

```
// Orders.js
const mongoose = require('mongoose');

const OrdersSchema = new mongoose.Schema({
  OrderID: String,
  CustomerName: String,
  ProductName: String,
  Quantity: Number,
  Date: { type: Date, default: Date.now }
});

module.exports = mongoose.model('Orders', OrdersSchema);
```

```
// server.js
const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');
const Orders = require('./Orders');

const app = express();
app.use(express.json());
app.use(cors());

mongoose.connect('mongodb://localhost:27017/', { useNewUrlParser: true,
useUnifiedTopology: true })
  .then(() => console.log('Connected to MongoDB'))
  .catch((err) => console.log(err));

app.get('/api', async (req, res) => {
  try {
    const data = await Orders.find();
    res.status(200).send(data);
  } catch (error) {
    res.status(500).send(error);
  }
});

app.get('/api/id/:id', async (req, res) => {
  try {
    const data = await Orders.findOne({ OrderID: req.params.id });
    data ? res.status(200).send(data) : res.send('Order Not Found');
  }
});
```

```

    } catch (error) {
      res.status(500).send(error);
    }
  });

app.post('/api/create', async (req, res) => {
  try {
    const { OrderID, CustomerName, ProductName, Quantity } = req.body;
    const newOrder = new Orders({ OrderID, CustomerName, ProductName,
Quantity });
    await newOrder.save();
    res.status(200).send(newOrder);
  } catch (error) {
    res.status(500).send(error);
  }
});

app.delete('/api/delete/:id', async (req, res) => {
  try {
    const data = await Orders.findOneAndDelete({ OrderID: req.params.id
});
    data ? res.status(200).send('Order Deleted') : res.send('Order Not
Found');
  } catch (error) {
    res.status(500).send(error);
  }
});

app.put('/api/update/:id', async (req, res) => {
  try {
    const data = await Orders.findOneAndUpdate({ OrderID: req.params.id
}, { $set: { Quantity: req.body.Quantity } }, { new: true });
    data ? res.status(200).send('Updated Successfully') :
res.send('Order Not Found');
  } catch (error) {
    res.status(500).send(error);
  }
});

app.listen(5002, () => {
  console.log('Server is running on port 5002');
});

```

```
// AddOrder.js
import React, { useState } from "react";
import axios from 'axios';
import "bootstrap/dist/css/bootstrap.min.css";
import { Button, Form, FormGroup, Input, Label } from "reactstrap";

function AddOrder() {
  const [data, sendData] = useState({
    OrderID: "",
    CustomerName: "",
    ProductName: "",
    Quantidy: 0
  });

  const handleSubmit = async (e) => {
    try {
      e.preventDefault();
      const response = await
axios.post("http://localhost:5002/api/create", data);
      if (response) {
        alert('Added Successfully');
      } else {
        alert('Cannot add');
      }
    } catch (error) {
      console.log(error);
    }
  };

  return (
    <div>
      <Form onSubmit={handleSubmit}>
        <FormGroup>
          <Label>Order Id: </Label>
          <Input type="text" placeholder="Enter Id"
value={data.OrderID} onChange={(e) => sendData({ ...data, OrderID:
e.target.value })} />
        </FormGroup>
        <FormGroup>
          <Label>Customer Name: </Label>
          <Input type="text" placeholder="Enter Customer Name"
value={data.CustomerName} onChange={(e) => sendData({ ...data,
CustomerName: e.target.value })} />

```

```

        </FormGroup>
        <FormGroup>
            <Label>Product Name: </Label>
            <Input type="text" placeholder="Enter Product Name"
value={data.ProductName} onChange={(e) => sendData({ ...data,
ProductName: e.target.value })} />
        </FormGroup>
        <FormGroup>
            <Label>Quantidy: </Label>
            <Input type="number" placeholder="Enter Quantidy"
value={data.Quantidy} onChange={(e) => sendData({ ...data, Quantidy:
e.target.value })} />
        </FormGroup>
        <Button color="primary" type="submit">Submit</Button>
    </Form>
</div>
);
};

export default AddOrder;

```

```

// DeleteOrder.js
import React, { useState } from "react";
import axios from 'axios';
import { Button, Form, FormGroup, Input, Label } from "reactstrap";

function DeleteOrder() {
    const [id, setId] = useState("");

    const handleSubmit = async (e) => {
        try {
            e.preventDefault();
            const response = await
axios.delete(`http://localhost:5002/api/delete/${id}`);
            if (response) {
                alert('Deleted');
            } else {
                alert('Not Found');
            }
        } catch (error) {
            console.log(error);
        }
    }
}

```

```

    }
  };

  return (
    <div>
      <Form onSubmit={handleSubmit}>
        <FormGroup>
          <Label>Order Id: </Label>
          <Input type="text" placeholder="Enter Id" value={id}
onChange={e => setId(e.target.value)} />
        </FormGroup>
        <Button color="primary" type="submit">Submit</Button>
      </Form>
    </div>
  );
};

export default DeleteOrder;

```

```

// SearchOrder.js
import React, { useState } from "react";
import axios from 'axios';
import { Button, Form, FormGroup, Input, Label } from "reactstrap";

function SearchOrder() {
  const [id, setId] = useState("");
  const [data, setData] = useState(null);

  const handleSubmit = async (e) => {
    try {
      e.preventDefault();
      const response = await
axios.get(`http://localhost:5002/api/id/${id}`);
      if (response) {
        setData(response.data);
      } else {
        alert('Not Found');
      }
    } catch (error) {
      console.log(error);
    }
  };
};

```

```

return (
  <div>
    <Form onSubmit={handleSubmit}>
      <FormGroup>
        <Label>Order Id: </Label>
        <Input type="text" placeholder="Enter Id" value={id}
onChange={(e) => setId(e.target.value)} />
      </FormGroup>
      <Button color="primary" type="submit">Submit</Button>
    </Form>
    {data && <div>
      <p>{data.OrderID}</p>
      <p>{data.CustomerName}</p>
      <p>{data.ProductName}</p>
      <p>{data.Quantidy}</p>
      <p>{data.Date}</p>
    </div>}
  </div>
);
};

export default SearchOrder;

```

```

// UpdateOrder.js
import React, { useState } from "react";
import axios from 'axios';
import { Button, Form, FormGroup, Input, Label } from "reactstrap";

function UpdateOrder() {
  const [id, setId] = useState("");
  const [q, setQ] = useState(null);

  const handleSubmit = async (e) => {
    try {
      e.preventDefault();
      const response = await
axios.put(`http://localhost:5002/api/update/${id}`, { Quantidy: q });
      if (response) {
        alert('Updated');
      }
    }
  }
}

```

```

    } else {
      alert('Not Found');
    }
  } catch (error) {
    console.log(error);
  }
};

return (
  <div>
    <Form onSubmit={handleSubmit}>
      <FormGroup>
        <Label>Order Id: </Label>
        <Input type="text" placeholder="Enter Id" value={id}
onChange={(e) => setId(e.target.value)} />
      </FormGroup>
      <FormGroup>
        <Label>Updated Quantity: </Label>
        <Input type="number" placeholder="Enter Quantity" value={q}
onChange={(e) => setQ(e.target.value)} />
      </FormGroup>
      <Button color="primary" type="submit">Submit</Button>
    </Form>
  </div>
);
};

export default UpdateOrder;

```

```

import React from 'react';
import { BrowserRouter as Router, Routes, Route, Link } from
"react-router-dom";

// Import your components
import AddOrder from './components/AddOrder';
import UpdateOrder from './components/UpdateOrder';
import SearchOrder from './components/SearchOrder';
import DeleteOrder from './components/DeleteOrder';

const App = () => {

```

```

return (
  <Router>
    <div>
      <nav>
        <ul>
          <li><Link to="/add">Add Order</Link></li>
          <li><Link to="/search">Search Order</Link></li>
          <li><Link to="/update">Update Order</Link></li>
          <li><Link to="/delete">Delete Order</Link></li>
        </ul>
      </nav>
    </div>
    <Routes>
      <Route path="/add" element={<AddOrder />} />
      <Route path="/search" element={<SearchOrder />} />
      <Route path="/update" element={<UpdateOrder />} />
      <Route path="/delete" element={<DeleteOrder />} />
    </Routes>
  </Router>
);
};

export default App;

```

```

import 'bootstrap/dist/css/bootstrap.min.css'; // Import Bootstrap CSS
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
import './index.css'; // Optional: Import any global styles if you have
one

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);

```



```
PS C:\Users\srini\AppData> mkdir backend
```

Directory: C:\Users\srini\AppData

Mode	LastWriteTime	Length	Name
d----	12-11-2024 01:44		backend

```
PS C:\Users\srini\AppData> cd backend
```

Create files

server.js

orders.js and write code,save

```
PS C:\Users\srini\AppData\backend> npm init -y
```

```
{
  "name": "backend",
  "version": "1.0.0",
  "main": "Orders.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}
```

```
PS C:\Users\srini\AppData\frontend> echo webpack.config.js
```

InputObject[1]:

webpack.config.js

```
PS C:\Users\srini\AppData\frontend> echo > webpack.config.js
```

cmdlet Write-Output at command pipeline position 1

Supply values for the following parameters:

InputObject[0]:

```

const path = require('path');
const HtmlWebpackPlugin = require('html-webpack-plugin');

module.exports = {
  entry: './src/app/index.js',
  output: {
    path: path.resolve(__dirname, 'dist'),
    filename: 'bundle.js',
    clean: true,
  },
  mode: process.env.NODE_ENV || 'development',
  module: {
    rules: [
      {
        test: /\.js$/,
        use: 'babel-loader',
      },
      {
        test: /\.svg$/,
        loader: "svg-inline-loader",
      },
      {
        test: /\.css$/,
        use: ['style-loader', 'css-loader']
      }
    ]
  },
  plugins: [
    new HtmlWebpackPlugin({
      template: './public/index.html',
    }),
  ],
};

```

After writing webpack file, create a sub folder called app, and inside that create index.js file

```

// app/index.js

// Sample order data, could be dynamic or fetched from an API in a
real-world scenario
const orders = [
  { id: 1, customer: "John Doe", totalAmount: 250, status: "Shipped" },

```

```
    { id: 2, customer: "Jane Smith", totalAmount: 450, status:
"Processing" },
    { id: 3, customer: "Alice Johnson", totalAmount: 320, status:
"Delivered" },
    { id: 4, customer: "Bob Brown", totalAmount: 150, status: "Pending" }
];

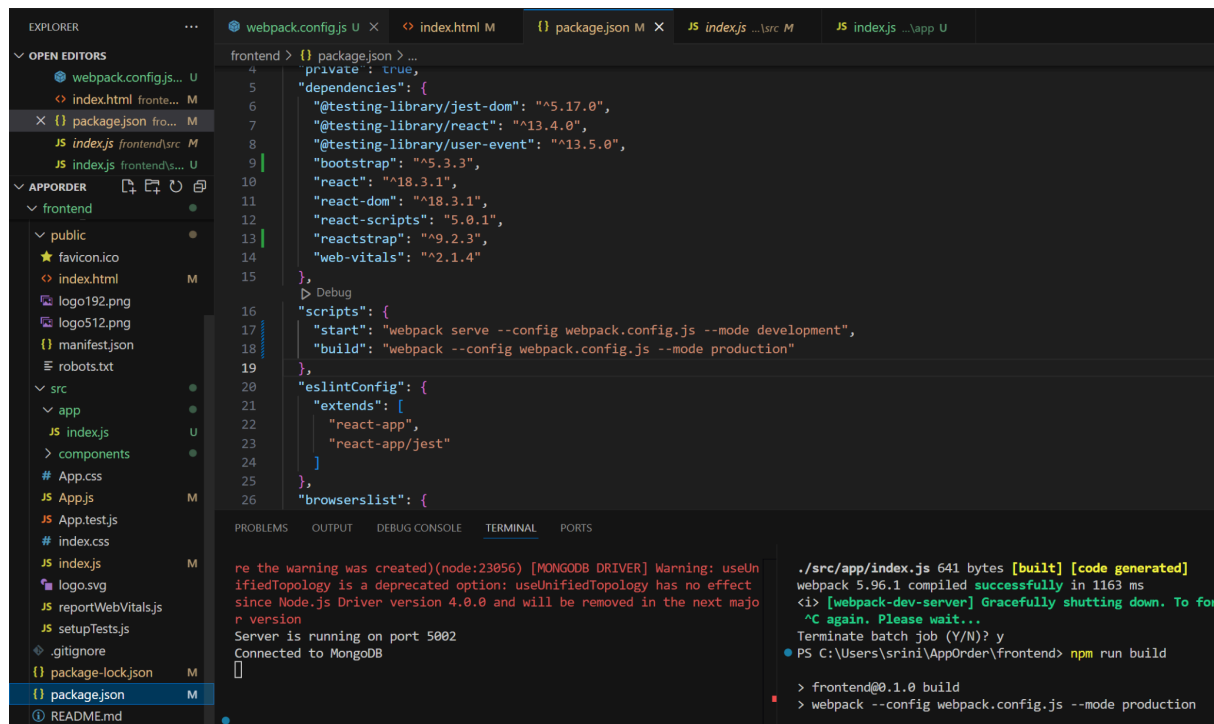
// Log the order data to the console for debugging or inspection
console.log("Orders List:", orders);

// Export the orders array for use in other parts of the application
export default orders;
```

/public/index.html  
Write the code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Student Enrollment System</title>
</head>
<body>
  <div id="root"></div>
</body>
</html>
```

Now :



In package.json, change scripts for webpacks, do this after showing all frontend and backend

PS C:\Users\sirini\AppData\Local\Temp\webpack-5.96.1\webpack-5.96.1> npm start

> frontend@0.1.0 start

> webpack serve --config webpack.config.js --mode development

<i> [webpack-dev-server] Project is running at:

<i> [webpack-dev-server] Loopback: http://localhost:8080/

<i> [webpack-dev-server] On Your Network (IPv4): http://172.22.35.231:8080/

<i> [webpack-dev-server] Content not from webpack is served from

'C:\Users\sirini\AppData\Local\Temp\webpack-5.96.1\webpack-5.96.1' directory

asset bundle.js 306 KiB [emitted] (name: main)

asset index.html 283 bytes [emitted]

runtime modules 27.4 KiB 12 modules

modules by path ./node\_modules/ 214 KiB

modules by path ./node\_modules/webpack-dev-server/client/ 76.3 KiB 16 modules

modules by path ./node\_modules/webpack/hot/\*.js 5.2 KiB

./node\_modules/webpack/hot/dev-server.js 1.91 KiB [built] [code generated]

./node\_modules/webpack/hot/log.js 1.75 KiB [built] [code generated]

+ 2 modules

modules by path ./node\_modules/html-entities/lib/\*.js 114 KiB

./node\_modules/html-entities/lib/index.js 5.72 KiB [built] [code generated]

./node\_modules/html-entities/lib/named-references.js 107 KiB [built] [code generated]

./node\_modules/html-entities/lib/numeric-unicode-map.js 464 bytes [built] [code

generated]

./node\_modules/html-entities/lib/surrogate-pairs.js 613 bytes [built] [code generated]

```
./node_modules/ansi-html-community/index.js 4.25 KiB [built] [code generated]
./node_modules/events/events.js 13.9 KiB [built] [code generated]
./src/app/index.js 641 bytes [built] [code generated]
webpack 5.96.1 compiled successfully in 1163 ms //this indicates that webpack is done
successfully
<i> [webpack-dev-server] Gracefully shutting down. To force exit, press ^C again. Please
wait...
```

Webpack is compiled successfully

To check if your application is running properly after the build and if the Webpack bundle has been correctly created, follow these steps:

## 1. Ensure Webpack Dev Server is Running

If you're using Webpack Dev Server for development, you should run the following command:

```
bash
Copy code
npm start
```











This will start the development server and you should see the output indicating that it's running on a local address, typically something like:

```
csharp
Copy code
Project is running at:
Loopback: http://localhost:8080/
On Your Network (IPv4): http://<your-ip>:8080/
```

Visit <http://localhost:8080/> in your browser. If everything is set up correctly, you should see your application rendered.

Directory:  
//in frontend you can see dist/ bundle.js —proof for webpack

▼ APPORDER	
▼ backend	
JS Orders.js	
{ } package.json	
JS server.js	
▼ frontend	●
> build	
▼ dist	●
JS bundle.js	U
<> index.html	U
> node_modules	
▼ public	●
★ favicon.ico	
<> index.html	M
🖼 logo192.png	
🖼 logo512.png	
{ } manifest.json	
≡ robots.txt	
▼ src	●
▼ app	●
JS index.js	U
▼ components	●
JS AddOrder.js	U
JS DeleteOrder.js	U
JS SearchOrder.js	U
JS UpdateOrder.js	U
# App.css	
JS App.js	M
JS App.test.js	
# index.css	
JS index.js	M
🖼 logo.svg	

	logo192.png	
	logo512.png	
	manifest.json	
	robots.txt	
▼	src	●
▼	app	●
JS	index.js	U
▼	components	●
JS	AddOrder.js	U
JS	DeleteOrder.js	U
JS	SearchOrder.js	U
JS	UpdateOrder.js	U
#	App.css	
JS	App.js	M
JS	App.test.js	
#	index.css	
JS	index.js	M
	logo.svg	
JS	reportWebVitals.js	
JS	setupTests.js	
	.gitignore	
	package-lock.json	M
	package.json	M
	README.md	
	webpack.config.js	U