

FILE: backend\controllers\notes.js

```

const Note = require("../models/Note");

// Get all
const getAllNotes = async (req, res) => {
  try {
    const notes = await Note.find();
    res.json(notes);
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
};

const createOneNote = async (req, res) => {
  const note = new Note({
    company: req.body.company,
    asset: req.body.asset,
    content: req.body.content,
  });
  try {
    await note.save();
    res.status(201).json(note);
  } catch (err) {
    res.status(400).json({ message: err.message });
  }
};

const updateOneNote = async (req, res) => {
  if (req.body.company !== null) {
    res.note.company = req.body.company;
  }
  if (req.body.asset !== null) {
    res.note.asset = req.body.asset;
  }
  if (req.body.content !== null) {
    res.note.content = req.body.content;
  }

  try {
    const updatedNote = await res.note.save();
    res.json(updatedNote);
  } catch (err) {
    res.status(400).json({ message: err.message });
  }
};

const deleteOneNote = async (req, res) => {
  try {
    await res.note.deleteOne();
    res.json({ message: "Deleted note" });
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
};

module.exports = {
  getAllNotes,
  createOneNote,
  updateOneNote,

```

```

    deleteOneNote,
  };

```

FILE: backend\controllers\profiles.js

```

const Profile = require("../models/Profile");

const getAllProfiles = async (req, res) => {
  try {
    const profiles = await Profile.find({});
    // const count = await Profile.countDocuments();
    // console.log(count);
    res.json(profiles.slice(0, 10));
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
};

const getOneProfile = async (req, res) => {
  try {
    const query = req.params.ticker;
    const profile = await Profile.find({
      ticker: new RegExp("^" + query, "i"),
    });
    // if (profile.length === 0) {
    //   return res.status(404).json({ message: "Could not find profile" });
    // }
    res.json(profile);
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
};

const debug = async (req, res) => {
  try {
    const profiles = await Profile.find();
    res.json({ count: profiles.length, first: profiles[0] });
  } catch (err) {
    res.status(500).json({ message: err.message });
  }
};

module.exports = { getAllProfiles, getOneProfile, debug };

```

FILE: backend\controllers\treasuryRates.js

FILE: backend\db\index.js

```
// db/index.js
module.exports = {
  investment_data_db: require("../investmentData"),
  macroeconomic_data_db: require("../macroeconomicData"),
  stock_data_db: require("../stockData"),
};
```

FILE: backend\db\investmentData.js

```
require("dotenv").config();
const mongoose = require("mongoose");

// Optional: Add better connection options
const connectionOptions = {
  useNewUrlParser: true,
  useUnifiedTopology: true,
  serverSelectionTimeoutMS: 30000, // Timeout after 30s if cannot connect
  socketTimeoutMS: 30000, // Prevents hanging socket operations
};

const investment_data_db = mongoose.createConnection(
  process.env.DATABASE_URL_INVESTMENT_DATA,
  connectionOptions
);

// Monitor connection events
investment_data_db.on("connected", () => {
  console.log("Connected to investment DB:", investment_data_db.name);
});

investment_data_db.on("error", (err) => {
  console.error("Investment DB connection error:", err);
});

investment_data_db.on("disconnected", () => {
  console.warn("Investment DB disconnected");
});

// Optional: If using Mongoose 7+, you can await the promise
investment_data_db.asPromise().catch((err) => {
  console.error("Initial DB connection error:", err);
});

module.exports = investment_data_db;
```

FILE: backend\db\macroeconomicData.js

```

require("dotenv").config();
const mongoose = require("mongoose");

// Optional: Add better connection options
const connectionOptions = {
  useNewUrlParser: true,
  useUnifiedTopology: true,
  serverSelectionTimeoutMS: 30000, // Timeout after 30s if cannot connect
  socketTimeoutMS: 30000, // Prevents hanging socket operations
};

const macroeconomic_data_db = mongoose.createConnection(
  process.env.DATABASE_URL_MACROECONOMIC_DATA,
  connectionOptions
);

// Monitor connection events
macroeconomic_data_db.on("connected", () => {
  console.log("Connected to macroeconomic DB:", macroeconomic_data_db.name);
});

macroeconomic_data_db.on("error", (err) => {
  console.error("Macroeconomic DB connection error:", err);
});

macroeconomic_data_db.on("disconnected", () => {
  console.warn("Macroeconomic DB disconnected");
});

// Optional: If using Mongoose 7+, you can await the promise
macroeconomic_data_db.asPromise().catch((err) => {
  console.error("Initial DB connection error:", err);
});

module.exports = macroeconomic_data_db;

```

FILE: backend\db\stockData.js

```

require("dotenv").config();
const mongoose = require("mongoose");

// Optional: Add better connection options
const connectionOptions = {
  useNewUrlParser: true,
  useUnifiedTopology: true,
  serverSelectionTimeoutMS: 30000, // Timeout after 30s if cannot connect
  socketTimeoutMS: 30000, // Prevents hanging socket operations
};

const stock_data_db = mongoose.createConnection(
  process.env.DATABASE_URL_STOCK_DATA,
  connectionOptions
);

// Monitor connection events
stock_data_db.on("connected", () => {
  console.log("Connected to stock DB:", stock_data_db.name);
});

stock_data_db.on("error", (err) => {
  console.error("Stock DB connection error:", err);
});

stock_data_db.on("disconnected", () => {
  console.warn("Stock DB disconnected");
});

// Optional: If using Mongoose 7+, you can await the promise
stock_data_db.asPromise().catch((err) => {
  console.error("Initial DB connection error:", err);
});

module.exports = stock_data_db;

```

FILE: backend\models\index.js

```

const Note = require("../Note");
const Profile = require("../Profile");
const { DGS1, DGS10 } = require("../TreasuryRates");

module.exports = { Note, Profile, DGS1, DGS10 };

```

FILE: backend\models\Note.js

```
const mongoose = require("mongoose");
const { investment_data_db } = require("../db/index");

// console.log("DB connection:", investment_data_db); // Should not be undefined

const noteSchema = new mongoose.Schema({
  title: {
    type: String,
    required: true,
  },
  company: {
    type: String,
  },
  asset: {
    type: String,
  },
  content: {
    type: String,
  },
  dateModified: {
    type: Date,
    required: false,
    default: Date.now,
  },
});

// Auto-update dateModified before save
// noteSchema.pre("save", function (next) {
//   this.dateModified = Date.now();
//   next();
// });

module.exports = investment_data_db.model("Note", noteSchema, "notes");
```

FILE: backend\models\Profile.js

```

const mongoose = require("mongoose");
const { stock_data_db } = require("../db/index");

const profileSchema = new mongoose.Schema(
  {
    ticker: String,
    name_damodaran: String,
    industry_damodaran: { type: String },
    sector_damodaran: { type: String },
    country_damodaran: { type: String },
    name_edgar: { type: String },
    industry_edgar: { type: String },
    industry_detail_edgar: { type: String },
    sector_yhoo: { type: String },
    industry_yhoo: { type: String },
    name_simfin: { type: String },
    sector_simfin: { type: String },
    industry_simfin: String,
    name_yhoo: String,
    name_seeking_alpha: { type: String },
    sector_seeking_alpha: { type: String },
    industry_seeking_alpha: { type: String },
  },
  {
    timestamps: true, // Automatically manage createdAt and updatedAt
    strict: false, // Allow extra fields that aren't in the schema
  }
);

// Auto-update dateModified before save
// noteSchema.pre("save", function (next) {
//   this.dateModified = Date.now();
//   next();
// });

module.exports = stock_data_db.model("Profile", profileSchema);

```

FILE: backend\models\TreasuryRates.js

```

const mongoose = require("mongoose");
const { macroeconomic_data_db } = require("../db/index");

const treasurydgs1Schema = new mongoose.Schema({
  data: Date,
  value: String,
});

// Auto-update dateModified before save
// noteSchema.pre("save", function (next) {
//   this.dateModified = Date.now();
//   next();
// });

const treasurydgs10Schema = new mongoose.Schema({
  data: Date,
  value: String,
});

// Auto-update dateModified before save
// noteSchema.pre("save", function (next) {
//   this.dateModified = Date.now();
//   next();
// });

const DGS1 = macroeconomic_data_db.model("DGS1", treasurydgs1Schema, "DGS1");
const DGS10 = macroeconomic_data_db.model(
  "DGS10",
  treasurydgs10Schema,
  "DGS10"
);

module.exports = { DGS1, DGS10 };

```


FILE: backend\routes\notes.js

```

const express = require("express");

const router = express.Router();
// const Note = require("../models/note");
const {
  getAllNotes,
  createOneNote,
  updateOneNote,
  deleteOneNote,
} = require("../controllers/notes");

// Get all
router.get("/", getAllNotes);

// Get one
router.get("/:id", getNote, (req, res) => {
  res.send(res.note);
});
// Creating one
router.post("/", createOneNote);

// Updating one
router.patch("/:id", getNote, updateOneNote);

// Deleting one
router.delete("/:id", getNote, deleteOneNote);

// Middleware - Get one
async function getNote(req, res, next) {
  let note;
  try {
    note = await Note.findById(req.params.id);
    if (note === null) {
      return res.status(404).json({ message: "Could not find note" }); // could not
find resources
    }
  } catch (err) {
    return res.status(500).json({ message: err.message });
  }
  res.note = note; // make res.note accessible to callback
  next();
}

module.exports = router;

```

FILE: backend\routes\profiles.js

```

const express = require("express");

const router = express.Router();
const { Profile } = require("../models/index");
const {
  getAllProfiles,
  getOneProfile,
  debug,
} = require("../controllers/profiles");

// Get all
router.get("/", getAllProfiles);

// Get one
router.get("/:ticker", getOneProfile);

// Get count and first one
router.get("/debug", debug);

// Get one
router.get("/:id", getProfile, (req, res) => {
  res.send(res.profile);
});

// Creating one
// router.post("/", async (req, res) => {
//   const note = new Note({
//     company: req.body.company,
//     asset: req.body.asset,
//     content: req.body.content,
//   });
//   try {
//     const newNote = await note.save();
//     res.status(201).json(note);
//   } catch (err) {
//     res.status(400).json({ message: err.message });
//   }
// });

// // Updating one
// router.patch("/:id", getNote, async (req, res) => {
//   if (req.body.company !== null) {
//     res.note.company = req.body.company;
//   }
//   if (req.body.asset !== null) {
//     res.note.asset = req.body.asset;
//   }
//   if (req.body.content !== null) {
//     res.note.content = req.body.content;
//   }

//   try {
//     const updatedNote = await res.note.save();
//     res.json(updatedNote);
//   } catch (err) {
//     res.status(400).json({ message: err.message });
//   }
// });

// // Deleting one

```

```

// router.delete("/:id", getNote, async (req, res) => {
//   try {
//     await res.note.deleteOne();
//     res.json({ message: "Deleted note" });
//   } catch (err) {
//     res.status(500).json({ message: err.message });
//   }
// });

// Middleware - Get one
async function getProfile(req, res, next) {
  let profile;
  try {
    profile = await Profile.findById(req.params.id);
    if (profile === null) {
      return res.status(404).json({ message: "Could not find profile" }); // could
not find resources
    }
  } catch (err) {
    return res.status(500).json({ message: err.message });
  }
  res.profile = profile; // make res.note accessible to callback
  next();
}

module.exports = router;

```

FILE: backend\routes\treasuryRates.js

```

const express = require("express");

const router = express.Router();
const { DGS1, DGS10 } = require("../models/TreasuryRates");

router.get("/", async (req, res) => {
  const dgs1 = DGS1.find({}).sort({ data: -1 }).limit(10).exec();
  const dgs10 = DGS10.find({}).sort({ data: -1 }).limit(10).exec();
  const results = await Promise.all([dgs1, dgs10]);
  res.send(results);
});

module.exports = router;

```

FILE: backend\server.js

```

require("dotenv").config();
const express = require("express");
const mongoose = require("mongoose");
const cors = require("cors");
const app = express();

// CORS setup
const allowedOrigins =
  process.env.NODE_ENV === "production"
    ? ["https://your-frontend-domain.com"]
    : ["http://localhost:5174"]; // Or other local environments

app.use(
  cors({
    origin: allowedOrigins,
    credentials: true,
  })
);

// // MongoDB connection
// const investment_data_db = mongoose.createConnection(
//   process.env.DATABASE_URL_INVESTMENT_DATA
// );

// investment_data_db
// .asPromise()
// .then(() => {
//   console.log("MongoDB connected to:", investment_data_db.name);
// })
// .catch((err) => {
//   console.error("MongoDB connection error:", err);
// });

// const stock_data_db = mongoose.createConnection(
//   process.env.DATABASE_URL_STOCK_DATA
// );

// stock_data_db
// .asPromise()
// .then(() => {
//   console.log("MongoDB connected to:", stock_data_db.name);
// })
// .catch((err) => {
//   console.error("MongoDB connection error:", err);
// });

// const macroeconomic_data_db = mongoose.createConnection(
//   process.env.DATABASE_URL_MACROECONOMIC_DATA
// );

// macroeconomic_data_db
// .asPromise()
// .then(() => {
//   console.log("MongoDB connected to:", macroeconomic_data_db.name);
// })
// .catch((err) => {
//   console.error("MongoDB connection error:", err);
// });

// Middleware

```

```
app.use(express.json()); // Accept JSON from body in POST requests

// Routes
const notesRouter = require("./routes/notes");
const profilesRouter = require("./routes/profiles");
const treasuryRatesRouter = require("./routes/treasuryRates");

app.use("/notes", notesRouter);
app.use("/profiles", profilesRouter);
app.use("/treasury-rates", treasuryRatesRouter);

// Error handling
app.use((req, res, next) => {
  res.status(404).json({ message: "Not Found" });
});

app.use((err, req, res, next) => {
  console.error(err.stack);
  res.status(500).json({ message: "Internal Server Error" });
});

// Start the server
app.listen(3000, () => console.log("Server started on port 3000"));
```

FILE: combine.js

```

// const map = new Map();
// console.log(map.set("dog", "chihuahua"));
// console.log(map.set("cat", "Siamese"));
// console.log(map.has("dog"));
// console.log(map.get("dog"));
// console.log(Array.from(map.keys()));
// console.log(Array.from(map.values()));
// console.log(Object.fromEntries(map));

// const { sp500, unemploymentRate } = require("./frontend/src/mockData");

// const mergeByDate = (a1, a2) => {const titleCamel = "initialClaims";
const titleCamel = "initialClaimsHere";
const title = titleCamel.split(/(?=[A-Z])/).join(" ");
console.log(title);
//   const map = new Map();

//   a1.forEach(({ date, value }) => {
//     if (!map.has(date)) map.set(date, { date });
//     map.get(date).dat1 = parseFloat(value);
//   });

//   a2.forEach(({ date, value }) => {
//     if (!map.has(date)) map.set(date, { date });
//     map.get(date).dat2 = parseFloat(value);
//   });

//   // Fill missing values with 0
//   return Array.from(map.values()).map((row) => ({
//     date: row.date,
//     dat1: row.dat1 ?? 0,
//     dat2: row.dat2 ?? 0,
//   }));
// };

// const merged = mergeByDate(sp500, unemploymentRate);

// console.log(merged);

// Slower and more verbose
// const merged = [];

// dat1.forEach(entry => {
//   const match = merged.find(m => m.date === entry.date);
//   if (match) {
//     match.dat1 = parseFloat(entry.value);
//   } else {
//     merged.push({ date: entry.date, dat1: parseFloat(entry.value), dat2: 0 });
//   }
// });

// dat2.forEach(entry => {
//   const match = merged.find(m => m.date === entry.date);
//   if (match) {
//     match.dat2 = parseFloat(entry.value);
//   } else {
//     merged.push({ date: entry.date, dat2: parseFloat(entry.value), dat1: 0 });
//   }
// });

```

```
// Robust array version
// -----
// const datasets = [
//   [
//     { date: "2019-02-01", value: "20" },
//     { date: "2019-02-02", value: "15" }
//   ],
//   [
//     { date: "2019-02-01", value: "50" },
//     { date: "2019-02-03", value: "35" }
//   ],
//   [
//     { date: "2019-02-02", value: "5" },
//     { date: "2019-02-04", value: "10" }
//   ]
// ];

// const mergeDatasetsByDate = (dataArrays) => {
//   const map = new Map();

//   dataArrays.forEach((dataset, index) => {
//     const key = `dat${index + 1}`;
//     dataset.forEach(({ date, value }) => {
//       if (!map.has(date)) map.set(date, { date });
//       map.get(date)[key] = parseFloat(value);
//     });
//   });

//   // Fill missing fields with 0
//   return Array.from(map.values()).map(entry => {
//     for (let i = 0; i < dataArrays.length; i++) {
//       const key = `dat${i + 1}`;
//       if (entry[key] === undefined) {
//         entry[key] = 0;
//       }
//     }
//     return entry;
//   });
// };

// const merged = mergeDatasetsByDate(datasets);

// console.log(merged);
```

FILE: frontend\next-env.d.ts

```
/// <reference types="next" />
/// <reference types="next/image-types/global" />

// NOTE: This file should not be edited
// see https://nextjs.org/docs/app/api-reference/config/typescript for more
information.
```

FILE: frontend\next.config.ts

```
import type { NextConfig } from "next";

const nextConfig: NextConfig = {
  /* config options here */
};

export default nextConfig;
```


FILE: frontend\src\app\about\page.tsx

```

import Image from "next/image";

export default function Home() {
  return (
    <div className="grid grid-rows-[20px_1fr_20px] items-center justify-items-center min-h-screen p-8 pb-20 gap-16 sm:p-20 font-[family-name:var(--font-geist-sans)]">
      <main className="flex flex-col gap-[32px] row-start-2 items-center sm:items-start">
        <Image
          className="dark:invert"
          src="/next.svg"
          alt="Next.js logo"
          width={180}
          height={38}
          priority
        />
        <ol className="list-inside list-decimal text-sm/6 text-center sm:text-left font-[family-name:var(--font-geist-mono)]">
          <li className="mb-2 tracking-[-.01em]">
            Get started by editing{" "}
            <code className="bg-black/[.05] dark:bg-white/[.06] px-1 py-0.5 rounded font-[family-name:var(--font-geist-mono)] font-semibold">
              src/app/page.tsx
            </code>
          </li>
          <li className="tracking-[-.01em]">
            Save and see your changes instantly.
          </li>
        </ol>

        <div className="flex gap-4 items-center flex-col sm:flex-row">
          <a
            className="rounded-full border border-solid border-transparent transition-colors flex items-center justify-center bg-foreground text-background gap-2 hover:bg-[#383838] dark:hover:bg-[#ccc] font-medium text-sm sm:text-base h-10 sm:h-12 px-4 sm:px-5 sm:w-auto"
            href="https://vercel.com/new?utm_source=create-next-app&utm_medium=appdir-template-tw&utm_campaign=create-next-app"
            target="_blank"
            rel="noopener noreferrer"
          >
            <Image
              className="dark:invert"
              src="/vercel.svg"
              alt="Vercel logomark"
              width={20}
              height={20}
            />
            Deploy now
          </a>
          <a
            className="rounded-full border border-solid border-black/[.08] dark:border-white/[.145] transition-colors flex items-center justify-center hover:bg-[#f2f2f2] dark:hover:bg-[#1a1a1a] hover:border-transparent font-medium text-sm sm:text-base h-10 sm:h-12 px-4 sm:px-5 w-full sm:w-auto md:w-[158px]"
            href="https://nextjs.org/docs?utm_source=create-next-app&utm_medium=appdir-template-tw&utm_campaign=create-next-app"
            target="_blank"
            rel="noopener noreferrer"
          >

```

```

    >
      Read our docs
    </a>
  </div>
</main>
<footer className="row-start-3 flex gap-[24px] flex-wrap items-center justify-
center">
  <a
    className="flex items-center gap-2 hover:underline hover:underline-offset-
4"
    href="https://nextjs.org/learn?utm_source=create-next-
app&utm_medium=appdir-template-tw&utm_campaign=create-next-app"
    target="_blank"
    rel="noopener noreferrer"
  >
    <Image
      aria-hidden
      src="/file.svg"
      alt="File icon"
      width={16}
      height={16}
    />
    Learn
  </a>
  <a
    className="flex items-center gap-2 hover:underline hover:underline-offset-
4"
    href="https://vercel.com/templates?framework=next.js&utm_source=create-
next-app&utm_medium=appdir-template-tw&utm_campaign=create-next-app"
    target="_blank"
    rel="noopener noreferrer"
  >
    <Image
      aria-hidden
      src="/window.svg"
      alt="Window icon"
      width={16}
      height={16}
    />
    Examples
  </a>
  <a
    className="flex items-center gap-2 hover:underline hover:underline-offset-
4"
    href="https://nextjs.org?utm_source=create-next-app&utm_medium=appdir-
template-tw&utm_campaign=create-next-app"
    target="_blank"
    rel="noopener noreferrer"
  >
    <Image
      aria-hidden
      src="/globe.svg"
      alt="Globe icon"
      width={16}
      height={16}
    />
    Go to nextjs.org →
  </a>
</footer>
</div>
);
}

```

FILE: frontend\src\app\data\page.tsx

```
import React from "react";

const Data = () => {
  return <div>Data</div>;
};

export default Data;
```

FILE: frontend\src\app\layout.tsx

```

"use client";
// import type { Metadata } from "next";
import { Geist, Geist_Mono } from "next/font/google";
import "@/globals.css";
import Header from "@/components/Header";
import Sidebar from "@/components/Sidebar";
import { useState } from "react";
import Footer from "@/components/Footer";
import LoadMockData from "@/components/LoadMockData";

const geistSans = Geist({
  variable: "--font-geist-sans",
  subsets: ["latin"],
});

const geistMono = Geist_Mono({
  variable: "--font-geist-mono",
  subsets: ["latin"],
});

// export const metadata: Metadata = {
//   title: "Investor Dashboard",
//   description: "A dashboard for investors to track their portfolios and notes.",
// };

export default function RootLayout({
  children,
}: Readonly<{
  children: React.ReactNode;
}>) {
  const [openSidebar, setOpenSidebar] = useState<boolean>(true);
  return (
    <html lang="en">
      <body
        className={` ${geistSans.variable} ${
          geistMono.variable
        } antialiased bg-stone-800 text-stone-100 gap-2 grid grid-rows-[auto_1fr]
min-h-screen font-[family-name:var(--font-geist-sans)] transition-all ease-in ${
          openSidebar
            ? "grid-cols-[140px_1fr]"
            : "grid-cols-[40px_1fr] bg-slate-900/20"
        }`
      >
        <Header />
        <Sidebar openSidebar={openSidebar} setOpenSidebar={setOpenSidebar} />
        <main
          // style={{ gridArea: "main" }}
          // style={{
            //   border: "2px solid black", // Border around the grid itself
            // }}
          className="grid-cols-1 col-end-3 row-start-2 row-end-5 mt-12 border border-
slate-800/50 rounded-md cols-start-2"
        >
          {children}
        </main>
        <Footer />
        <LoadMockData />
      </body>
    </html>
  );
}

```

```

    );
  }

```

FILE: frontend\src\app\macro\page.tsx

```

import { PriceChart } from "@components/charts/PriceChart";
import React from "react";

const Macro = () => {
  return (
    <div>
      <p>
        This is the Macroeconomics page and it will hold data about the larger
        economy.
      </p>
      <PriceChart
        titleCamel="Unemployment Rate"
        mockDataPath="unemploymentRate" // Use the dataset name inside mockData
        (e.g., 'initialClaims')
        valueKey="value" // The key you want to plot (e.g., "value")
        xKey="date" // This is optional, use it if the x-axis is date or other key
        barColor="rgb(20, 200, 240)"
      />
      { /* <GDPGrowthChart /> */ }
    </div>
  );
};

export default Macro;

```

FILE: frontend\src\app\page.tsx

```

"use client";
// import Image from "next/image";
import { PriceChart } from "@components/charts/PriceChart";
import Notepad from "@components/Notepad";
import Table from "@components/Table";
import CompanySearch from "@components/CompanySearch";
import Description from "@components/Profile";
import Notes from "@components/Notes";
import { EXJPUSChart } from "@components/EXJPUSChart";

export default function Home() {
  return (
    <div className="flex flex-wrap gap-2">
      <EXJPUSChart />
      <PriceChart
        titleCamel="Unemployment Rate"
        mockDataPath="unemploymentRate" // Use the dataset name inside mockData
        (e.g., 'initialClaims')
        valueKey="value" // The key you want to plot (e.g., "value")
        xKey="date" // This is optional, use it if the x-axis is date or other key
        barColor="rgb(20, 200, 240)"
      />

      <Notes />
      <CompanySearch />
      <Notepad />
      <PriceChart
        titleCamel="Initial Claims"
        mockDataPath="initialClaims" // Use the dataset name inside mockData (e.g.,
        'initialClaims')
        valueKey="value" // The key you want to plot (e.g., "value")
        xKey="date" // This is optional, use it if the x-axis is date or other key
        barColor="rgb(200, 100, 200)"
      />
      <Description />
      { /* <PriceChart /> */ }
      <Table />
    </div>
  );
}

```

FILE: frontend\src\app\payments\columns.tsx

```
"use client";

import { ColumnDef } from "@tanstack/react-table";

// This type is used to define the shape of our data.
// You can use a Zod schema here if you want.
export type Payment = {
  id: string;
  amount: number;
  status: "pending" | "processing" | "success" | "failed";
  email: string;
};

export const columns: ColumnDef<Payment>[] = [
  {
    accessorKey: "status",
    header: "Status",
  },
  {
    accessorKey: "email",
    header: "Email",
  },
  {
    accessorKey: "amount",
    header: "Amount",
  },
];
```

FILE: frontend\src\app\payments\data-table.tsx

```

"use client";

import {
  ColumnDef,
  flexRender,
  getCoreRowModel,
  useReactTable,
} from "@tanstack/react-table";

import {
  Table,
  TableBody,
  TableCell,
  TableHead,
  TableHeader,
  TableRow,
} from "@components/ui/table";

interface DataTableProps<TData, TValue> {
  columns: ColumnDef<TData, TValue>[];
  data: TData[];
}

export function DataTable<TData, TValue>({
  columns,
  data,
}: DataTableProps<TData, TValue>) {
  const table = useReactTable({
    data,
    columns,
    getCoreRowModel: getCoreRowModel(),
  });

  return (
    <div className="border border-border rounded-t-md overflow-hidden">
      <Table>
        <TableHeader className="bg-border">
          {table.getHeaderGroups().map((headerGroup) => (
            <TableRow key={headerGroup.id}>
              {headerGroup.headers.map((header) => {
                return (
                  <TableHead key={header.id}>
                    {header.isPlaceholder
                      ? null
                      : flexRender(
                        header.column.columnDef.header,
                        header.getContext()
                      )}
                </TableHead>
              )}
            </TableRow>
          ))}
        </TableHeader>
        <TableBody>
          {table.getRowModel().rows?.length ? (
            table.getRowModel().rows.map((row) => (
              <TableRow
                key={row.id}
                data-state={row.getIsSelected() && "selected"}
              >

```



```

>
    {row.getVisibleCells().map((cell) => (
      <TableCell key={cell.id}>
        {flexRender(cell.column.columnDef.cell, cell.getContext())}
      </TableCell>
    ))}
  </TableRow>
))
) : (
  <TableRow>
    <TableCell colspan={columns.length} className="h-24 text-center">
      No results.
    </TableCell>
  </TableRow>
)}
</TableBody>
</Table>
</div>
);
}

```

FILE: frontend\src\components\charts\PriceChart.tsx

```

"use client";

import { Bar, BarChart, CartesianGrid, XAxis } from "recharts";

import {
  ChartConfig,
  ChartContainer,
  ChartTooltip,
  ChartTooltipContent,
} from "@components/ui/chart";
import Pane from "../Pane";
import { useEffect, useState } from "react";

// interface IChartData {
//   _id: {
//     $oid: string;
//   };
//   date: string;
//   value: number | string;
// }

// const titleCamel = "initialClaims";

// const chartData = [
//   { month: "January", desktop: 186, mobile: 80 },
//   { month: "February", desktop: 305, mobile: 200 },
//   { month: "March", desktop: 237, mobile: 120 },
//   { month: "April", desktop: 73, mobile: 190 },
//   { month: "May", desktop: 209, mobile: 130 },
//   { month: "June", desktop: 214, mobile: 140 },
// ];

interface IChart<T> {
  titleCamel: string;
  mockDataPath: string;
  chartData?: T[];
  valueKey: keyof T;
  xKey?: keyof T;
  barColor?: string;
}

export function PriceChart<T extends Record<string, unknown>>({
  titleCamel,
  mockDataPath,
  chartData: propData,
  valueKey,
  xKey = "date" as keyof T,
  barColor = "rgb(100,200,100)",
}: IChart<T>) {
  const [chartData, setChartData] = useState<T[]>([]);

  const title = titleCamel
    ? titleCamel
      .split(/(?=[A-Z])/)
      .join(" ")
      .replace(/^../, (s) => s.toUpperCase())
    : "";

  useEffect(() => {
    if (propData && Array.isArray(propData)) {

```

```

const normalized = propData.map((d) => ({
  ...d,
  [valueKey]: Number(d[valueKey]),
}));
setChartData(normalized);
return;
}

import("@/mockData")
.then((module) => {
  const data = module[mockDataPath] as T[] | undefined;
  if (!data || !Array.isArray(data)) {
    console.error("Mock data not found or invalid:", mockDataPath);
    return;
  }

  const normalized = data.map((d) => ({
    ...d,
    [valueKey]: Number(d[valueKey]),
  }));
  setChartData(normalized);
})
.catch((err) => {
  console.error("Failed to load mock data:", err);
});
}, [mockDataPath, propData, valueKey]);

const chartConfig: ChartConfig = {
  InitialClaims: {
    label: title,
    color: "#60a5fa",
  },
};
return (
  <Pane>
    <h1>{title}</h1>
    <ChartContainer config={chartConfig} className="max-w-full h-[200px]">
      <BarChart accessibilityLayer data={chartData}>
        <CartesianGrid vertical={false} />
        <XAxis
          dataKey={xKey as string}
          tickLine={false}
          tickMargin={10}
          axisLine={false}
          tickFormatter={(value) => {
            const date = new Date(value);
            return date.toLocaleDateString("en-US", {
              month: "short",
              day: "numeric",
            });
          }}
        />
        <ChartTooltip content={<ChartTooltipContent />} />
        <Bar dataKey={valueKey as string} fill={barColor} radius={4} />
      </BarChart>
    </ChartContainer>
  </Pane>
);
}

```

FILE: frontend\src\components\charts\PriceChart1.tsx

```

"use client";

import { Bar, BarChart, CartesianGrid, XAxis } from "recharts";

import {
  ChartConfig,
  ChartContainer,
  ChartTooltip,
  ChartTooltipContent,
} from "@components/ui/chart";
import Pane from "../Pane";
import { SP500, unemploymentRate } from "../../mockData";

interface IChartData {
  _id: string;
  date: string;
  value: number | string;
}

const mergeByDate = (a1: IChartData[], a2: IChartData[]) => {
  const map = new Map();

  a1.forEach(({ date, value }) => {
    if (!map.has(date)) map.set(date, { date });
    map.get(date).sp500 = value as number;
  });

  a2.forEach(({ date, value }) => {
    if (!map.has(date)) map.set(date, { date });
    map.get(date).unemploymentRate = parseFloat(value as string);
  });

  // Fill missing values with 0
  return Array.from(map.values()).map((row) => ({
    date: row.date,
    sp500: row.sp500 ?? 0,
    unemploymentRate: row.unemploymentRate ?? 0,
  }));
};

const chartData = mergeByDate(SP500, unemploymentRate);

// const chartData = [
//   { month: "January", desktop: 186, mobile: 80 },
//   { month: "February", desktop: 305, mobile: 200 },
//   { month: "March", desktop: 237, mobile: 120 },
//   { month: "April", desktop: 73, mobile: 190 },
//   { month: "May", desktop: 209, mobile: 130 },
//   { month: "June", desktop: 214, mobile: 140 },
// ];

const chartConfig = {
  sp500: {
    label: "S&P 500",
    color: "#2563eb",
  },
  unemploymentRate: {
    label: "Unemployment Rate",
    color: "#60a5fa",
  },
};

```

```

} satisfies ChartConfig;

export function PriceChart() {
  return (
    <Pane>
      <h1>S&P 500 vs. Unemployment Rate</h1>
      <ChartContainer config={chartConfig} className="max-w-full h-[200px]">
        <BarChart accessibilityLayer data={chartData}>
          <CartesianGrid vertical={false} />
          <XAxis
            dataKey="date"
            tickLine={false}
            tickMargin={10}
            axisLine={false}
            // tickFormatter={(value) => value.slice(0, 3)}
            tickFormatter={(value) => {
              const date = new Date(value);
              return date.toLocaleDateString("en-US", {
                month: "short",
                day: "numeric",
              });
            }}
          />
          <ChartTooltip content={<ChartTooltipContent />} />
          <Bar dataKey="sp500" fill="rgb(100, 100, 200)" radius={4} />
          <Bar dataKey="unemploymentRate" fill="rgb(100,200,100)" radius={4} />
        </BarChart>
      </ChartContainer>
    </Pane>
  );
}

```

FILE: frontend\src\components\CompanySearch.tsx

```

"use client";
import React from "react";
import { useCompanySearch } from "@/hooks/useCompanySearch";
import { useStore } from "@/store";
import Pane from "../Pane";

const CompanySearch = () => {
  const { inputValue, setInputValue, results, loading } = useCompanySearch();
  const setTicker = useStore((state) => state.setTicker);
  const setProfile = useStore((state) => state.setProfile);

  return (
    <Pane>
      <form
        onSubmit={(e) => e.preventDefault()}
        className="flex flex-col gap-2"
      >
        <label htmlFor="ticker">Choose a ticker:</label>
        <div className="flex gap-2">
          <input
            type="text"
            value={inputValue}
            onChange={(e) => setInputValue(e.target.value)}
            className="px-4 py-1 rounded outline-1 outline-border max-w-[calc(100%-
            _4rem)] text-tertiary"
            placeholder="Search"
          />
        </div>
      </form>

      {loading ? (
        <p className="mt-2">Loading...</p>
      ) : (
        inputValue &&
        results && (
          <ul className="flex flex-wrap gap-2 mt-4 p-2 border border-border rounded
            max-h-[160px] overflow-auto">
            {results?.map((profile) => (
              <li
                key={profile.ticker}
                className="px-2 py-1 bg-border hover:bg-border/90 rounded duration-
                200 ease-in-out cursor-pointer"
                onClick={() => {
                  setTicker(profile.ticker);
                  setProfile(profile);
                }}
              >
                {profile.ticker}
              </li>
            ))}
          </ul>
        )
      )}
    </Pane>
  );
};

export default CompanySearch;

```

FILE: frontend\src\components\EXJPUSChart.tsx

```

"use client";

import { Bar, BarChart, CartesianGrid, XAxis, YAxis } from "recharts";

import {
  ChartConfig,
  ChartContainer,
  ChartTooltip,
  ChartTooltipContent,
} from "@components/ui/chart";
import Pane from "../Pane";
import { EXJPUS } from "../mockData";

interface IChartData {
  _id: { $oid: string };
  date: string;
  EXJPUS: number;
}

const modifiedData = EXJPUS.map((item) => ({
  _id: item._id,
  date: item.date,
  EXJPUS: parseFloat(item.value),
})).filter((item) => item.date > "2023-01-01");
const chartData: IChartData[] = modifiedData;

const maxVal = Math.max(...chartData.map((item) => item.EXJPUS));
const minVal = Math.min(...chartData.map((item) => item.EXJPUS));

// const chartData = [
//   { month: "January", desktop: 186, mobile: 80 },
//   { month: "February", desktop: 305, mobile: 200 },
//   { month: "March", desktop: 237, mobile: 120 },
//   { month: "April", desktop: 73, mobile: 190 },
//   { month: "May", desktop: 209, mobile: 130 },
//   { month: "June", desktop: 214, mobile: 140 },
// ];

const chartConfig = {
  EXJPUS: {
    label: "JPY/USD",
    color: "#2563eb",
  },
} satisfies ChartConfig;

export function EXJPUSChart() {
  return (
    <Pane>
      <h1>JPY/USD Exchange Rate</h1>
      <ChartContainer config={chartConfig} className="max-w-full h-[200px]">
        <BarChart
          accessibilityLayer
          data={chartData}
          margin={{ top: 10, right: 10, bottom: 20, left: 10 }} // 🖱️ Add margin to
chart
        >
          <CartesianGrid vertical={false} />
          <XAxis
            dataKey="date"
            tickLine={false}

```

```

tickMargin={10}
axisLine={false}
tick={({props}) => {
  const { x, y, payload } = props;
  const date = new Date(payload.value);
  const formattedDate = date.toLocaleDateString("en-US", {
    year: "numeric",
    month: "short",
    day: "numeric",
  });
  return (
    <text
      x={x}
      y={y}
      dy={3}
      textAnchor="end"
      fontSize={10}
      transform={`rotate(-35, ${x}, ${y})`}
      // style={{ fontSize: "12px" }}
    >
      {formattedDate}
    </text>
  );
}}
// tickFormatter={(value) => value.slice(0, 3)}
tickFormatter={(value) => {
  const date = new Date(value);
  return date.toLocaleDateString("en-US", {
    year: "numeric",
    month: "short",
    day: "numeric",
  });
}}
/>
<YAxis
  domain={[minVal * 0.98, maxVal * 1.02]}
  tickLine={false}
  tickMargin={10}
  axisLine={false}
  tickFormatter={(value) => {
    return value.toLocaleString("en-US", {
      style: "currency",
      currency: "JPY",
      minimumFractionDigits: 0,
    });
  }}
/>
<ChartTooltip content={<ChartTooltipContent />} />
<Bar
  dataKey="EXJPUS"
  radius={4}
  fill="#2563eb"
  data={chartData.map((entry) => {
    // console.log(entry.date);
    return {
      ...entry,
      fill:
        new Date(entry.date) < new Date("2024-05-01")
          ? "#ff4d4f"
          : "#2563eb",
    };
  })}
/>
</BarChart>

```



```
        </ChartContainer>  
      </Pane>  
    );  
  }
```

FILE: frontend\src\components\Footer.tsx

```

import React from "react";
import Image from "next/image";

const Footer = () => {
  return (
    <footer
      // style={{ gridArea: "footer" }}
      className="flex justify-center items-center gap-6 col-span-1 col-start-2 row-
start-5 row-end-6 py-4 bg-border/20 border-x border-t border-border/60 rounded-t"
    >
      <a
        className="flex items-center gap-2 hover:underline hover:underline-offset-4"
        href="https://nextjs.org/learn?utm_source=create-next-app&utm_medium=appdir-
template-tw&utm_campaign=create-next-app"
        target="_blank"
        rel="noopener noreferrer"
      >
        <Image
          aria-hidden
          src="/file.svg"
          alt="File icon"
          width={16}
          height={16}
        />
        Learn
      </a>
      <a
        className="flex items-center gap-2 hover:underline hover:underline-offset-4"
        href="https://vercel.com/templates?framework=next.js&utm_source=create-next-
app&utm_medium=appdir-template-tw&utm_campaign=create-next-app"
        target="_blank"
        rel="noopener noreferrer"
      >
        <Image
          aria-hidden
          src="/window.svg"
          alt="Window icon"
          width={16}
          height={16}
        />
        Examples
      </a>
      <a
        className="flex items-center gap-2 hover:underline hover:underline-offset-4"
        href="https://nextjs.org?utm_source=create-next-app&utm_medium=appdir-
template-tw&utm_campaign=create-next-app"
        target="_blank"
        rel="noopener noreferrer"
      >
        <Image
          aria-hidden
          src="/globe.svg"
          alt="Globe icon"
          width={16}
          height={16}
        />
        Go to nextjs.org →
      </a>
    </footer>
  );
};

```

```
};

export default Footer;
```

FILE: frontend\src\components\Header.tsx

```
import React from "react";
import Link from "next/link";

const Header = () => {
  return (
    <header
      // style={{ gridArea: "header" }}
      className="z-10 fixed col-span-2 row-start-1 row-end-2 w-screen"
    >
      <div className="flex items-center bg-slate-800/90 px-4 border-b border-
border/60 h-[46px]">
        <Link href="/" className="hover:text-green-200 text-2xl">
          Investor Dashboard
        </Link>
      </div>
    </header>
  );
};

export default Header;
```

FILE: frontend\src\components\LoadMockData.tsx

```
import { useEffect } from "react";
import { useStore } from "@store";
import { notes as mockNotes, profiles as mockProfiles } from "@mockData";

const LoadMockData = () => {
  const setNotes = useStore((state) => state.setNotes);
  const setProfiles = useStore((state) => state.setProfiles);

  useEffect(() => {
    if (process.env.NEXT_PUBLIC_ENVIRONMENT === "production") {
      setNotes(mockNotes);
      setProfiles(mockProfiles);
    }
  }, [setNotes, setProfiles]);

  return null; // doesn't render anything
};

export default LoadMockData;
```

FILE: frontend\src\components\Notepad.tsx

```
import React from "react";
import NotepadForm from "../NotepadForm";

const Notepad = () => {
  return <NotepadForm />;
};

export default Notepad;
```

FILE: frontend\src\components\NotepadForm.tsx

```

// import axios from "axios";
import React, { FormEvent } from "react";
import Pane from "../Pane";

const NotepadForm = () => {
  const tickerRef = React.useRef<HTMLInputElement>(null);
  const assetRef = React.useRef<HTMLInputElement>(null);
  const titleRef = React.useRef<HTMLInputElement>(null);
  // const [note, setNote] = useState("");

  const handleSubmit = async (e: FormEvent<HTMLElement>) => {
    e.preventDefault();

    // await axios.post("http://localhost:3000/notes", {
    //   title: titleRef.current?.value,
    //   ticker: tickerRef.current?.value,
    //   asset: assetRef.current?.value,
    //   note: note,
    // });
  };

  return (
    <Pane>
      <form
        onSubmit={(e) => handleSubmit(e)}
        className="flex flex-col gap-2 mx-auto rounded max-w-[400px]"
      >
        { /* <label htmlFor="note" id="note">
          Note:
        </label> */ }
        <div className="flex gap-2">
          <input
            type="text"
            ref={titleRef}
            placeholder="Title"
            className="bg-emerald-900/20 px-4 py-1 rounded outline-1 outline-border
w-1/2 text-slate-200 text-tertiary"
          />
          <input
            type="text"
            ref={tickerRef}
            placeholder="Ticker"
            className="bg-emerald-900/20 px-4 py-1 rounded outline-1 outline-border
w-1/2 text-slate-200 text-tertiary"
          />
          <input
            type="text"
            ref={assetRef}
            placeholder="Asset"
            className="bg-emerald-900/20 px-4 py-1 rounded outline-1 outline-border
w-1/2 text-slate-200 text-tertiary"
          />
        </div>

        <textarea
          className="bg-emerald-900/20 p-2 ring-border rounded-sm outline-none ring-1
max-w-[400px] h-[160px] text-slate-200"
          // value={note}
          name="note"
          placeholder="Add note"
          // onChange={(e) => setNote(e.target.value)}

```

```

    ></textarea>
    <button
      // onClick={() => setNote(note)}
      className="py-1 bg-border hover:bg-border/90 rounded-sm text-slate-900
cursor-pointer"
    >
      Save
    </button>
  </form>
</Pane>
);
};

export default NotepadForm;

```

FILE: frontend\src\components\Notes.tsx

```

"use client";
import { useStore } from "@store";
import Pane from "../Pane";

const Notes = () => {
  const notes = useStore((state) => state.notes);

  return (
    <Pane>
      {/* Sticky header */}
      <div className="top-0 z-10 sticky">
        <h1 className="border-slate-100/20 border-b w-full text-xl">Notes:</h1>
      </div>

      {/* Scrollable notes list */}
      <div className="flex-1 mt-2 overflow-auto hide-scrollbar">
        {notes?.map((note) => {
          return (
            <div
              className="my-4 px-2 py-4 border border-border-60 rounded"
              key={note._id}
            >
              <div className="mr-2 text-slate-400 text-xs">
                {note.dateModified.toString()}
              </div>
              <span className="mr-2 text-border text-xs">
                {note.ticker || note.asset || note.title || ""}
              </span>
              {note.content}
            </div>
          );
        })}
      </div>
    </Pane>
  );
};

export default Notes;

```

FILE: frontend\src\components\Pane.tsx

```
import React from "react";

const Pane: React.FC<{ children: React.ReactNode }> = ({ children }) => {
  return (
    <div className="relative flex flex-col bg-slate-800 p-6 border border-slate-600
w-[300px] md:w-[500px] lg:w-[600px] min-w-[200px] max-w-[400px] h-full max-h-[300px]
overflow-auto hide-scrollbar">
      {children}
    </div>
  );
};

export default Pane;
```

FILE: frontend\src\components\Profile.tsx

```
"use client";
import { useStore } from "@/store";
import React from "react";
import Pane from "../Pane";

const Profile = () => {
  const profile = useStore((state) => state.profile);

  return (
    <Pane>
      <h1 className="border-slate-300/20 border-b w-full text-xl">Profile:</h1>

      <h2 className="mt-2 text-emerald-300 text-sm">Name Damodaran</h2>
      <div>{profile ? profile.name_damodaran : "No profile selected"}</div>
      <h2 className="mt-2 text-emerald-300 text-sm">Industry Damodaran</h2>
      <div>{profile ? profile.industry_damodaran : "No profile selected"}</div>
      <h2 className="mt-2 text-emerald-300 text-sm">Sector Damodaran</h2>
      <div>{profile ? profile.sector_damodaran : "No profile selected"}</div>
    </Pane>
  );
};

export default Profile;
```

FILE: frontend\src\components\Ratios.tsx

```
import React from "react";

const Ratios = () => {
  return <div>Ratios</div>;
};

export default Ratios;
```

FILE: frontend\src\components\Sidebar.tsx

```

"use client";
import React from "react";
import Link from "next/link";
import { FiAlignJustify } from "react-icons/fi";
import { FaRegQuestionCircle } from "react-icons/fa";
import { IoStatsChartSharp } from "react-icons/io5";
import { CiViewTable } from "react-icons/ci";

interface ISidebar {
  openSidebar: boolean;
  setOpenSidebar: (c: boolean) => void;
}

const Sidebar: React.FC<ISidebar> = ({ openSidebar, setOpenSidebar }) => {
  return (
    <aside
      // style={{ gridArea: "sidebar" }}
      className="relative row-start-2 row-end-5 mt-12 min-h-[calc(100vh-36px)]"
    >
      <div
        className={`top-14 fixed w-36 border-r border-y border-border/60 bg-border/20
rounded-tr overflow-hidden transition-all duration-50 ease-out h-[calc(100vh-40px)]
z-10 flex flex-col items-center justify-start gap-4 shadow-lg
${!openSidebar && "w-[42px] h-[40px] rounded-br-xl border-b"}`}
      >
        <div className="relative w-full">
          <FiAlignJustify
            className="top-2 right-2 absolute hover:text-green-200 text-xl cursor-
pointer"
            onClick={() => setOpenSidebar(!openSidebar)}
          />
        </div>

        <div
          className={`flex flex-col gap-4 pt-16 w-full
            ${!openSidebar ? "px-1" : "pl-4"}`}
        >
          <Link href="/about" className="hover:text-green-300">
            <div className="flex items-center gap-1">
              <FaRegQuestionCircle className="mx-2 text-xl" />
              {openSidebar && <p>About</p>}
            </div>
          </Link>
          <Link href="/macro" className="hover:text-green-300">
            <div className="flex items-center gap-1">
              <IoStatsChartSharp className="mx-2 text-xl" />
              {openSidebar && <p>Macro</p>}
            </div>
          </Link>
          <Link href="/data" className="hover:text-green-300">
            <div className="flex items-center gap-1">
              <CiViewTable className="mx-2 text-xl" />
              {openSidebar && <p>Data</p>}
            </div>
          </Link>
        </div>
      </div>
    </aside>
  );
};

```



```
export default Sidebar;
```

FILE: frontend\src\components\Table.tsx

```
// "use client";

import { useEffect, useState } from "react";
import { Payment, columns } from "../app/payments/columns";
import { DataTable } from "../app/payments/data-table";
import Pane from "../Pane";

export default function DemoPage() {
  const [data, setData] = useState<Payment[] | null>([]);
  async function getData(): Promise<Payment[]> {
    // Fetch data from your API here.
    return [
      {
        id: "728ed52f",
        amount: 100,
        status: "pending",
        email: "m@example.com",
      },
      {
        id: "489e1d42",
        amount: 125,
        status: "processing",
        email: "example@gmail.com",
      },
      // ...
    ];
  }

  useEffect(() => {
    const fetchData = async () => {
      const result: Payment[] = await getData();
      setData(result);
    };
    fetchData();
  }, []);

  return (
    <Pane>
      <div className="mx-auto py-10 container">
        <DataTable columns={columns} data={data || []} />
      </div>
    </Pane>
  );
}
```

FILE: frontend\src\components\ui\card.tsx

```

import * as React from "react"

import { cn } from "@/lib/utils"

function Card({ className, ...props }: React.ComponentProps<"div">) {
  return (
    <div
      data-slot="card"
      className={cn(
        "bg-card text-card-foreground flex flex-col gap-6 rounded-xl border py-6
shadow-sm",
        className
      )}
      {...props}
    />
  )
}

function CardHeader({ className, ...props }: React.ComponentProps<"div">) {
  return (
    <div
      data-slot="card-header"
      className={cn(
        "@container/card-header grid auto-rows-min grid-rows-[auto_auto] items-start
gap-1.5 px-6 has-data-[slot=card-action]:grid-cols-[1fr_auto] [.border-b]:pb-6",
        className
      )}
      {...props}
    />
  )
}

function CardTitle({ className, ...props }: React.ComponentProps<"div">) {
  return (
    <div
      data-slot="card-title"
      className={cn("leading-none font-semibold", className)}
      {...props}
    />
  )
}

function CardDescription({ className, ...props }: React.ComponentProps<"div">) {
  return (
    <div
      data-slot="card-description"
      className={cn("text-muted-foreground text-sm", className)}
      {...props}
    />
  )
}

function CardAction({ className, ...props }: React.ComponentProps<"div">) {
  return (
    <div
      data-slot="card-action"
      className={cn(
        "col-start-2 row-span-2 row-start-1 self-start justify-self-end",
        className
      )}
    />
  )
}

```

```

        {...props}
      />
    )
  }

function CardContent({ className, ...props }: React.ComponentProps<"div">) {
  return (
    <div
      data-slot="card-content"
      className={cn("px-6", className)}
      {...props}
    />
  )
}

function CardFooter({ className, ...props }: React.ComponentProps<"div">) {
  return (
    <div
      data-slot="card-footer"
      className={cn("flex items-center px-6 [.border-t]:pt-6", className)}
      {...props}
    />
  )
}

export {
  Card,
  CardHeader,
  CardFooter,
  CardTitle,
  CardAction,
  CardDescription,
  CardContent,
}

```

FILE: frontend\src\components\ui\chart.tsx

```
"use client";

import * as React from "react";
import * as RechartsPrimitive from "recharts";

import { cn } from "@lib/utils";

// Format: { THEME_NAME: CSS_SELECTOR }
const THEMES = { light: "", dark: ".dark" } as const;

export type ChartConfig = {
  [k in string]: {
    label?: React.ReactNode;
    icon?: React.ComponentType;
  } & (
    | { color?: string; theme?: never }
    | { color?: never; theme: Record<keyof typeof THEMES, string> }
  );
};

type ChartContextProps = {
  config: ChartConfig;
};

const ChartContext = React.createContext<ChartContextProps | null>(null);

function useChart() {
  const context = React.useContext(ChartContext);

  if (!context) {
    throw new Error("useChart must be used within a <ChartContainer />");
  }

  return context;
}

function ChartContainer({
  id,
  className,
  children,
  config,
  ...props
}: React.ComponentProps<"div"> & {
  config: ChartConfig;
  children: React.ComponentProps<
    typeof RechartsPrimitive.ResponsiveContainer
  >["children"];
}) {
  const uniqueId = React.useId();
  const chartId = `chart-${id || uniqueId.replace(/:/g, "")}`;

  return (
    <ChartContext.Provider value={{ config }}>
      <div
        data-slot="chart"
        data-chart={chartId}
        className={cn(
          "[&_recharts-cartesian-axis-tick_text]:fill-muted-foreground [&_recharts-cartesian-grid_line[stroke='#ccc']]:stroke-border/50 [&_recharts-curve.recharts-tooltip-cursor]:stroke-border [&_recharts-polar-grid_[stroke='#ccc']]:stroke-border"
        )}
      >
        {children}
      </div>
    </ChartContext.Provider>
  );
}
```

```

[&_.recharts-radial-bar-background-sector]:fill-muted [&_.recharts-
rectangle.recharts-tooltip-cursor]:fill-emerald-500/40 [&_.recharts-reference-
line_[stroke='#ccc']]:stroke-border flex aspect-video justify-center text-xs
[&_.recharts-dot[stroke='#fff']]:stroke-transparent [&_.recharts-layer]:outline-
hidden [&_.recharts-sector]:outline-hidden [&_.recharts-
sector[stroke='#fff']]:stroke-transparent [&_.recharts-surface]:outline-hidden",
    className
  )}
  {...props}
}
>
<ChartStyle id={chartId} config={config} />
<RechartsPrimitive.ResponsiveContainer>
  {children}
</RechartsPrimitive.ResponsiveContainer>
</div>
</ChartContext.Provider>
);
}

const ChartStyle = ({ id, config }: { id: string; config: ChartConfig }) => {
  const colorConfig = Object.entries(config).filter(
    ([, config]) => config.theme || config.color
  );

  if (!colorConfig.length) {
    return null;
  }

  return (
    <style
      dangerouslySetInnerHTML={{
        __html: Object.entries(THEMES)
          .map(
            ([theme, prefix]) => `
${prefix} [data-chart=${id}] {
${colorConfig
  .map(([key, itemConfig]) => {
    const color =
      itemConfig.theme?.[theme as keyof typeof itemConfig.theme] ||
      itemConfig.color;
    return color ? `  --color-${key}: ${color};` : null;
  })
  .join("\n")}
    `
      }
    )
    .join("\n"),
    )
  );
};

```

```
const ChartTooltip = RechartsPrimitive.Tooltip;
```

```
function ChartTooltipContent({
  active,
  payload,
  className,
  indicator = "dot",
  hideLabel = false,
  hideIndicator = false,
  label,
  labelFormatter,
  labelClassName,

```

```

    formatter,
    color,
    nameKey,
    labelKey,
  }: React.ComponentProps<typeof RechartsPrimitive.Tooltip> &
  React.ComponentProps<"div"> & {
    hideLabel?: boolean;
    hideIndicator?: boolean;
    indicator?: "line" | "dot" | "dashed";
    nameKey?: string;
    labelKey?: string;
  }) {
    const { config } = useChart();

    const tooltipLabel = React.useMemo(() => {
      if (hideLabel || !payload?.length) {
        return null;
      }

      const [item] = payload;
      const key = `${labelKey || item?.dataKey || item?.name || "value"}`;
      const itemConfig = getPayloadConfigFromPayload(config, item, key);
      const value =
        !labelKey && typeof label === "string"
          ? config[label as keyof typeof config]?.label || label
          : itemConfig?.label;

      if (labelFormatter) {
        return (
          <div className={cn("font-medium", labelClassName)}>
            {labelFormatter(value, payload)}
          </div>
        );
      }

      if (!value) {
        return null;
      }

      return (
        <div className={cn("font-medium text-slate-600", labelClassName)}>
          {value}
        </div>
      );
    }, [
      label,
      labelFormatter,
      payload,
      hideLabel,
      labelClassName,
      config,
      labelKey,
    ]);

    if (!active || !payload?.length) {
      return null;
    }

    const nestLabel = payload.length === 1 && indicator !== "dot";

    return (
      <div
        className={cn(
          "border-border/50 bg-green-200 grid min-w-[8rem] items-start gap-1.5 rounded-

```

```

lg border px-2.5 py-1.5 text-xs shadow-xl",
  className
)}
>
{!nestLabel ? tooltipLabel : null}
<div className="gap-1.5 grid">
  {payload.map((item, index) => {
    const key = `${nameKey || item.name || item.dataKey || "value"}`;
    const itemConfig = getPayloadConfigFromPayload(config, item, key);
    const indicatorColor = color || item.payload.fill || item.color;

    return (
      <div
        key={item.dataKey}
        className={cn(
          "[&svg]:text-muted-foreground flex w-full flex-wrap items-stretch
gap-2 [&svg]:h-2.5 [&svg]:w-2.5",
          indicator === "dot" && "items-center"
        )}
      >
        {formatter && item?.value !== undefined && item.name ? (
          formatter(item.value, item.name, item, index, item.payload)
        ) : (
          <>
            {itemConfig?.icon ? (
              <itemConfig.icon />
            ) : (
              !hideIndicator && (
                <div
                  className={cn(
                    "shrink-0 rounded-[2px] border-(--color-border) bg-blue-
400",
                    {
                      "h-2.5 w-2.5": indicator === "dot",
                      "w-1": indicator === "line",
                      "w-0 border-[1.5px] border-dashed bg-transparent":
                        indicator === "dashed",
                      "my-0.5": nestLabel && indicator === "dashed",
                    }
                  )}
                  style={
                    {
                      "--color-bg": indicatorColor,
                      "--color-border": indicatorColor,
                    } as React.CSSProperties
                  }
                />
              )
            )}
            <div
              className={cn(
                "flex flex-1 justify-between leading-none",
                nestLabel ? "items-end" : "items-center"
              )}
            >
              <div className="gap-1.5 grid">
                {nestLabel ? tooltipLabel : null}
                <span className="text-muted-foreground">
                  {itemConfig?.label || item.name}
                </span>
              </div>
              {item.value && (
                <span className="font-mono font-medium tabular-nums text-
foreground">

```

```

        {item.value.toLocaleString()}
      </span>
    )}
  </div>
</>
)}
</div>
);
}}
</div>
</div>
);
}

const ChartLegend = RechartsPrimitive.Legend;

function ChartLegendContent({
  className,
  hideIcon = false,
  payload,
  verticalAlign = "bottom",
  nameKey,
}: React.ComponentProps<"div"> &
Pick<RechartsPrimitive.LegendProps, "payload" | "verticalAlign"> & {
  hideIcon?: boolean;
  nameKey?: string;
}) {
  const { config } = useChart();

  if (!payload?.length) {
    return null;
  }

  return (
    <div
      className={cn(
        "flex items-center justify-center gap-4",
        verticalAlign === "top" ? "pb-3" : "pt-3",
        className
      )}
    >
      {payload.map((item) => {
        const key = `${nameKey || item.dataKey || "value"}`;
        const itemConfig = getPayloadConfigFromPayload(config, item, key);

        return (
          <div
            key={item.value}
            className={cn(
              "[&svg]:text-muted-foreground flex items-center gap-1.5 [&svg]:h-3
[&svg]:w-3"
            )}
          >
            {itemConfig?.icon && !hideIcon ? (
              <itemConfig.icon />
            ) : (
              <div
                className="rounded-[2px] w-2 h-2 shrink-0"
                style={{
                  backgroundColor: item.color,
                }}
              />
            )}
            {itemConfig?.label}
          </div>
        )
      })}
    </div>
  );
}

```



```

        </div>
    );
    })}
</div>
);
}

// Helper to extract item config from a payload.
function getPayloadConfigFromPayload(
    config: ChartConfig,
    payload: unknown,
    key: string
) {
    if (typeof payload !== "object" || payload === null) {
        return undefined;
    }

    const payloadPayload =
        "payload" in payload &&
        typeof payload.payload === "object" &&
        payload.payload !== null
        ? payload.payload
        : undefined;

    let configLabelKey: string = key;

    if (
        key in payload &&
        typeof payload[key as keyof typeof payload] === "string"
    ) {
        configLabelKey = payload[key as keyof typeof payload] as string;
    } else if (
        payloadPayload &&
        key in payloadPayload &&
        typeof payloadPayload[key as keyof typeof payloadPayload] === "string"
    ) {
        configLabelKey = payloadPayload[
            key as keyof typeof payloadPayload
        ] as string;
    }

    return configLabelKey in config
        ? config[configLabelKey]
        : config[key as keyof typeof config];
}

export {
    ChartContainer,
    ChartTooltip,
    ChartTooltipContent,
    ChartLegend,
    ChartLegendContent,
    ChartStyle,
};

```

FILE: frontend\src\components\ui\table.tsx

```

"use client";

import * as React from "react";

import { cn } from "@lib/utils";

function Table({ className, ...props }: React.ComponentProps<"table">) {
  return (
    <div
      data-slot="table-container"
      className="relative w-full overflow-x-auto"
    >
      <table
        data-slot="table"
        className={cn("w-full caption-bottom text-sm", className)}
        {...props}
      />
    </div>
  );
}

function TableHeader({ className, ...props }: React.ComponentProps<"thead">) {
  return (
    <thead
      data-slot="table-header"
      className={cn("&_tr:border-b", className)}
      {...props}
    />
  );
}

function TableBody({ className, ...props }: React.ComponentProps<"tbody">) {
  return (
    <tbody
      data-slot="table-body"
      className={cn("&_tr:last-child:border-0", className)}
      {...props}
    />
  );
}

function TableFooter({ className, ...props }: React.ComponentProps<"tfoot">) {
  return (
    <tfoot
      data-slot="table-footer"
      className={cn(
        "bg-muted/50 border-t font-medium [&>tr]:last:border-b-0",
        className
      )}
      {...props}
    />
  );
}

function TableRow({ className, ...props }: React.ComponentProps<"tr">) {
  return (
    <tr
      data-slot="table-row"
      className={cn(
        "hover:bg-muted/50 data-[state=selected]:bg-muted border-b transition-

```

```

colors",
    className
  )}
  {...props}
/>
);
}

function TableHead({ className, ...props }: React.ComponentProps<"th">) {
  return (
    <th
      data-slot="table-head"
      className={cn(
        "text-foreground h-10 px-2 text-left align-middle font-medium whitespace-
nowrap [&:has([role=checkbox]):pr-0 [&[role=checkbox]]:translate-y-[2px]",
        className
      )}
      {...props}
    />
  );
}

function TableCell({ className, ...props }: React.ComponentProps<"td">) {
  return (
    <td
      data-slot="table-cell"
      className={cn(
        "p-2 align-middle whitespace-nowrap [&:has([role=checkbox]):pr-0 [&
[role=checkbox]]:translate-y-[2px]",
        className
      )}
      {...props}
    />
  );
}

function TableCaption({
  className,
  ...props
}: React.ComponentProps<"caption">) {
  return (
    <caption
      data-slot="table-caption"
      className={cn("text-muted-foreground mt-4 text-sm", className)}
      {...props}
    />
  );
}

export {
  Table,
  TableHeader,
  TableBody,
  TableFooter,
  TableHead,
  TableRow,
  TableCell,
  TableCaption,
};

```

FILE: frontend\src\hooks\useCompanySearch.ts

```

"use client";
import { useEffect, useState } from "react";
import axios from "axios";
import { useStore } from "@/store";
import { profiles as mockProfiles } from "@/mockData";

export const useCompanySearch = () => {
  const [inputValue, setInputValue] = useState("");
  const [results, setResults] = useState<IProfile[]>([]);
  const [loading, setLoading] = useState(false);

  const profiles = useStore((state) => state.profiles);
  const setProfiles = useStore((state) => state.setProfiles);

  // Fetch profiles once
  useEffect(() => {
    const fetchProfiles = async () => {
      try {
        if (process.env.NEXT_PUBLIC_ENVIRONMENT === "production") {
          setProfiles(mockProfiles);
        } else {
          const res = await axios.get("http://localhost:3000/profiles");
          setProfiles(res.data);
        }
      } catch (err) {
        console.error("Error loading profiles", err);
      }
    };

    if (!profiles) fetchProfiles();
  }, [profiles, setProfiles]);

  // Debounced input search
  useEffect(() => {
    const delay = setTimeout(async () => {
      if (!inputValue.trim()) {
        setResults([]);
        return;
      }

      setLoading(true);
      try {
        if (process.env.NEXT_PUBLIC_ENVIRONMENT === "production") {
          const matches = profiles?.filter((prof) =>
            prof.ticker.toUpperCase().includes(inputValue.toUpperCase())
          );
          setResults(matches ?? []);
        } else {
          const res = await axios.get(
            `http://localhost:3000/profiles/${inputValue}`
          );
          setResults(res.data);
        }
      } catch (err) {
        console.error("Error fetching ticker matches", err);
        setResults([]);
      } finally {
        setLoading(false);
      }
    }, 400); // 400ms debounce
  }, [inputValue]);

```

```
    return () => clearTimeout(delay);  
  }, [inputValue, profiles]);  
  
  return {  
    inputValue,  
    setInputValue,  
    results,  
    loading,  
  };  
};
```

FILE: frontend\src\lib\utils.ts

```
import { clsx, type ClassValue } from "clsx"  
import { twMerge } from "tailwind-merge"  
  
export function cn(...inputs: ClassValue[]) {  
  return twMerge(clsx(inputs))  
}
```

FILE: frontend\src\mockData.js

```

const notes = [
  {
    _id: "1",
    title: "Apple",
    ticker: "AAPL",
    asset: "stock",
    content: "This is a note about Apple.",
    dateModified: new Date("2023-10-01T10:00:00Z"),
  },
  {
    _id: "2",
    title: "Microsoft",
    ticker: "MSFT",
    asset: "bond",
    content: "This is a note about Microsoft.",
    dateModified: new Date("2023-10-02T11:00:00Z"),
  },
  {
    _id: "3",
    title: "Google",
    ticker: "GOOGL",
    asset: "stock",
    content: "This is a note about Google.",
    dateModified: new Date("2023-10-03T12:00:00Z"),
  },
];

const profiles = [
  {
    _id: {
      $oid: "639824f7ba6700005400224c",
    },
    ticker: "AAPL",
    name_damodaran: "Apple Inc. (NasdaqGS:AAPL)",
    industry_damodaran: "computers_peripherals",
    sector_damodaran: "information_technology",
    country_damodaran: "United States",
    name_edgar: "APPLE INC",
    industry_edgar: "manufacturing",
    industry_detail_edgar: "electronic_computers",
    sector_yhoo: "technology",
    industry_yhoo: "consumer_electronics",
    long_business_summary:
      "Apple Inc. designs, manufactures, and markets smartphones, personal computers, tablets, wearables, and accessories worldwide. It also sells various related services. In addition, the company offers iPhone, a line of smartphones; Mac, a line of personal computers; iPad, a line of multi-purpose tablets; AirPods Max, an over-ear wireless headphone; and wearables, home, and accessories comprising AirPods, Apple TV, Apple Watch, Beats products, HomePod, and iPod touch. Further, it provides AppleCare support services; cloud services store services; and operates various platforms, including the App Store that allow customers to discover and download applications and digital content, such as books, music, video, games, and podcasts. Additionally, the company offers various services, such as Apple Arcade, a game subscription service; Apple Music, which offers users a curated listening experience with on-demand radio stations; Apple News+, a subscription news and magazine service; Apple TV+, which offers exclusive original content; Apple Card, a co-branded credit card; and Apple Pay, a cashless payment service, as well as licenses its intellectual property. The company serves consumers, and small and mid-sized businesses; and the education, enterprise, and government markets. It distributes third-party applications for its products through the App Store. The company also sells its

```

products through its retail and online stores, and direct sales force; and third-party cellular network carriers, wholesalers, retailers, and resellers. Apple Inc. was incorporated in 1977 and is headquartered in Cupertino, California."

```

    name_simfin: "APPLE INC",
    sector_simfin: "technology",
    industry_simfin: "computer_hardware",
    name_yhoo: "",
    name_seeking_alpha: "",
    sector_seeking_alpha: "unknown",
    industry_seeking_alpha: "unknown",
    dateModified: new Date("2023-04-01T10:00:00Z"),
  },
  {
    _id: {
      $oid: "639824f8ba67000054002b67",
    },
    ticker: "MSFT",
    name_damodaran: "Microsoft Corporation (NasdaqGS:MSFT)",
    industry_damodaran: "software_system_application",
    sector_damodaran: "information_technology",
    country_damodaran: "United States",
    name_edgar: "MICROSOFT CORP",
    industry_edgar: "services",
    industry_detail_edgar: "services_prepackaged_software",
    sector_yhoo: "technology",
    industry_yhoo: "software_application",
    long_business_summary:

```

"Microsoft Corporation develops, licenses, and supports software, services, devices, and solutions worldwide. Its Productivity and Business Processes segment offers Office, Exchange, SharePoint, Microsoft Teams, Office 365 Security and Compliance, and Skype for Business, as well as related Client Access Licenses (CAL); Skype, Outlook.com, OneDrive, and LinkedIn; and Dynamics 365, a set of cloud-based and on-premises business solutions for organizations and enterprise divisions. Its Intelligent Cloud segment licenses SQL, Windows Servers, Visual Studio, System Center, and related CALs; GitHub that provides a collaboration platform and code hosting service for developers; and Azure, a cloud platform. It also offers support services and Microsoft consulting services to assist customers in developing, deploying, and managing Microsoft server and desktop solutions; and training and certification on Microsoft products. Its More Personal Computing segment provides Windows original equipment manufacturer (OEM) licensing and other non-volume licensing of the Windows operating system; Windows Commercial, such as volume licensing of the Windows operating system, Windows cloud services, and other Windows commercial offerings; patent licensing; Windows Internet of Things; and MSN advertising. It also offers Surface, PC accessories, PCs, tablets, gaming and entertainment consoles, and other devices; Gaming, including Xbox hardware, and Xbox content and services; video games and third-party video game royalties; and Search, including Bing and Microsoft advertising. It sells its products through OEMs, distributors, and resellers; and directly through digital marketplaces, online stores, and retail stores. It has collaborations with Dynatrace, Inc., Morgan Stanley, Micro Focus, WPP plc, ACI Worldwide, Inc., and iCIMS, Inc., as well as strategic relationships with Avaya Holdings Corp. and wejo Limited. Microsoft Corporation was founded in 1975 and is based in Redmond, Washington."

```

    name_simfin: "MICROSOFT CORP",
    sector_simfin: "technology",
    industry_simfin: "application_software",
    name_yhoo: "",
    name_seeking_alpha: "",
    sector_seeking_alpha: "unknown",
    industry_seeking_alpha: "unknown",
    dateModified: new Date("2023-03-01T10:00:00Z"),
  },
  {
    _id: {
      $oid: "639824f8ba67000054002848",
    },

```

```

    },
    ticker: "GOOGL",
    name_damodaran: "",
    industry_damodaran: "unknown",
    sector_damodaran: "unknown",
    country_damodaran: "",
    name_edgar: "",
    industry_edgar: "unknown",
    industry_detail_edgar: "unknown",
    sector_yhoo: "communication_services",
    industry_yhoo: "internet_content_information",
    long_business_summary:
      "Alphabet Inc. provides online advertising services in the United States,
      Europe, the Middle East, Africa, the Asia-Pacific, Canada, and Latin America. The
      company offers performance and brand advertising services. It operates through Google
      Services, Google Cloud, and Other Bets segments. The Google Services segment provides
      products and services, such as ads, Android, Chrome, hardware, Google Maps, Google
      Play, Search, and YouTube, as well as technical infrastructure; and digital content.
      The Google Cloud segment offers infrastructure and data analytics platforms,
      collaboration tools, and other services for enterprise customers. The Other Bets
      segment sells internet and TV services, as well as licensing and research and
      development services. The company was founded in 1998 and is headquartered in
      Mountain View, California.",
    name_simfin: "",
    sector_simfin: "unknown",
    industry_simfin: "unknown",
    name_yhoo: "",
    name_seeking_alpha: "",
    sector_seeking_alpha: "unknown",
    industry_seeking_alpha: "unknown",
    dateModified: new Date("2023-10-01T10:00:00Z"),
  },
  {
    _id: {
      $oid: "639824f9ba67000054002d6a",
    },
    ticker: "ZM",
    name_damodaran: "Zoom Video Communications, Inc. (NasdaqGS:ZM)",
    industry_damodaran: "software_system_application",
    sector_damodaran: "information_technology",
    country_damodaran: "United States",
    name_edgar: "ZOOM VIDEO COMMUNICATIONS INC",
    industry_edgar: "services",
    industry_detail_edgar: "services_computer_programming_data_processing_etc",
    sector_yhoo: "technology",
    industry_yhoo: "application_software",
    long_business_summary:
      "Zoom Video Communications, Inc. provides a video-first communications platform
      in the Americas, the Asia Pacific, Europe, the Middle East, and Africa. The company
      offers Zoom Meetings that offers HD video, voice, chat, and content sharing through
      mobile devices, desktops, laptops, telephones, and conference room systems; Zoom
      Phone, an enterprise cloud phone system that provides secure call routing, call
      queuing, call detail reports, call recording, call quality monitoring, voicemail,
      switch to video, and other services, as well as inbound and outbound calling
      services; and Zoom Chat enables to share messages, images, audio files, and content
      in desktop, laptop, tablet, and mobile devices for meeting and phone customers. It
      also provides Zoom Rooms, a software-based conference room system; Zoom Conference
      Room Connector, a gateway for SIP/H.323 endpoints to join Zoom meetings; Zoom Video
      Webinars to provide video presentations to large audiences from many devices; and
      Zoom Hardware-as-a-Service allows users to access video communication technology with
      subscription options for phone and meeting room hardware. The company offers Zoom for
      Developers to integrate its video, phone, chat, and content sharing into other
      applications, as well as manages Zoom accounts; Zoom App Marketplace enhance
      developers to publish their apps; OnZoom, a platform for users to create, host, and

```


monetize online events; and Zoom Apps to access from Zoom Meetings and the Zoom Desktop client to facilitate collaboration and engagement during meetings. It serves individuals; and education, entertainment/media, enterprise infrastructure, finance, government, healthcare, manufacturing, non-profit/not for profit and social impact, retail/consumer products, and software/Internet industries. The company was formerly known as Zoom Communications, Inc. and changed its name to Zoom Video Communications, Inc. in May 2012. The company was incorporated in 2011 and is headquartered in San Jose, California.",

```
name_simfin: "Zoom Video Communications, Inc.",
sector_simfin: "technology",
industry_simfin: "application_software",
name_yhoo: "",
name_seeking_alpha: "",
sector_seeking_alpha: "unknown",
industry_seeking_alpha: "unknown",
dateModified: new Date("2023-10-02T10:00:00Z"),
```

```
},
```

```
{
```

```
_id: {
  $oid: "639824f7ba67000054002336",
},
ticker: "AMZN",
name_damodaran: "Amazon.com, Inc. (NasdaqGS:AMZN)",
industry_damodaran: "retail_online",
sector_damodaran: "consumer_discretionary",
country_damodaran: "United States",
name_edgar: "AMAZON COM INC",
industry_edgar: "unknown",
industry_detail_edgar: "retail_catalog_mail_order_houses",
sector_yhoo: "consumer_cyclical",
industry_yhoo: "internet_retail",
long_business_summary:
```

"Amazon.com, Inc. engages in the retail sale of consumer products and subscriptions in North America and internationally. The company operates through three segments: North America, International, and Amazon Web Services (AWS). It sells merchandise and content purchased for resale from third-party sellers through physical and online stores. The company also manufactures and sells electronic devices, including Kindle, Fire tablets, Fire TVs, Rings, and Echo and other devices; provides Kindle Direct Publishing, an online service that allows independent authors and publishers to make their books available in the Kindle Store; and develops and produces media content. In addition, it offers programs that enable sellers to sell their products on its websites, as well as its stores; and programs that allow authors, musicians, filmmakers, skill and app developers, and others to publish and sell content. Further, the company provides compute, storage, database, analytics, machine learning, and other services, as well as fulfillment, advertising, publishing, and digital content subscriptions. Additionally, it offers Amazon Prime, a membership program, which provides free shipping of various items; access to streaming of movies and TV episodes; and other services. The company serves consumers, sellers, developers, enterprises, and content creators. Amazon.com, Inc. was founded in 1994 and is headquartered in Seattle, Washington.",

```
name_simfin: "AMAZON COM INC",
sector_simfin: "consumer_cyclical",
industry_simfin: "retail_apparel_specialty",
name_yhoo: "",
name_seeking_alpha: "",
sector_seeking_alpha: "unknown",
industry_seeking_alpha: "unknown",
dateModified: new Date("2023-10-07T10:00:00Z"),
```

```
},
```

```
{
```

```
_id: {
  $oid: "639824f7ba6700005400260a",
},
ticker: "CTAQ",
```

```

name_damodaran: "Carney Technology Acquisition Corp. II (NasdaqCM:CTAQ)",
industry_damodaran: "investments_asset_management",
sector_damodaran: "financials",
country_damodaran: "United States",
name_edgar: "CARNEY TECHNOLOGY ACQUISITION CORP II",
industry_edgar: "unknown",
industry_detail_edgar: "blank_checks",
sector_yhoo: "financial_services",
industry_yhoo: "shell_companies",
long_business_summary:
  "Carney Technology Acquisition Corp. II does not have significant operations.
  It intends to effect a merger, capital stock exchange, asset acquisition, stock
  purchase, reorganization, or similar business combination with one or more businesses
  in the technology industry. The company was incorporated in 2020 and is based in
  Burlingame, California.",
name_simfin: "",
sector_simfin: "unknown",
industry_simfin: "unknown",
name_yhoo: "",
name_seeking_alpha: "",
sector_seeking_alpha: "unknown",
industry_seeking_alpha: "unknown",
dateModified: new Date("2023-10-11T10:00:00Z"),
},
];

const treasuries = [
  {
    _id: "1",
    maturity: "2024-01-01",
    coupon: 0.5,
    yield: 0.25,
    price: 99.5,
  },
  {
    _id: "2",
    maturity: "2025-01-01",
    coupon: 1.0,
    yield: 0.75,
    price: 98.5,
  },
  {
    _id: "3",
    maturity: "2026-01-01",
    coupon: 1.5,
    yield: 1.25,
    price: 97.5,
  },
];

const SP500 = [
  {
    _id: "1",
    date: "1980-01-31",
    value: 23,
  },
  {
    _id: "2",
    date: "1980-02-28",
    value: 45,
  },
  {
    _id: "3",

```

```

    date: "1980-03-31",
    value: 67,
  },
  {
    _id: "4",
    date: "1980-04-30",
    value: 89,
  },
];

```

```

const unemploymentRate = [
  {
    _id: "1",
    date: "1980-01-31",
    value: "3.8",
  },
  {
    _id: "2",
    date: "1980-02-28",
    value: "3.9",
  },
  {
    _id: "3",
    date: "1980-03-31",
    value: "5.4",
  },
  {
    _id: "4",
    date: "1980-04-30",
    value: "5.3",
  },
];

```

```

const EXJPUS = [
  {
    _id: {
      $oid: "66f500be272a5d9b72c0996a",
    },
    date: "2000-01-01",
    value: "105.2960",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c0996b",
    },
    date: "2000-02-01",
    value: "109.3885",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c0996c",
    },
    date: "2000-03-01",
    value: "106.3074",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c0996d",
    },
    date: "2000-04-01",
    value: "105.6270",
  },
];

```

```

    _id: {
      $oid: "66f500be272a5d9b72c09996e",
    },
    date: "2000-05-01",
    value: "108.3205",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09996f",
    },
    date: "2000-06-01",
    value: "106.1255",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099970",
    },
    date: "2000-07-01",
    value: "108.2115",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099971",
    },
    date: "2000-08-01",
    value: "108.0804",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099972",
    },
    date: "2000-09-01",
    value: "106.8375",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099973",
    },
    date: "2000-10-01",
    value: "108.4429",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099974",
    },
    date: "2000-11-01",
    value: "109.0095",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099975",
    },
    date: "2000-12-01",
    value: "112.2090",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099976",
    },
    date: "2001-01-01",
    value: "116.6719",
  },
  {
    _id: {

```

```

    $oid: "66f500be272a5d9b72c09977",
  },
  date: "2001-02-01",
  value: "116.2337",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09978",
  },
  date: "2001-03-01",
  value: "121.5050",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09979",
  },
  date: "2001-04-01",
  value: "123.7710",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c0997a",
  },
  date: "2001-05-01",
  value: "121.7682",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c0997b",
  },
  date: "2001-06-01",
  value: "122.3510",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c0997c",
  },
  date: "2001-07-01",
  value: "124.4981",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c0997d",
  },
  date: "2001-08-01",
  value: "121.3670",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c0997e",
  },
  date: "2001-09-01",
  value: "118.6117",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c0997f",
  },
  date: "2001-10-01",
  value: "121.4536",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09980",

```

```

    },
    date: "2001-11-01",
    value: "122.4055",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09981",
    },
    date: "2001-12-01",
    value: "127.5945",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09982",
    },
    date: "2002-01-01",
    value: "132.6833",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09983",
    },
    date: "2002-02-01",
    value: "133.6426",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09984",
    },
    date: "2002-03-01",
    value: "131.0610",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09985",
    },
    date: "2002-04-01",
    value: "130.7718",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09986",
    },
    date: "2002-05-01",
    value: "126.3750",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09987",
    },
    date: "2002-06-01",
    value: "123.2905",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09988",
    },
    date: "2002-07-01",
    value: "117.8991",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09989",
    },

```

```

    date: "2002-08-01",
    value: "118.9927",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c0998a",
    },
    date: "2002-09-01",
    value: "121.0780",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c0998b",
    },
    date: "2002-10-01",
    value: "123.9077",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c0998c",
    },
    date: "2002-11-01",
    value: "121.6079",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c0998d",
    },
    date: "2002-12-01",
    value: "121.8929",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c0998e",
    },
    date: "2003-01-01",
    value: "118.8133",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c0998f",
    },
    date: "2003-02-01",
    value: "119.3379",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09990",
    },
    date: "2003-03-01",
    value: "118.6871",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09991",
    },
    date: "2003-04-01",
    value: "119.8950",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09992",
    },
    date: "2003-05-01",

```

```

    value: "117.3681",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09993",
    },
    date: "2003-06-01",
    value: "118.3290",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09994",
    },
    date: "2003-07-01",
    value: "118.6959",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09995",
    },
    date: "2003-08-01",
    value: "118.6624",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09996",
    },
    date: "2003-09-01",
    value: "114.80",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09997",
    },
    date: "2003-10-01",
    value: "109.4955",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09998",
    },
    date: "2003-11-01",
    value: "109.1778",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09999",
    },
    date: "2003-12-01",
    value: "107.7377",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09999a",
    },
    date: "2004-01-01",
    value: "106.2685",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09999b",
    },
    date: "2004-02-01",
    value: "106.7079",
  },

```



```

},
{
  _id: {
    $oid: "66f500be272a5d9b72c0999c",
  },
  date: "2004-03-01",
  value: "108.5157",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c0999d",
  },
  date: "2004-04-01",
  value: "107.6564",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c0999e",
  },
  date: "2004-05-01",
  value: "112.1960",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c0999f",
  },
  date: "2004-06-01",
  value: "109.4336",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c0999a0",
  },
  date: "2004-07-01",
  value: "109.4871",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c0999a1",
  },
  date: "2004-08-01",
  value: "110.2336",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c0999a2",
  },
  date: "2004-09-01",
  value: "110.0914",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c0999a3",
  },
  date: "2004-10-01",
  value: "108.7835",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c0999a4",
  },
  date: "2004-11-01",
  value: "104.6990",
},

```

```

{
  _id: {
    $oid: "66f500be272a5d9b72c099a5",
  },
  date: "2004-12-01",
  value: "103.8104",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099a6",
  },
  date: "2005-01-01",
  value: "103.3410",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099a7",
  },
  date: "2005-02-01",
  value: "104.9442",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099a8",
  },
  date: "2005-03-01",
  value: "105.2543",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099a9",
  },
  date: "2005-04-01",
  value: "107.1938",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099aa",
  },
  date: "2005-05-01",
  value: "106.5952",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099ab",
  },
  date: "2005-06-01",
  value: "108.7473",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099ac",
  },
  date: "2005-07-01",
  value: "111.9535",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099ad",
  },
  date: "2005-08-01",
  value: "110.6065",
},
{

```

```

    _id: {
      $oid: "66f500be272a5d9b72c099ae",
    },
    date: "2005-09-01",
    value: "111.2390",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099af",
    },
    date: "2005-10-01",
    value: "114.8695",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099b0",
    },
    date: "2005-11-01",
    value: "118.4540",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099b1",
    },
    date: "2005-12-01",
    value: "118.4624",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099b2",
    },
    date: "2006-01-01",
    value: "115.4765",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099b3",
    },
    date: "2006-02-01",
    value: "117.8605",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099b4",
    },
    date: "2006-03-01",
    value: "117.2778",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099b5",
    },
    date: "2006-04-01",
    value: "117.0695",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099b6",
    },
    date: "2006-05-01",
    value: "111.7305",
  },
  {
    _id: {

```

```

    $oid: "66f500be272a5d9b72c099b7",
  },
  date: "2006-06-01",
  value: "114.6250",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099b8",
  },
  date: "2006-07-01",
  value: "115.7670",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099b9",
  },
  date: "2006-08-01",
  value: "115.9243",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099ba",
  },
  date: "2006-09-01",
  value: "117.2145",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099bb",
  },
  date: "2006-10-01",
  value: "118.6090",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099bc",
  },
  date: "2006-11-01",
  value: "117.3205",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099bd",
  },
  date: "2006-12-01",
  value: "117.3220",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099be",
  },
  date: "2007-01-01",
  value: "120.4471",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099bf",
  },
  date: "2007-02-01",
  value: "120.5047",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099c0",

```

```

    },
    date: "2007-03-01",
    value: "117.26",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099c1",
    },
    date: "2007-04-01",
    value: "118.9324",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099c2",
    },
    date: "2007-05-01",
    value: "120.7732",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099c3",
    },
    date: "2007-06-01",
    value: "122.6886",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099c4",
    },
    date: "2007-07-01",
    value: "121.4148",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099c5",
    },
    date: "2007-08-01",
    value: "116.7335",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099c6",
    },
    date: "2007-09-01",
    value: "115.0435",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099c7",
    },
    date: "2007-10-01",
    value: "115.8661",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099c8",
    },
    date: "2007-11-01",
    value: "111.0729",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099c9",
    },

```

```

    date: "2007-12-01",
    value: "112.4490",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099ca",
    },
    date: "2008-01-01",
    value: "107.8181",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099cb",
    },
    date: "2008-02-01",
    value: "107.03",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099cc",
    },
    date: "2008-03-01",
    value: "100.7562",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099cd",
    },
    date: "2008-04-01",
    value: "102.6777",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099ce",
    },
    date: "2008-05-01",
    value: "104.3595",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099cf",
    },
    date: "2008-06-01",
    value: "106.9152",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099d0",
    },
    date: "2008-07-01",
    value: "106.8518",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099d1",
    },
    date: "2008-08-01",
    value: "109.3624",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099d2",
    },
    date: "2008-09-01",

```

```

    value: "106.5748",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099d3",
    },
    date: "2008-10-01",
    value: "99.9659",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099d4",
    },
    date: "2008-11-01",
    value: "96.9656",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099d5",
    },
    date: "2008-12-01",
    value: "91.2750",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099d6",
    },
    date: "2009-01-01",
    value: "90.1205",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099d7",
    },
    date: "2009-02-01",
    value: "92.9158",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099d8",
    },
    date: "2009-03-01",
    value: "97.8550",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099d9",
    },
    date: "2009-04-01",
    value: "98.9200",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099da",
    },
    date: "2009-05-01",
    value: "96.6445",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099db",
    },
    date: "2009-06-01",
    value: "96.6145",
  },

```

```

},
{
  _id: {
    $oid: "66f500be272a5d9b72c099dc",
  },
  date: "2009-07-01",
  value: "94.3670",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099dd",
  },
  date: "2009-08-01",
  value: "94.8971",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099de",
  },
  date: "2009-09-01",
  value: "91.2748",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099df",
  },
  date: "2009-10-01",
  value: "90.3671",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099e0",
  },
  date: "2009-11-01",
  value: "89.2674",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099e1",
  },
  date: "2009-12-01",
  value: "89.9509",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099e2",
  },
  date: "2010-01-01",
  value: "91.1011",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099e3",
  },
  date: "2010-02-01",
  value: "90.1395",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099e4",
  },
  date: "2010-03-01",
  value: "90.7161",
},

```



```

{
  _id: {
    $oid: "66f500be272a5d9b72c099e5",
  },
  date: "2010-04-01",
  value: "93.4527",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099e6",
  },
  date: "2010-05-01",
  value: "91.9730",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099e7",
  },
  date: "2010-06-01",
  value: "90.8059",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099e8",
  },
  date: "2010-07-01",
  value: "87.5005",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099e9",
  },
  date: "2010-08-01",
  value: "85.3727",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099ea",
  },
  date: "2010-09-01",
  value: "84.3571",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099eb",
  },
  date: "2010-10-01",
  value: "81.7285",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099ec",
  },
  date: "2010-11-01",
  value: "82.5180",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099ed",
  },
  date: "2010-12-01",
  value: "83.3376",
},
{

```

```

    _id: {
      $oid: "66f500be272a5d9b72c099ee",
    },
    date: "2011-01-01",
    value: "82.6250",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099ef",
    },
    date: "2011-02-01",
    value: "82.5368",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099f0",
    },
    date: "2011-03-01",
    value: "81.6470",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099f1",
    },
    date: "2011-04-01",
    value: "83.1771",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099f2",
    },
    date: "2011-05-01",
    value: "81.1257",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099f3",
    },
    date: "2011-06-01",
    value: "80.4259",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099f4",
    },
    date: "2011-07-01",
    value: "79.2425",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099f5",
    },
    date: "2011-08-01",
    value: "76.9657",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c099f6",
    },
    date: "2011-09-01",
    value: "76.7957",
  },
  {
    _id: {

```

```

    $oid: "66f500be272a5d9b72c099f7",
  },
  date: "2011-10-01",
  value: "76.6430",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099f8",
  },
  date: "2011-11-01",
  value: "77.5595",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099f9",
  },
  date: "2011-12-01",
  value: "77.7967",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099fa",
  },
  date: "2012-01-01",
  value: "76.9640",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099fb",
  },
  date: "2012-02-01",
  value: "78.4700",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099fc",
  },
  date: "2012-03-01",
  value: "82.4659",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099fd",
  },
  date: "2012-04-01",
  value: "81.2524",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099fe",
  },
  date: "2012-05-01",
  value: "79.6668",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c099ff",
  },
  date: "2012-06-01",
  value: "79.3152",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a00",

```

```

    },
    date: "2012-07-01",
    value: "78.9348",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a01",
    },
    date: "2012-08-01",
    value: "78.6909",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a02",
    },
    date: "2012-09-01",
    value: "78.1353",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a03",
    },
    date: "2012-10-01",
    value: "79.0132",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a04",
    },
    date: "2012-11-01",
    value: "81.0305",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a05",
    },
    date: "2012-12-01",
    value: "83.7905",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a06",
    },
    date: "2013-01-01",
    value: "89.0581",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a07",
    },
    date: "2013-02-01",
    value: "93.0016",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a08",
    },
    date: "2013-03-01",
    value: "94.7700",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a09",
    },

```

```

    date: "2013-04-01",
    value: "97.7582",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a0a",
    },
    date: "2013-05-01",
    value: "100.9186",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a0b",
    },
    date: "2013-06-01",
    value: "97.2350",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a0c",
    },
    date: "2013-07-01",
    value: "99.6727",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a0d",
    },
    date: "2013-08-01",
    value: "97.8123",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a0e",
    },
    date: "2013-09-01",
    value: "99.2100",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a0f",
    },
    date: "2013-10-01",
    value: "97.7700",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a10",
    },
    date: "2013-11-01",
    value: "100.0737",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a11",
    },
    date: "2013-12-01",
    value: "103.4600",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a12",
    },
    date: "2014-01-01",

```

```

    value: "103.7614",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a13",
    },
    date: "2014-02-01",
    value: "102.1253",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a14",
    },
    date: "2014-03-01",
    value: "102.3395",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a15",
    },
    date: "2014-04-01",
    value: "102.4582",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a16",
    },
    date: "2014-05-01",
    value: "101.7738",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a17",
    },
    date: "2014-06-01",
    value: "102.0629",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a18",
    },
    date: "2014-07-01",
    value: "101.7400",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a19",
    },
    date: "2014-08-01",
    value: "102.9438",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a1a",
    },
    date: "2014-09-01",
    value: "107.4257",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a1b",
    },
    date: "2014-10-01",
    value: "108.0264",
  },

```

```

},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a1c",
  },
  date: "2014-11-01",
  value: "116.2994",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a1d",
  },
  date: "2014-12-01",
  value: "119.3233",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a1e",
  },
  date: "2015-01-01",
  value: "118.2500",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a1f",
  },
  date: "2015-02-01",
  value: "118.7600",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a20",
  },
  date: "2015-03-01",
  value: "120.3945",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a21",
  },
  date: "2015-04-01",
  value: "119.5095",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a22",
  },
  date: "2015-05-01",
  value: "120.7980",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a23",
  },
  date: "2015-06-01",
  value: "123.7186",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a24",
  },
  date: "2015-07-01",
  value: "123.3109",
},

```

```

{
  _id: {
    $oid: "66f500be272a5d9b72c09a25",
  },
  date: "2015-08-01",
  value: "123.0038",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a26",
  },
  date: "2015-09-01",
  value: "120.1476",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a27",
  },
  date: "2015-10-01",
  value: "120.0481",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a28",
  },
  date: "2015-11-01",
  value: "122.6432",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a29",
  },
  date: "2015-12-01",
  value: "121.6350",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a2a",
  },
  date: "2016-01-01",
  value: "118.2258",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a2b",
  },
  date: "2016-02-01",
  value: "114.6155",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a2c",
  },
  date: "2016-03-01",
  value: "112.9317",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a2d",
  },
  date: "2016-04-01",
  value: "109.5519",
},
{

```



```

    _id: {
      $oid: "66f500be272a5d9b72c09a2e",
    },
    date: "2016-05-01",
    value: "108.8481",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a2f",
    },
    date: "2016-06-01",
    value: "105.3509",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a30",
    },
    date: "2016-07-01",
    value: "104.1910",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a31",
    },
    date: "2016-08-01",
    value: "101.2383",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a32",
    },
    date: "2016-09-01",
    value: "101.7843",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a33",
    },
    date: "2016-10-01",
    value: "103.9075",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a34",
    },
    date: "2016-11-01",
    value: "108.4430",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a35",
    },
    date: "2016-12-01",
    value: "115.9981",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a36",
    },
    date: "2017-01-01",
    value: "114.8721",
  },
  {
    _id: {

```

```

    $oid: "66f500be272a5d9b72c09a37",
  },
  date: "2017-02-01",
  value: "112.9116",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a38",
  },
  date: "2017-03-01",
  value: "112.9165",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a39",
  },
  date: "2017-04-01",
  value: "110.0910",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a3a",
  },
  date: "2017-05-01",
  value: "112.2436",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a3b",
  },
  date: "2017-06-01",
  value: "110.9141",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a3c",
  },
  date: "2017-07-01",
  value: "112.417",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a3d",
  },
  date: "2017-08-01",
  value: "109.827",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a3e",
  },
  date: "2017-09-01",
  value: "110.776",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a3f",
  },
  date: "2017-10-01",
  value: "112.9148",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a40",

```

```

    },
    date: "2017-11-01",
    value: "112.819",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a41",
    },
    date: "2017-12-01",
    value: "112.9405",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a42",
    },
    date: "2018-01-01",
    value: "110.871",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a43",
    },
    date: "2018-02-01",
    value: "107.9700",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a44",
    },
    date: "2018-03-01",
    value: "106.0468",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a45",
    },
    date: "2018-04-01",
    value: "107.6562",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a46",
    },
    date: "2018-05-01",
    value: "109.6882",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a47",
    },
    date: "2018-06-01",
    value: "110.0638",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a48",
    },
    date: "2018-07-01",
    value: "111.5210",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a49",
    },

```

```

    date: "2018-08-01",
    value: "110.9965",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a4a",
    },
    date: "2018-09-01",
    value: "112.0974",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a4b",
    },
    date: "2018-10-01",
    value: "112.7218",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a4c",
    },
    date: "2018-11-01",
    value: "113.3380",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a4d",
    },
    date: "2018-12-01",
    value: "112.1994",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a4e",
    },
    date: "2019-01-01",
    value: "108.9605",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a4f",
    },
    date: "2019-02-01",
    value: "110.4400",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a50",
    },
    date: "2019-03-01",
    value: "111.1443",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a51",
    },
    date: "2019-04-01",
    value: "111.6414",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a52",
    },
    date: "2019-05-01",

```

```

    value: "109.9714",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a53",
    },
    date: "2019-06-01",
    value: "108.0685",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a54",
    },
    date: "2019-07-01",
    value: "108.2864",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a55",
    },
    date: "2019-08-01",
    value: "106.1886",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a56",
    },
    date: "2019-09-01",
    value: "107.5400",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a57",
    },
    date: "2019-10-01",
    value: "108.1368",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a58",
    },
    date: "2019-11-01",
    value: "108.8579",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a59",
    },
    date: "2019-12-01",
    value: "109.1010",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a5a",
    },
    date: "2020-01-01",
    value: "109.2667",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a5b",
    },
    date: "2020-02-01",
    value: "110.0295",
  },

```

```

},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a5c",
  },
  date: "2020-03-01",
  value: "107.6673",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a5d",
  },
  date: "2020-04-01",
  value: "107.7386",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a5e",
  },
  date: "2020-05-01",
  value: "107.2000",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a5f",
  },
  date: "2020-06-01",
  value: "107.5782",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a60",
  },
  date: "2020-07-01",
  value: "106.6818",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a61",
  },
  date: "2020-08-01",
  value: "106.0129",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a62",
  },
  date: "2020-09-01",
  value: "105.5890",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a63",
  },
  date: "2020-10-01",
  value: "105.2095",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a64",
  },
  date: "2020-11-01",
  value: "104.4061",
},

```

```

{
  _id: {
    $oid: "66f500be272a5d9b72c09a65",
  },
  date: "2020-12-01",
  value: "103.7952",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a66",
  },
  date: "2021-01-01",
  value: "103.7883",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a67",
  },
  date: "2021-02-01",
  value: "105.3774",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a68",
  },
  date: "2021-03-01",
  value: "108.6991",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a69",
  },
  date: "2021-04-01",
  value: "109.0445",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a6a",
  },
  date: "2021-05-01",
  value: "109.1120",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a6b",
  },
  date: "2021-06-01",
  value: "110.1073",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a6c",
  },
  date: "2021-07-01",
  value: "110.2143",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a6d",
  },
  date: "2021-08-01",
  value: "109.8527",
},
{

```

```

    _id: {
      $oid: "66f500be272a5d9b72c09a6e",
    },
    date: "2021-09-01",
    value: "110.1610",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a6f",
    },
    date: "2021-10-01",
    value: "113.1215",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a70",
    },
    date: "2021-11-01",
    value: "113.9650",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a71",
    },
    date: "2021-12-01",
    value: "113.8329",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a72",
    },
    date: "2022-01-01",
    value: "114.8255",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a73",
    },
    date: "2022-02-01",
    value: "115.2763",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a74",
    },
    date: "2022-03-01",
    value: "118.5774",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a75",
    },
    date: "2022-04-01",
    value: "126.3743",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a76",
    },
    date: "2022-05-01",
    value: "128.8486",
  },
  {
    _id: {

```



```

    $oid: "66f500be272a5d9b72c09a77",
  },
  date: "2022-06-01",
  value: "133.9619",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a78",
  },
  date: "2022-07-01",
  value: "136.7090",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a79",
  },
  date: "2022-08-01",
  value: "135.2835",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a7a",
  },
  date: "2022-09-01",
  value: "143.2843",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a7b",
  },
  date: "2022-10-01",
  value: "147.0515",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a7c",
  },
  date: "2022-11-01",
  value: "142.4450",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a7d",
  },
  date: "2022-12-01",
  value: "134.9148",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a7e",
  },
  date: "2023-01-01",
  value: "130.4475",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a7f",
  },
  date: "2023-02-01",
  value: "133.0463",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a80",

```

```

    },
    date: "2023-03-01",
    value: "133.6643",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a81",
    },
    date: "2023-04-01",
    value: "133.4745",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a82",
    },
    date: "2023-05-01",
    value: "137.0532",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a83",
    },
    date: "2023-06-01",
    value: "141.3581",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a84",
    },
    date: "2023-07-01",
    value: "140.9360",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a85",
    },
    date: "2023-08-01",
    value: "144.7804",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a86",
    },
    date: "2023-09-01",
    value: "147.8450",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a87",
    },
    date: "2023-10-01",
    value: "149.5933",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a88",
    },
    date: "2023-11-01",
    value: "149.6790",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a89",
    },

```

```

    date: "2023-12-01",
    value: "143.9815",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a8a",
    },
    date: "2024-01-01",
    value: "146.2943",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a8b",
    },
    date: "2024-02-01",
    value: "149.6150",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a8c",
    },
    date: "2024-03-01",
    value: "149.8186",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a8d",
    },
    date: "2024-04-01",
    value: "153.8900",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a8e",
    },
    date: "2024-05-01",
    value: "155.8691",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a8f",
    },
    date: "2024-06-01",
    value: "157.86",
  },
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a90",
    },
    date: "2024-07-01",
    value: "157.5182",
  },
];

const initialClaims = [
  {
    _id: {
      $oid: "66f500be272a5d9b72c09a91",
    },
    date: "2000-01-01",
    value: "243",
  },
  {
    _id: {

```

```

    $oid: "66f500be272a5d9b72c09a92",
  },
  date: "2000-02-01",
  value: "234",
},
{
  _id: {
    $oid: "66f500be272a5d9b72c09a93",
  },
  date: "2000-03-01",
  value: "250",
},
];

module.exports = {
  notes,
  profiles,
  treasuries,
  SP500,
  unemploymentRate,
  EXJPUS,
  initialClaims,
};

```

FILE: frontend\src\store.ts

```

import { create } from "zustand";

export interface IState {
  ticker: string;
  setTicker: (ticker: string) => void;
  profiles: IProfile[] | null;
  setProfiles: (profiles: IProfile[]) => void;
  // increasePopulation: () => void;
  // removeAllBears: () => void;
  profile: IProfile | null;
  setProfile: (profile: IProfile) => void;
  notes: INote[] | null;
  setNotes: (notes: INote[]) => void;
}

export const useStore = create<IState>((set) => ({
  ticker: "",
  setTicker: (ticker: string) => set({ ticker }),
  profiles: null,
  setProfiles: (profiles: IProfile[]) => set({ profiles }),
  profile: null,
  setProfile: (profile: IProfile) => set({ profile }),
  notes: null,
  setNotes: (notes: INote[]) => set({ notes }),
}));

// console.log("Profiles: ", useProfileStore.getState().profiles);

```

FILE: frontend\types.d.ts

```
export {};  
  
// make globally accessible (no need to import)  
declare global {  
  interface INote {  
    _id: string;  
    title: string;  
    ticker: string;  
    asset: string;  
    content: string;  
    dateModified: Date;  
  }  
  
  interface IProfile {  
    _id: { $oid: string };  
    ticker: string;  
    name_damodaran: string;  
    name_edgar: string;  
    name_yhoo: string;  
    name_simfin: string;  
    name_seeking_alpha: string;  
    industry_damodaran: string;  
    industry_edgar: string;  
    industry_detail_edgar: string;  
    industry_yhoo: string;  
    industry_simfin: string;  
    industry_seeking_alpha: string;  
    sector_damodaran: string;  
    sector_simfin: string;  
    sector_yhoo: string;  
    sector_seeking_alpha: string;  
    country_damodaran: string;  
    dateModified: Date;  
  }  
}
```