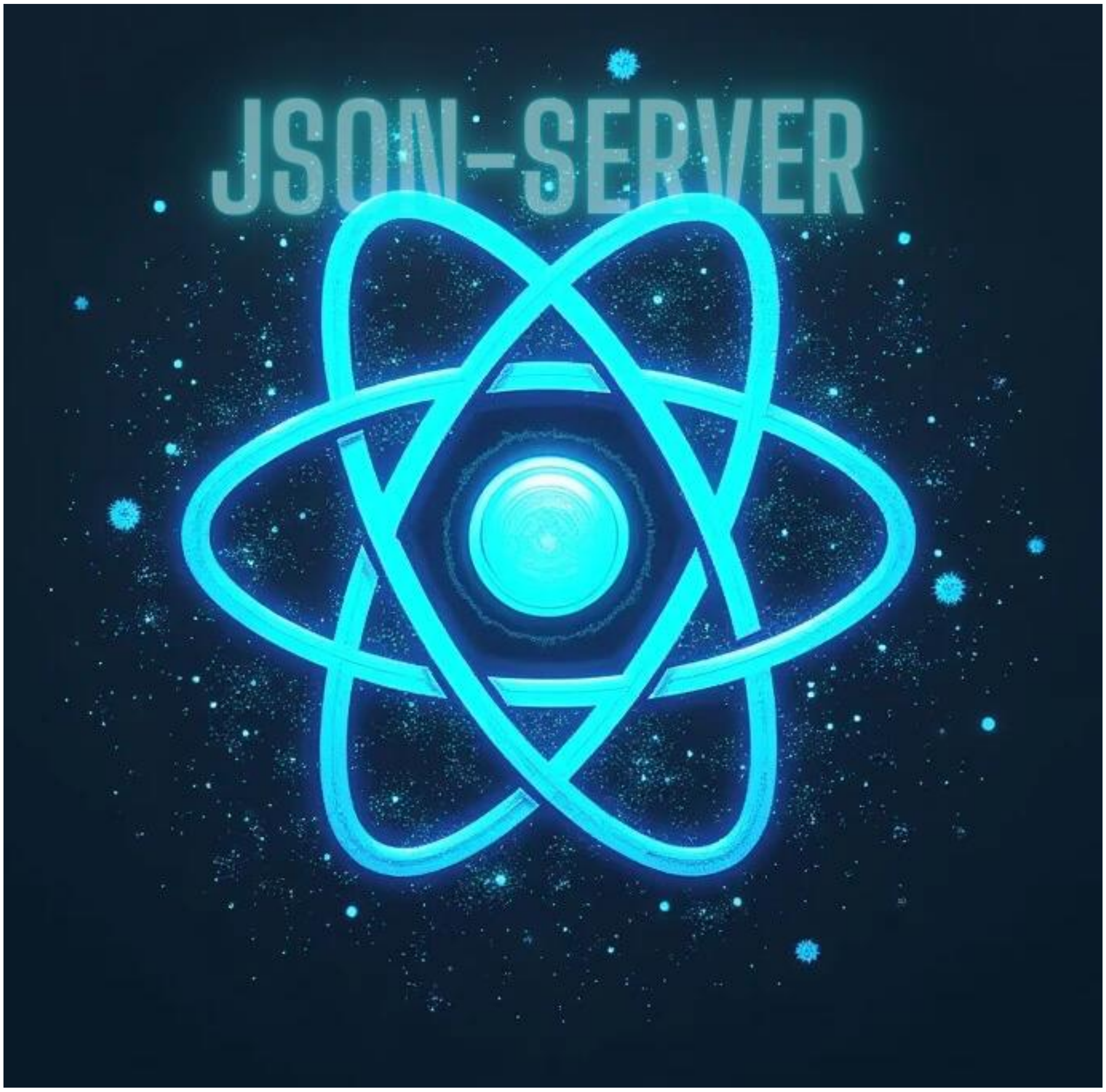

React – Json-Server

Question 1: What is Json-Server? How we use in React ?

Ans.

JSON-Server is a simple tool to create a mock REST API with a fake database. It allows you to quickly test your front-end applications without needing a real back-end.

How to use JSON-Server in React:

1. **Install JSON-Server:**

```
npm install -g json-server
```

2. **Create a db.json file:**

```
{
    "products": [
        { "id": 1, "name": "Laptop", "price": 500 },
        { "id": 2, "name": "Phone", "price": 200 }
    ]
}
```

3. **Start JSON-Server:**

```
json-server --watch db.json --port 5000
```

4. **Use in React:**

Fetch data using `fetch` or `axios` in your React component:

```
import React, { useEffect, useState } from "react";
import axios from "axios";

const App = () => {
  const [products, setProducts] = useState([]);

  useEffect(() => {
    axios.get("http://localhost:5000/products")
      .then((response) => setProducts(response.data))
      .catch((error) => console.error(error));
  }, []);

  return (
    <div>
      <h1>Products</h1>
      <ul>
        {products.map((product) => (
          <li key={product.id}>{product.name} - ${product.price}</li>
        ))}
      </ul>
    </div>
  );
};

export default App;
```

JSON-Server is great for development and testing!

Question 2: How do you fetch data from a Json-server API in React? Explain the role of fetch() or axios()in making API requests.

Ans.

Fetching data from a JSON-Server API in React:

You can use either the **`fetch()` API** or the **Axios library** to make HTTP requests to the JSON-Server.

1. **Using fetch() :**

   The `fetch()` method is a built-in JavaScript function for making HTTP requests.

```
import React, { useEffect, useState } from "react";

const App = () => {
  const [data, setData] = useState([]);

  useEffect(() => {
    fetch("http://localhost:5000/products") // URL of the JSON-Server API
      .then((response) => response.json()) // Convert response to JSON
      .then((result) => setData(result)) // Update the state with fetched data
      .catch((error) => console.error("Error fetching data:", error));
  }, []);

  return (
    <div>
      <h1>Products</h1>
      <ul>
        {data.map((item) => (
          <li key={item.id}>{item.name} - ${item.price}</li>
        ))}
      </ul>
    </div>
  );
};

export default App;
```

2. **Using Axios :**

   Axios is a third-party library that simplifies API requests. Install it using:

```
      npm install axios
```

```
import React, { useEffect, useState } from "react";
import axios from "axios";

const App = () => {
  const [data, setData] = useState([]);

  useEffect(() => {
    axios
      .get("http://localhost:5000/products") // URL of the JSON-Server API
      .then((response) => setData(response.data)) // Update the state with
fetched data
      .catch((error) => console.error("Error fetching data:", error));
  }, []);

  return (
    <div>
      <h1>Products</h1>
      <ul>
```

```
        {data.map((item) => (
          <li key={item.id}>{item.name} - ${item.price}</li>
        ))}
      </ul>
    </div>
  );
};

export default App;
```

Role of fetch() or axios():

1. **Fetch() :**

- Native to JavaScript, works in modern browsers.
- Requires manual handling of JSON parsing (`response.json()`).
- Doesn't handle HTTP errors directly, so additional error handling is needed.

2. **Axios :**

- Simplifies API requests with a cleaner syntax.
- Automatically parses JSON responses.
- Handles HTTP errors and request cancellation easily.
- Provides additional features like interceptors and timeout settings.