
  - এখানে একটি জাভাস্ক্রিপ্ট অবজেক্ট তৈরি করা হয়েছে, যেটি API-তে পাঠানো হবে।

- **Key** (উদাহরণ: `ProductName`, `ProductCode`) ডাটাবেসের কলাম নামের সাথে মিলতে হবে।
- **Value** (উদাহরণ: `productName`, `productCode`) ইনপুট ফিল্ড থেকে আসা ডেটা।

```
        const response = await axios.post(url, object); // storing the
response in a variable

        window.location = "index.html"; // after creating the data, it will
redirect to the index page
```

**ব্যাখ্যা:**

- **API URL:**
  `https://crud.teamrabbil.com/api/v1/CreateProduct` API ঠিকানাটি যেখানে নতুন পণ্য যোগ করার জন্য ডেটা পাঠানো হবে।
- **ডেটা পাঠানো:**
  `axios.post(url, object)` ফাংশনটি অবজেক্টের ডেটা API-তে POST রিকোয়েস্ট দিয়ে পাঠায়।
  - `url` : API ঠিকানা।
  - `object` : ইনপুট ফিল্ড থেকে নেওয়া ডেটা।
  - `response` : সার্ভার থেকে পাওয়া রেসপন্স।

```
window.location = "index.html";
```

**ব্যাখ্যা:**

- **Redirect:**
  সফলভাবে নতুন ডেটা যোগ করার পর ব্যবহারকারীকে `index.html` পৃষ্ঠায় নিয়ে যাওয়া হয়।

### কোডের কাজের ধাপ সহজ ভাষায়

1. **ইনপুট নেওয়া:**
   ফর্ম থেকে পণ্যের নাম, কোড, ছবি, দাম, পরিমাণ এবং মোট দাম ইনপুট হিসেবে নেওয়া হয়।
2. **অবজেক্ট তৈরি করা:**
   ইনপুট ডেটাগুলোকে একটি অবজেক্টে সাজানো হয়।
3. **API রিকোয়েস্ট পাঠানো:**
   অবজেক্টটি API-তে POST রিকোয়েস্টের মাধ্যমে পাঠানো হয়।
4. **Redirect করা:**
   ডেটা সফলভাবে যোগ করার পর ব্যবহারকারীকে অন্য পৃষ্ঠায় (index.html) নিয়ে যাওয়া হয়।

### ফাইনাল আউটপুট:

- ফর্মটি পূরণ করে `Submit` ক্লিক করলে, নতুন পণ্য ডাটাবেসে যোগ হবে।
- এরপর ব্যবহারকারীকে মূল তালিকা দেখানোর জন্য `index.html` -এ পাঠানো হবে।

## CRUD : Update

https://crud.teamrabbil.com/api/v1/ReadProductByID/${id}
https://crud.teamrabbil.com/api/v1/UpdateProduct/${productID}

### Update পৃষ্ঠায় যাওয়া (from index.html)

```
<td><button onclick="goUpdatePage('${product["_id"]}') ">Update</button>
</td>
```

```
function goUpdatePage(id) {
    window.location = `./update.html?id=${id}`;
}
```

**ব্যাখ্যা:**

- যখন **goUpdatePage(id)** ফাংশনটি কল হবে, এটি ব্যবহারকারীকে `update.html` পৃষ্ঠায় নিয়ে যাবে এবং `id` প্যারামিটারটি URL-এ পাঠাবে।

```
<!DOCTYPE html>

<html lang="en">

  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <link rel="stylesheet" href="./css/milligram.css" />
  </head>

  <body>

    <div class="container">
      <div class="row">
        <div class="column">

          <label>Product ID</label>
          <input id="ProductID" type="text" />

          <label>Product Name</label>
          <input id="ProductName" type="text" />

          <label>Product Code</label>
          <input id="ProductCode" type="text" />
```

```
          <label>Product Image</label>
          <input id="Img" type="text" />

          <label>Unit Price</label>
          <input id="UnitPrice" type="text" />

          <label>Product Qty</label>
          <input id="Quantity" type="text" />

          <label>Product Total</label>
          <input id="TotalPrice" type="text" />

          <button onclick="updateData()">Submit</button>

        </div>
      </div>
    </div>

    <script src="./js/axios.min.js"></script>

    <script>

      // showing the existing data in the update page
      async function fillExistingData() {
        const urlParameters = new URLSearchParams(window.location.search);
// all the query parameters will be stored in 'urlParameters'

        const id = urlParameters.get("id"); // from the 'urlParameters' we
will pick the 'id' parameter.

        // existing details show
        const URL =
`https://crud.teamrabbil.com/api/v1/ReadProductByID/${id}`; // picking the
URL first from the API

        const response = await axios.get(URL); // storing the response in a
variable

        const data = response.data["data"][0]; //

        document.getElementById("ProductID").value = data["_id"];
        document.getElementById("ProductName").value = data["ProductName"];
        document.getElementById("ProductCode").value = data["ProductCode"];
        document.getElementById("Img").value = data["Img"];
        document.getElementById("UnitPrice").value = data["UnitPrice"];
        document.getElementById("Quantity").value = data["Qty"];
        document.getElementById("TotalPrice").value = data["TotalPrice"];
      }

      fillExistingData();
```

```
      // data update
      async function updateData() {

        const productID = document.getElementById("ProductID").value;
        const productName = document.getElementById("ProductName").value;
        const productCode = document.getElementById("ProductCode").value;
        const img = document.getElementById("Img").value;
        const unitPrice = document.getElementById("UnitPrice").value;
        const quantity = document.getElementById("Quantity").value;
        const productTotal = document.getElementById("TotalPrice").value;

        const object = {

          ProductName: productName,
          ProductCode: productCode,
          Img: img,
          UnitPrice: unitPrice,
          Qty: quantity,
          TotalPrice: productTotal,
        };

        const url =
`https://crud.teamrabbil.com/api/v1/UpdateProduct/${productID}`; // picking
the URL first from the API

        const response = await axios.post(url, object); // posting the data
to the API with post method with url and object

        window.location = "index.html"; // after creating the data, it will
redirect to the index page
      }

    </script>

  </body>

</html>
```

## কোডের ব্যাখ্যা: Update Operation

এই কোডটি পণ্যের বিদ্যমান তথ্য দেখানো এবং সেগুলো আপডেট করার জন্য লেখা হয়েছে। যখন ব্যবহারকারী আপডেট পৃষ্ঠায় যান, বিদ্যমান তথ্যগুলো দেখানো হয় এবং সেগুলো সম্পাদনা করে সংরক্ষণ করা হয়।

### কোডের প্রতিটি অংশের ব্যাখ্যা

### HTML ফর্ম

```
    <div class="container">

      <div class="row">

        <div class="column">

          <label>Product ID</label>
          <input id="ProductID" type="text" />

          <label>Product Name</label>
          <input id="ProductName" type="text" />

          <label>Product Code</label>
          <input id="ProductCode" type="text" />

          <label>Product Image</label>
          <input id="Img" type="text" />

          <label>Unit Price</label>
          <input id="UnitPrice" type="text" />

          <label>Product Qty</label>
          <input id="Quantity" type="text" />

          <label>Product Total</label>
          <input id="TotalPrice" type="text" />

          <button onclick="updateData()">Submit</button>

        </div>

      </div>

    </div>
```

**ব্যাখ্যা:**

1. **ইনপুট ফিল্ড:**
   - বিদ্যমান তথ্য দেখানো এবং নতুন তথ্য ইনপুট দেওয়ার জন্য প্রতিটি `<input>` ট্যাগ ব্যবহার করা হয়েছে।
   - `ProductID` ফিল্ডটি **readonly** নয়, তবে এটি শুধুমাত্র API-এর জন্য প্রয়োজন।
2. **Submit বোতাম:**
   - `Submit` বোতামে ক্লিক করলে **updateData()** ফাংশনটি কল হবে।

## বিদ্যমান তথ্য দেখানো

```
      // showing the existing data in the update page
      async function fillExistingData() {

        const urlParameters = new URLSearchParams(window.location.search);
// all the query parameters will be stored in 'urlParameters'
        const id = urlParameters.get("id"); // from the 'urlParameters' we
will pick the 'id' parameter.

        // existing details show
        const URL =
`https://crud.teamrabbil.com/api/v1/ReadProductByID/${id}`; // picking the
URL first from the API

        const response = await axios.get(URL); // storing the response in a
variable

        const data = response.data["data"][0]; //

        document.getElementById("ProductID").value = data["_id"];
        document.getElementById("ProductName").value = data["ProductName"];
        document.getElementById("ProductCode").value = data["ProductCode"];
        document.getElementById("Img").value = data["Img"];
        document.getElementById("UnitPrice").value = data["UnitPrice"];
        document.getElementById("Quantity").value = data["Qty"];
        document.getElementById("TotalPrice").value = data["TotalPrice"];
      }

      fillExistingData();
```

### ব্যাখ্যা:

1. **URL থেকে `id` নেওয়া:**
   - `window.location.search` থেকে `id` প্যারামিটার নেওয়া হয়েছে।
   - উদাহরণস্বরূপ: `update.html?id=123` থেকে `id = 123`।
2. **API কল:**
   - `https://crud.teamrabbil.com/api/v1/ReadProductByID/${id}` API ব্যবহার করে পণ্যের বিদ্যমান তথ্য আনা হয়েছে।
3. **ইনপুট ফিল্ড পূরণ:**
   - বিদ্যমান তথ্য ইনপুট ফিল্ডে দেখানো হয়েছে।

## ডেটা আপডেট করা

```
      async function updateData() {

        const productID = document.getElementById("ProductID").value;
        const productName = document.getElementById("ProductName").value;
```

```
        const productCode = document.getElementById("ProductCode").value;
        const img = document.getElementById("Img").value;
        const unitPrice = document.getElementById("UnitPrice").value;
        const quantity = document.getElementById("Quantity").value;
        const productTotal = document.getElementById("TotalPrice").value;

        const object = {

          ProductName: productName,
          ProductCode: productCode,
          Img: img,
          UnitPrice: unitPrice,
          Qty: quantity,
          TotalPrice: productTotal,
        };

        const url =
`https://crud.teamrabbil.com/api/v1/UpdateProduct/${productID}`; // picking
the URL first from the API

        const response = await axios.post(url, object); // posting the data
to the API with post method with url and object

        window.location = "index.html"; // after creating the data, it will
redirect to the index page
      }
```

### ব্যাখ্যা:

1. **ইনপুট থেকে ডেটা সংগ্রহ:**
   - ইনপুট ফিল্ড থেকে নতুন ডেটা নেওয়া হয়েছে।
   - উদাহরণ: `document.getElementById("ProductName").value`।
2. **অবজেক্ট তৈরি:**
   - নতুন ডেটা একটি অবজেক্টে রাখা হয়েছে, যার **Key** ডাটাবেস কলামের নামের সাথে মিলতে হবে।
3. **API কল:**
   - `axios.post(url, object)` API-তে ডেটা আপডেট করার জন্য পাঠানো হয়েছে।
4. **Redirect:**
   - ডেটা সফলভাবে আপডেট করার পর ব্যবহারকারীকে `index.html`-এ নিয়ে যাওয়া হয়েছে।

## কোডের কাজের ধাপ সহজ ভাষায়

1. **Update পৃষ্ঠায় যাওয়া:**
   - যখন ব্যবহারকারী **Update** বোতামে ক্লিক করবে, তখন তাকে `update.html`-এ নিয়ে যাওয়া হবে এবং পণ্যের `id` পাস করা হবে।
2. **বিদ্যমান তথ্য দেখানো:**

- API থেকে পণ্যের তথ্য নিয়ে ইনপুট ফিল্ডে দেখানো হবে।

3. **ডেটা সম্পাদনা:**
   - ব্যবহারকারী ইনপুট ফিল্ড সম্পাদনা করতে পারবে।
4. **Submit করলে:**
   - সম্পাদিত ডেটা API-তে পাঠানো হবে।
   - আপডেট সফল হলে, ব্যবহারকারীকে `index.html`-এ ফিরিয়ে দেওয়া হবে।

## ফাইনাল আউটপুট:

- ব্যবহারকারী বিদ্যমান পণ্যের তথ্য সম্পাদনা করতে পারবে এবং সেগুলো সংরক্ষণ করতে পারবে।
- সম্পাদনার পর তালিকায় আপডেটেড তথ্য দেখতে পারবে।