Sure! To create a beautiful frontend for showing the RAM and CPU usage data, we can use a frontend framework like React.js. Here are the steps:

1. Install React.js and other necessary packages:

```
npx create-react-app cpu-ram-usage-app
cd cpu-ram-usage-app
npm install axios
```

2. Replace the contents of `src/App.js` with the following code:

```
import React, { useState, useEffect } from "react";
import axios from "axios";
import "./App.css";

function App() {
  const [usageData, setUsageData] = useState({});

  useEffect(() => {
    const fetchData = async () => {
      const result = await axios.get("/usage");
      setUsageData(result.data);
    };
    fetchData();
    const intervalId = setInterval(fetchData, 5000); // Fetch data every 5 seconds
    return () => clearInterval(intervalId);
  }, []);

  return (
    <div className="container">
      <h1>CPU and RAM Usage</h1>
      <div className="usage-container">
        <div className="usage">
          <h2>CPU Usage</h2>
          <div className="usage-bar">
            <div
              className="usage-progress"
              style={{ width: `${usageData.cpu}%` }}
            ></div>
          </div>
          <p>{usageData.cpu.toFixed(2)}%</p>
        </div>
        <div className="usage">
          <h2>RAM Usage</h2>
          <div className="usage-bar">
            <div
              className="usage-progress"
              style={{ width: `${usageData.ram}%` }}
            ></div>
          </div>
          <p>{usageData.ram.toFixed(2)}%</p>
        </div>
      </div>
    </div>
  );
}

export default App;
```

3. Add the following CSS styles to `src/App.css`:

```
.container {
  margin: 50px auto;
  text-align: center;
}
```

```
.usage-container {
  display: flex;
  justify-content: space-between;
  margin-top: 50px;
}

.usage {
  flex: 1;
  margin: 10px;
}

.usage h2 {
  font-size: 24px;
}

.usage-bar {
  width: 100%;
  height: 20px;
  background-color: #f0f0f0;
  border-radius: 10px;
  overflow: hidden;
}

.usage-progress {
  height: 100%;
  background-color: #34c759;
}

.usage p {
  font-size: 18px;
  margin-top: 10px;
}
```

4. Replace the contents of `src/index.js` with the following code:

```
import React from "react";
import ReactDOM from "react-dom";
import App from "./App";
import "./index.css";

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById("root")
);
```

5. Add the following CSS styles to `src/index.css`:

```
body {
  margin: 0;
  padding: 0;
  font-family: sans-serif;
}
```

Sure, here's the complete `index.js` file with the necessary modifications to serve the React app:

```
const express = require("express");
const app = express();
const shell = require("shelljs");
const path = require("path");

// Serve the static files from the React app
app.use(express.static(path.join(__dirname, "client/build")));

// API endpoint to get the CPU and RAM usage data
```

```
app.get("/usage", (req, res) => {
  const totalMem = shell.exec("awk '/^MemTotal:/{print $2}' /proc/meminfo", {
    silent: true,
  }).stdout;
  const freeMem = shell.exec("awk '/^MemFree:/{print $2}' /proc/meminfo", {
    silent: true,
  }).stdout;
  const usedMem = totalMem - freeMem;
  const ramUsage = (usedMem / totalMem) * 100;

  const cpuUsage = shell
    .exec("top -bn1 | awk '/%Cpu/{print $2}' | cut -d'.' -f1", { silent: true })
    .stdout.trim();

  res.json({
    ram: ramUsage,
    cpu: cpuUsage,
  });
});

// Handles any requests that don't match the above
app.get("*", (req, res) => {
  res.sendFile(path.join(__dirname + "/client/build/index.html"));
});

const port = process.env.PORT || 5000;
app.listen(port);

console.log(`Live RAM and CPU usage app listening on port ${port}`);
```

Note that we added two new endpoints:

- `/usage` to get the CPU and RAM usage data
- `*` to handle any other requests and serve the React app

Also, we modified the code to serve the static files from the `client/build` directory, which will be generated when you run the `npm run build` command in the React app's directory.