```
// /app/types/chat.ts

export interface FinancialMetadata {
    instruments?: string[];
    sentiment?: ’bullish’ | ’bearish’ | ’neutral’;
    riskLevel?: ’low’ | ’medium’ | ’high’;
    confidenceScore?: number;
    technicalIndicators?: string[];
    fundamentalFactors?: string[];
  }

  export interface Message {
    id: string;
    role: string;
    content: string;
    metadata?: FinancialMetadata;
  }

  export interface ChatResponse {
    id: string;
    response: string;
    thinking: string;
    user_mood: string;
    suggested_questions: string[];
    financial_context: {
      instruments_mentioned: string[];
      market_sentiment: ’bullish’ | ’bearish’ | ’neutral’;
      risk_level: ’low’ | ’medium’ | ’high’;
      confidence_score: number;
      technical_indicators: string[];
      fundamental_factors: string[];
    };
    debug: {
      context_used: boolean;
      market_data_used: boolean;
    };
  }
```

```
// config/navigation.ts

import { LayoutDashboard, BarChart2, FileText } from "lucide-react";

export const mainNav = [
  {
    title: "Chat",
    href: "/",
    icon: LayoutDashboard,
  },
  {
    title: "Market",
    href: "/market",
    icon: BarChart2,
  },
  {
    title: "Documents",
    href: "/documents",
    icon: FileText,
  },
];
```

```
// /app/config/market.ts

export const API_KEY = process.env.NEXT_PUBLIC_TWELVE_DATA_API_KEY || 'demo';
export const API_URL = 'https://api.twelvedata.com';

export interface CompanyLogo {
  meta: {
    symbol: string;
    name: string;
    currency: string;
    exchange: string;
    mic_code: string;
    exchange_timezone: string;
  };
  url: string;
}

export const SAUDI_MARKET = {
  exchange: 'Tadawul',
  mic_code: 'XSAU',
  currency: 'SAR',
  timezone: 'Asia/Riyadh',
  trading_hours: 'Su-Th, 10:00am - 3:00pm',
  data_delay: '15min'
} as const;

// Updated with correct symbols and standardized format
export const SAUDI_SYMBOLS = [
  // Banks & Financial Services

      {
          symbol: '2222:TADAWUL',
          tickerSymbol: '2222',
          name: 'Saudi Aramco',
          name_ar: 'Ø£Ø±Ø§Ù\205Ù\203Ù\210 Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212Ø©',
          sector: 'Energy',
          website: 'www.saudiaramco.com'
      },
      {
          symbol: '2381:TADAWUL',
          tickerSymbol: '2381',
          name: 'Arabian Drilling',
          name_ar: 'Ø´Ø±Ù\203Ø© Ø§Ù\204Ø-Ù\201Ø± Ø§Ù\204Ø¹Ø±Ø¨Ù\212Ø©',
          sector: 'Energy',
          website: 'www.arabdrill.com'
      },
      {
          symbol: '2382:TADAWUL',
          tickerSymbol: '2382',
          name: 'ADES',
          name_ar: 'Ø£Ø¯Ù\212Ø³',
          sector: 'Energy',
          website: 'www.adesgroup.com'
      },
      {
          symbol: '2380:TADAWUL',
          tickerSymbol: '2380',
          name: 'Petro Rabigh',
          name_ar: 'Ø¨ØªØ±Ù\210Ø±Ø§Ø¨Øº',
          sector: 'Energy',
          website: 'www.petrorabigh.com'
      },
      {
          symbol: '4030:TADAWUL',
          tickerSymbol: '4030',
          name: 'Bahri',
          name_ar: 'Ø§Ù\204Ø¨Ø-Ø±Ù\212',
          sector: 'Transportation',
          website: 'www.bahri.sa'
      },
      {
          symbol: '4200:TADAWUL',
          tickerSymbol: '4200',
          name: 'Aldrees',
          name_ar: 'Ø§Ù\204Ø¯Ø±Ù\212Ø³',
          sector: 'Energy',
          website: 'www.aldrees.com'
      },
      {
          symbol: '2030:TADAWUL',
          tickerSymbol: '2030',
          name: 'SARCO',
          name_ar: 'Ø³Ø§Ø±Ù\203Ù\210',
          sector: 'Materials',
          website: 'www.sarco.com.sa'
      },
      {
          symbol: '1202:TADAWUL',
          tickerSymbol: '1202',
          name: 'MEPCO',
          name_ar: 'Ù\205Ø¨Ù\203Ù\210',
          sector: 'Materials',
          website: 'www.mepco.biz'
      },
      {
          symbol: '3007:TADAWUL',
          tickerSymbol: '3007',
          name: 'Oasis',
          name_ar: 'Ù\210Ø§ØØ©',
          sector: 'Materials',
          website: 'www.oasis.com.sa'
      },
      {
          symbol: '1201:TADAWUL',
          tickerSymbol: '1201',
          name: 'Takween',
          name_ar: 'ØªÙ\203Ù\210Ù\212Ù\206',
          sector: 'Materials',
          website: 'www.takween.com.sa'
      },
      {
          symbol: '1322:TADAWUL',
          tickerSymbol: '1322',
          name: 'AMAK',
          name_ar: 'Ø£Ù\205Ù\203',
          sector: 'Materials',
          website: 'www.amak.com.sa'
      },
      {
          symbol: '3008:TADAWUL',
          tickerSymbol: '3008',
          name: 'Alkathiri',
          name_ar: 'Ø§Ù\204Ù\203Ø«Ù\212Ø±Ù\212',
          sector: 'Materials',
          website: 'www.alkathiri.com'
      },
      {
          symbol: '2001:TADAWUL',
          tickerSymbol: '2001',
          name: 'Chemanol',
          name_ar: 'Ù\203Ù\212Ù\205Ø§Ù\206Ù\210Ù\204',
          sector: 'Materials',
          website: 'www.chemanol.com'
      },
      {
          symbol: '2010:TADAWUL',
          tickerSymbol: '2010',
          name: 'SABIC',
          name_ar: 'Ø³Ø§Ø¨Ù\203',
          sector: 'Materials',
          website: 'www.sabic.com'
      },
      {
          symbol: '2020:TADAWUL',
          tickerSymbol: '2020',
          name: 'SABIC Agri-Nutrients',
          name_ar: 'Ø³Ø§Ø¨Ù\203 Ù\204Ù\204Ù\205ØºØºÙ\212Ø§Øª Ø§Ù\204Ø²Ø±Ø§Ø¹Ù\212Ø©',
          sector: 'Materials',
          website: 'www.sabic-agri.com'
      },
      {
          symbol: '2060:TADAWUL',
          tickerSymbol: '2060',
          name: 'Tasnee',
          name_ar: 'ØªØµÙ\206Ù\212Ø¹',
          sector: 'Materials',
          website: 'www.tasnee.com'
      },
      {
          symbol: '2170:TADAWUL',
          tickerSymbol: '2170',
          name: 'Alujain',
          name_ar: 'Ø§Ù\204Ù\204Ø¬Ù\212Ù\206',
          sector: 'Materials',
          website: 'www.alujain.com'
      },
      {
          symbol: '2210:TADAWUL',
          tickerSymbol: '2210',
          name: 'Nama Chemicals',
          name_ar: 'Ù\206Ù\205Ø§Ø¡ Ù\204Ù\204Ù\203Ù\212Ù\205Ø§Ù\210Ù\212Ø§Øª',
          sector: 'Materials',
          website: 'www.nama.com.sa'
      },
      {
          symbol: '2250:TADAWUL',
          tickerSymbol: '2250',
          name: 'SIIG',
          name_ar: 'Ø§Ù\204Ù\205Ø¬Ù\205Ù\210Ø¹Ø© Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212Ø©',
          sector: 'Materials',
          website: 'www.siig.com.sa'
      },
      {
          symbol: '2290:TADAWUL',
          tickerSymbol: '2290',
          name: 'Yansab',
          name_ar: 'Ù\212Ù\206Ø³Ø§Ø¨',
          sector: 'Materials',
          website: 'www.yansab.com.sa'
      },
      {
          symbol: '2310:TADAWUL',
          tickerSymbol: '2310',
          name: 'Sipchem',
          name_ar: 'Ø³Ø¨Ù\203Ù\212Ù\205',
          sector: 'Materials',
          website: 'www.sipchem.com'
      },
      {
          symbol: '2330:TADAWUL',
          tickerSymbol: '2330',
          name: 'Advanced',
          name_ar: 'Ø§Ù\204Ù\205ØªÙ\202Ø¯Ù\205Ø©',
          sector: 'Materials',
          website: 'www.advancedpetrochem.com'
      },
      {
          symbol: '2350:TADAWUL',
          tickerSymbol: '2350',
          name: 'Saudi Kayan',
          name_ar: 'Ù\203Ù\212Ø§Ù\206 Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212Ø©',
          sector: 'Materials',
          website: 'www.saudikayan.com'
      },
      {
          symbol: '2223:TADAWUL',
          tickerSymbol: '2223',
          name: 'Luberef',
          name_ar: 'Ù\204Ù\210Ø¨Ø±Ù\212Ù\201',
          sector: 'Energy',
          website: 'www.luberef.com'
      },
      {
          symbol: '1301:TADAWUL',
          tickerSymbol: '1301',
          name: 'Aslak',
          name_ar: 'Ø£Ø³Ù\204Ø§Ù\203',
          sector: 'Materials',
          website: 'www.aslak.com.sa'
      },
      {
          symbol: '1320:TADAWUL',
          tickerSymbol: '1320',
          name: 'SSP',
          name_ar: 'Ø³Ø¨Ù\203Ù\212Ù\205',
          sector: 'Materials',
          website: 'www.ssp.com.sa'
      },
      {
          symbol: '2090:TADAWUL',
          tickerSymbol: '2090',
          name: 'NGC',
          name_ar: 'ØºØ§Ø²',
          sector: 'Materials',
          website: 'www.ngc.com.sa'
      },
      {
          symbol: '2200:TADAWUL',
          tickerSymbol: '2200',
          name: 'APC',
          name_ar: 'Ø§Ù\204Ø£Ø³Ù\205Ø¯Ø© Ø§Ù\204Ø¹Ø±Ø¨Ù\212Ø©',
          sector: 'Materials',
          website: 'www.apc.com.sa'
      },
      {
          symbol: '2240:TADAWUL',
          tickerSymbol: '2240',
          name: 'Zamil Industrial',
          name_ar: ’Ø§Ù\204Ø²Ø§Ù\205Ù\204 Ù\204Ù\204ØµÙ\206Ø§Ø¹Ø©’,
          sector: ’Materials’,
          website: ’www.zamilindustrial.com’
      },
      {
          symbol: ’2360:TADAWUL’,
          tickerSymbol: ’2360’,
          name: ’SVCP’,
          name_ar: ’Ø§Ù\204Ù\201Ø®Ø§Ø±Ù\212Ø©’,
          sector: ’Materials’,
          website: ’www.svcp.com.sa’
      },
      {
          symbol: ’1304:TADAWUL’,
          tickerSymbol: ’1304’,
          name: ’Alyamamah Steel’,
          name_ar: ’Ø§Ù\204Ù\212Ù\205Ø§Ù\205Ø© Ù\204Ù\204Ø-Ø¯Ù\212Ø¯’,
          sector: ’Materials’,
          website: ’www.alyamamahsteel.com’
      },
      {
          symbol: ’1321:TADAWUL’,
          tickerSymbol: ’1321’,
          name: ’East Pipes’,
          name_ar: ’Ø£Ù\206Ø§Ø¨Ù\212Ø¨ Ø§Ù\204Ø´Ø±Ù\202’,
          sector: ’Materials’,
          website: ’www.eastpipes.com’
      },
      {
          symbol: ’1210:TADAWUL’,
          tickerSymbol: ’1210’,
          name: ’BCI’,
          name_ar: ’Ø§Ù\204Ù\203Ù\212Ù\205Ù\212Ø§Ø¦Ù\212Ø© Ø§Ù\204Ø£Ø³Ø§Ø³Ù\212Ø©’,
          sector: ’Materials’,
          website: ’www.bci.com.sa’
      },
      {
          symbol: ’1211:TADAWUL’,
          tickerSymbol: ’1211’,
          name: ’Maaden’,
          name_ar: ’Ù\205Ø¹Ø§Ø¯Ù\206’,
          sector: ’Materials’,
          website: ’www.maaden.com.sa’
      },
      {
          symbol: ’2150:TADAWUL’,
          tickerSymbol: ’2150’,
          name: ’Zoujaj’,
          name_ar: ’Ø§Ù\204Ø²Ø¬Ø§Ø¬’,
          sector: ’Materials’,
          website: ’www.zoujaj.com’
      },
      {
          symbol: ’2180:TADAWUL’,
          tickerSymbol: ’2180’,
          name: ’FIPCO’,
          name_ar: ’Ù\201Ù\212Ø¨Ù\203Ù\210’,
          sector: ’Materials’,
          website: ’www.fipco.com.sa’
      },
      {
          symbol: ’2220:TADAWUL’,
          tickerSymbol: ’2220’,
          name: ’Maadaniyah’,
          name_ar: ’Ù\205Ø¹Ø§Ø¯Ù\206’,
          sector: ’Materials’,
          website: ’www.maadaniyah.com’
      },
      {
          symbol: ’2300:TADAWUL’,
          tickerSymbol: ’2300’,
          name: ’SPM’,
          name_ar: ’Ø§Ù\204Ø³Ø¨Ù\203’,
          sector: ’Materials’,
          website: ’www.spm.com.sa’
      },
      {
          symbol: ’3002:TADAWUL’,
          tickerSymbol: ’3002’,
          name: ’Najran Cement’,
          name_ar: ’Ø£Ø³Ù\205Ù\206Øª Ù\206Ø¬Ø±Ø§Ù\206’,
          sector: ’Materials’,
          website: ’www.najrancement.com’
      },
      {
          symbol: ’3003:TADAWUL’,
          tickerSymbol: ’3003’,
          name: ’City Cement’,
          name_ar: ’Ø£Ø³Ù\205Ù\206Øª Ø§Ù\204Ù\205Ø¯Ù\212Ù\206Ø©’,
          sector: ’Materials’,
          website: ’www.citycement.com.sa’
      },
      {
          symbol: ’3004:TADAWUL’,
          tickerSymbol: ’3004’,
          name: ’Northern Cement’,
          name_ar: ’Ø£Ø³Ù\205Ù\206Øª Ø§Ù\204Ø´Ù\205Ø§Ù\204Ù\212Ø©’,
          sector: ’Materials’,
          website: ’www.northerncement.com’
      },
      {
          symbol: ’3010:TADAWUL’,
          tickerSymbol: ’3010’,
          name: ’ACC’,
          name_ar: ’Ø£Ø³Ù\205Ù\206Øª Ø§Ù\204Ø¹Ø±Ø¨Ù\212Ø©’,
          sector: ’Materials’,
          website: ’www.arabiacement.com.sa’
      },
      {
          symbol: ’3020:TADAWUL’,
          tickerSymbol: ’3020’,
          name: ’YC’,
          name_ar: ’Ø£Ø³Ù\205Ù\206Øª Ù\212Ù\206Ø¨Ø¹’,
          sector: ’Materials’,
          website: ’www.yanbu-cement.com’
      },
      {
          symbol: ’3030:TADAWUL’,
          tickerSymbol: ’3030’,
          name: ’Saudi Cement’,
          name_ar: ’Ø£Ø³Ù\205Ù\206Øª Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212Ø©’,
          sector: ’Materials’,
          website: ’www.saudicement.com.sa’
      },
      {
          symbol: ’3040:TADAWUL’,
          tickerSymbol: ’3040’,
          name: ’QACCO’,
          name_ar: ’Ø£Ø³Ù\205Ù\206Øª Ø§Ù\204Ù\202ØµÙ\212Ù\205’,
          sector: ’Materials’,
          website: ’www.qcc.com.sa’
      },
      {
          symbol: ’3050:TADAWUL’,
          tickerSymbol: ’3050’,
          name: ’SPCC’,
          name_ar: ’Ø£Ø³Ù\205Ù\206Øª Ø§Ù\204Ø¬Ù\206Ù\210Ø¨’,
          sector: ’Materials’,
          website: ’www.spcc.com.sa’
      },
      {
          symbol: ’3060:TADAWUL’,
          tickerSymbol: ’3060’,
          name: ’YCC’,
          name_ar: ’Ø£Ø³Ù\205Ù\206Øª Ù\212Ù\206Ø¨Ø¹’,
          sector: ’Materials’,
          website: ’www.yanbu-cement.com’
      },
      {
          symbol: ’3080:TADAWUL’,
          tickerSymbol: ’3080’,
          name: ’EPCCO’,
          name_ar: ’Ø£Ø³Ù\205Ù\206Øª Ø§Ù\204Ø´Ø±Ù\202Ù\212Ø©’,
          sector: ’Materials’,
          website: ’www.epcco.com.sa’
      },
      {
          symbol: ’3090:TADAWUL’,
          tickerSymbol: ’3090’,
          name: ’TCC’,
          name_ar: ’Ø£Ø³Ù\205Ù\206Øª ØªØ¨Ù\210Ù\203’,
          sector: ’Materials’,
          website: ’www.tcc.sa’
      },
      {
          symbol: ’3091:TADAWUL’,
          tickerSymbol: ’3091’,
          name: ’Jouf Cement’,
          name_ar: ’Ø£Ø³Ù\205Ù\206Øª Ø§Ù\204Ø¬Ù\210Ù\201’,
          sector: ’Materials’,
          website: ’www.joufcement.com’
      },
      {
          symbol: '3005:TADAWUL',
          tickerSymbol: '3005',
          name: 'UACC',
          name_ar: 'Ø£Ø³Ù\205Ù\206Øª Ø§Ù\204Ù\205ØªØ-Ø¯Ø©',
          sector: 'Materials',
          website: 'www.uacc.com.sa'
      },
      {
          symbol: '3092:TADAWUL',
          tickerSymbol: '3092',
          name: 'Riyadh Cement',
          name_ar: 'Ø£Ø³Ù\205Ù\206Øª Ø§Ù\204Ø±Ù\212Ø§Ø¶',
          sector: 'Materials',
          website: 'www.riyadhcement.com.sa'
      },
      {
          symbol: '4142:TADAWUL',
          tickerSymbol: '4142',
          name: 'Riyadh Cables',
          name_ar: 'Ø§Ù\204Ù\203Ø§Ø¨Ù\204Ø§Øª Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212Ø©',
          sector: 'Capital Goods',
          website: 'www.riyadh-cables.com'
      },
      {
          symbol: '2160:TADAWUL',
          tickerSymbol: '2160',
          name: 'Amiantit',
          name_ar: 'Ø£Ù\205Ù\212Ø§Ù\206ØªÙ\212Øª',
          sector: 'Capital Goods',
          website: 'www.amiantit.com'
      },
      {
          symbol: '4140:TADAWUL',
          tickerSymbol: '4140',
          name: 'SIECO',
          name_ar: 'Ø³Ù\212Ø³Ù\203Ù\210',
          sector: 'Capital Goods',
          website: 'www.sieco.com.sa'
      },
      {
          symbol: '1212:TADAWUL',
          tickerSymbol: '1212',
          name: 'Astra Industrial',
          name_ar: 'Ù\205Ø¬Ù\205Ù\210Ø¹Ø© Ø£Ø³ØªØ±Ø§ Ø§Ù\204ØµÙ\206Ø§Ø¹Ù\212Ø©',
          sector: 'Capital Goods',
          website: 'www.astraindustrial.com'
      },
      {
          symbol: '1302:TADAWUL',
          tickerSymbol: '1302',
          name: 'Bawan',
          name_ar: 'Ø¨Ù\210Ø§Ù\206',
          sector: 'Capital Goods',
          website: 'www.bawan.com.sa'
      },
      {
          symbol: '2370:TADAWUL',
          tickerSymbol: '2370',
          name: 'MESC',
          name_ar: 'Ù\205Ø³Ù\203',
          sector: 'Capital Goods',
          website: 'www.mesc.com.sa'
      },
      {
          symbol: '1303:TADAWUL',
          tickerSymbol: '1303',
          name: 'EIC',
          name_ar: 'Ø¥Ù\212Ù\203',
          sector: 'Capital Goods',
          website: 'www.eic.com.sa'
      },
      {
          symbol: '1214:TADAWUL',
          tickerSymbol: '1214',
          name: 'Shaker',
          name_ar: 'Ø´Ø§Ù\203Ø±',
          sector: 'Capital Goods',
          website: 'www.shaker.com.sa'
      },
      {
          symbol: '2320:TADAWUL',
          tickerSymbol: '2320',
          name: 'Albabtain',
          name_ar: 'Ø§Ù\204Ø¨Ø§Ø¨Ø·Ù\212Ù\206',
          sector: 'Capital Goods',
          website: 'www.albabtain.com.sa'
      },
      {
          symbol: '2110:TADAWUL',
          tickerSymbol: '2110',
          name: 'Saudi Cable',
          name_ar: 'Ø§Ù\204Ù\203Ø§Ø¨Ù\204Ø§Øª Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212Ø©',
          sector: 'Capital Goods',
          website: 'www.saudicable.com'
      },
      {
          symbol: '2040:TADAWUL',
          tickerSymbol: '2040',
          name: 'Saudi Ceramics',
          name_ar: 'Ø§Ù\204Ø®Ø²Ù\201 Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212',
          sector: 'Capital Goods',
          website: 'www.saudiceramics.com'
      },
      {
          symbol: '4110:TADAWUL',
          tickerSymbol: '4110',
          name: 'BATIC',
          name_ar: 'Ø¨Ø§ØªÙ\203',
          sector: 'Capital Goods',
          website: 'www.batic.com.sa'
      },
      {
          symbol: '4141:TADAWUL',
          tickerSymbol: '4141',
          name: 'Alomran',
          name_ar: 'Ø§Ù\204Ø¹Ù\205Ø±Ø§Ù\206',
          sector: 'Capital Goods',
          website: 'www.alomran.com.sa'
      },
      {
          symbol: '4143:TADAWUL',
          tickerSymbol: '4143',
          name: 'TALCO',
          name_ar: 'ØªØ§Ù\204Ù\203Ù\210',
          sector: 'Capital Goods',
          website: 'www.talco.com.sa'
      },
      {
          symbol: '4270:TADAWUL',
          tickerSymbol: '4270',
          name: 'SPPC',
          name_ar: 'Ø§Ù\204Ù\205Ø•Ø¨Ø¹Ø©',
          sector: 'Commercial & Professional Services',
          website: 'www.sppc.com.sa'
      },
      {
          symbol: '1831:TADAWUL',
          tickerSymbol: '1831',
          name: 'Maharah',
          name_ar: 'Ù\205Ù\207Ø§Ø±Ø©',
          sector: 'Commercial & Professional Services',
          website: 'www.maharah.com.sa'
      },
      {
          symbol: '6004:TADAWUL',
          tickerSymbol: '6004',
          name: 'Catrion',
          name_ar: 'Ù\203Ø§ØªØ±Ù\212Ù\210Ù\206',
          sector: 'Commercial & Professional Services',
          website: 'www.catrion.com.sa'
      },
      {
          symbol: '1833:TADAWUL',
          tickerSymbol: '1833',
          name: 'Almawarid',
          name_ar: 'Ø§Ù\204Ù\205Ù\210Ø§Ø±Ø¯',
          sector: 'Commercial & Professional Services',
          website: 'www.almawarid.com.sa'
      },
      {
          symbol: '1832:TADAWUL',
          tickerSymbol: '1832',
          name: 'Sadr',
          name_ar: 'ØµØ¯Ø±',
          sector: 'Commercial & Professional Services',
          website: 'www.sadr.com.sa'
      },
      {
          symbol: '1834:TADAWUL',
          tickerSymbol: '1834',
          name: 'Smasco',
          name_ar: ’Ø³Ù\205Ø§Ø³Ù\203Ù\210’,
          sector: ’Commercial & Professional Services’,
          website: ’www.smasco.com.sa’
      },
      {
          symbol: ’4040:TADAWUL’,
          tickerSymbol: ’4040’,
          name: ’SAPTCO’,
          name_ar: ’Ø³Ø§Ø¨ØªÙ\203Ù\210’,
          sector: ’Transportation’,
          website: ’www.saptco.com.sa’
      },
      {
          symbol: ’4260:TADAWUL’,
          tickerSymbol: ’4260’,
          name: ’Budget Saudi’,
          name_ar: ’Ø¨Ø¯Ø¬Øª Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212Ø©’,
          sector: ’Transportation’,
          website: ’www.budgetsaudi.com’
      },
      {
          symbol: ’4031:TADAWUL’,
          tickerSymbol: ’4031’,
          name: ’SGS’,
          name_ar: ’Ø§Ù\204Ø®Ø¯Ù\205Ø§Øª Ø§Ù\204Ø£Ø±Ø¶Ù\212Ø©’,
          sector: ’Transportation’,
          website: ’www.sgs.com.sa’
      },
      {
          symbol: ’4261:TADAWUL’,
          tickerSymbol: ’4261’,
          name: ’Theeb’,
          name_ar: ’Ø°Ù\212Ø¨’,
          sector: ’Transportation’,
          website: ’www.theeb.com.sa’
      },
      {
          symbol: ’4262:TADAWUL’,
          tickerSymbol: ’4262’,
          name: ’Lumi’,
          name_ar: ’Ù\204Ù\210Ù\205Ù\212’,
          sector: ’Transportation’,
          website: ’www.lumi.com.sa’
      },
      {
          symbol: ’4263:TADAWUL’,
          tickerSymbol: ’4263’,
          name: ’SAL’,
          name_ar: ’Ø³Ø§Ù\204’,
          sector: ’Transportation’,
          website: ’www.sal.com.sa’
      },
      {
          symbol: ’2190:TADAWUL’,
          tickerSymbol: ’2190’,
          name: ’SISCO Holding’,
          name_ar: ’Ø³Ù\212Ø³Ù\203Ù\210’,
          sector: ’Transportation’,
          website: ’www.sisco.com.sa’
      },
      {
          symbol: ’4180:TADAWUL’,
          tickerSymbol: ’4180’,
          name: ’Fitaihi Group’,
          name_ar: ’Ù\205Ø¬Ù\205Ù\210Ø¹Ø© Ù\201ØªÙ\212ØÙ\212’,
          sector: ’Consumer Durables & Apparel’,
          website: ’www.fitaihi.com.sa’
      },
      {
          symbol: ’4012:TADAWUL’,
          tickerSymbol: ’4012’,
          name: ’Alaseel’,
          name_ar: ’Ø§Ù\204Ø£ØµÙ\212Ù\204’,
          sector: ’Consumer Durables & Apparel’,
          website: ’www.alaseel.com.sa’
      },
      {
          symbol: ’4011:TADAWUL’,
          tickerSymbol: ’4011’,
          name: ’Lazurde’,
          name_ar: ’Ù\204Ø§Ø²Ù\210Ø±Ø¯Ù\212’,
          sector: ’Consumer Durables & Apparel’,
          website: ’www.lazurde.com’
      },
      {
          symbol: ’2130:TADAWUL’,
          tickerSymbol: ’2130’,
          name: ’SIDC’,
          name_ar: ’Ø³Ù\212Ø¯Ù\203’,
          sector: ’Consumer Durables & Apparel’,
          website: ’www.sidc.com.sa’
      },
      {
          symbol: ’1213:TADAWUL’,
          tickerSymbol: ’1213’,
          name: ’Naseej’,
          name_ar: ’Ù\206Ø³Ù\212Ø¬’,
          sector: ’Consumer Durables & Apparel’,
          website: ’www.naseej.com.sa’
      },
      {
          symbol: ’2340:TADAWUL’,
          tickerSymbol: ’2340’,
          name: ’Artex’,
          name_ar: ’Ø£Ø±ØªÙ\203Ø³’,
          sector: ’Consumer Durables & Apparel’,
          website: ’www.artex.com.sa’
      },
      {
          symbol: ’1810:TADAWUL’,
          tickerSymbol: ’1810’,
          name: ’Seera’,
          name_ar: ’Ø³Ù\212Ø±Ø§’,
          sector: ’Consumer Services’,
          website: ’www.seera.sa’
      },
      {
          symbol: ’4170:TADAWUL’,
          tickerSymbol: ’4170’,
          name: ’TECO’,
          name_ar: ’ØªÙ\212Ù\203Ù\210’,
          sector: ’Consumer Services’,
          website: ’www.teco.com.sa’
      },
      {
          symbol: ’1820:TADAWUL’,
          tickerSymbol: ’1820’,
          name: ’Alhokair Group’,
          name_ar: ’Ù\205Ø¬Ù\205Ù\210Ø¹Ø© Ø§Ù\204Ø–Ù\203Ù\212Ø±’,
          sector: ’Consumer Services’,
          website: ’www.alhokair.com.sa’
      },
      {
          symbol: ’6002:TADAWUL’,
          tickerSymbol: ’6002’,
          name: ’Herfy Foods’,
          name_ar: ’Ù\207Ø±Ù\201Ù\212 Ù\204Ù\204Ø£ØºØºÙ\212Ø©’,
          sector: ’Consumer Services’,
          website: ’www.herfy.com’
      },
      {
          symbol: ’6016:TADAWUL’,
          tickerSymbol: ’6016’,
          name: ’Burgerizzr’,
          name_ar: ’Ø¨Ø±ØºØ±Ø§Ù\212Ø²Ø²Ø±’,
          sector: ’Consumer Services’,
          website: ’www.burgerizzr.com’
      },
      {
          symbol: ’6013:TADAWUL’,
          tickerSymbol: ’6013’,
          name: ’DWF’,
          name_ar: ’Ø¯Ù\210Ù\201’,
          sector: ’Consumer Services’,
          website: ’www.dwf.com.sa’
      },
      {
          symbol: ’6015:TADAWUL’,
          tickerSymbol: ’6015’,
          name: ’Americana’,
          name_ar: ’Ø£Ù\205Ø±Ù\212Ù\203Ø§Ù\206Ø§’,
          sector: ’Consumer Services’,
          website: ’www.americana.com.sa’
      },
      {
          symbol: ’1830:TADAWUL’,
          tickerSymbol: ’1830’,
          name: ’Leejam Sports’,
          name_ar: ’Ù\204Ø¬Ø§Ù\205 Ù\204Ù\204Ø±Ù\212Ø§Ø¶Ø©’,
          sector: ’Consumer Services’,
          website: ’www.leejam.com.sa’
      },
      {
          symbol: '4291:TADAWUL',
          tickerSymbol: '4291',
          name: 'NCLE',
          name_ar: 'Ù\206Ù\203Ù\204Ù\212',
          sector: 'Consumer Services',
          website: 'www.ncle.com.sa'
      },
      {
          symbol: '4292:TADAWUL',
          tickerSymbol: '4292',
          name: 'ATAA',
          name_ar: 'Ø¹Ø•Ø§Ø¡',
          sector: 'Consumer Services',
          website: 'www.ataa.com.sa'
      },
      {
          symbol: '6012:TADAWUL',
          tickerSymbol: '6012',
          name: 'Raydan',
          name_ar: 'Ø±Ø§Ù\212Ø¯Ø§Ù\206',
          sector: 'Consumer Services',
          website: 'www.raydan.com.sa'
      },
      {
          symbol: '6014:TADAWUL',
          tickerSymbol: '6014',
          name: 'Alamar',
          name_ar: 'Ø§Ù\204Ù\205Ø±',
          sector: 'Consumer Services',
          website: 'www.alamar.com.sa'
      },
      {
          symbol: '4290:TADAWUL',
          tickerSymbol: '4290',
          name: 'Alkhaleej Trng',
          name_ar: 'Ø§Ù\204Ø®Ù\204Ù\212Ø¬ Ù\204Ù\204ØªØ¯Ø±Ù\212Ø¨',
          sector: 'Consumer Services',
          website: 'www.alkhaleej.com.sa'
      },
      {
          symbol: '4070:TADAWUL',
          tickerSymbol: '4070',
          name: 'TAPRCO',
          name_ar: 'Ø•Ø§Ø¨Ù\203Ù\210',
          sector: 'Media and Entertainment',
          website: 'www.taprco.com.sa'
      },
      {
          symbol: '4210:TADAWUL',
          tickerSymbol: '4210',
          name: 'SRMG',
          name_ar: 'Ø§Ù\204Ù\205Ø¬Ù\205Ù\210Ø¹Ø© Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212Ø©',
          sector: 'Media and Entertainment',
          website: 'www.srmg.com'
      },
      {
          symbol: '4071:TADAWUL',
          tickerSymbol: '4071',
          name: 'Alarabiya',
          name_ar: 'Ø§Ù\204Ø¹Ø±Ø¨Ù\212Ø©',
          sector: 'Media and Entertainment',
          website: 'www.alarabiya.com.sa'
      },
      {
          symbol: '4072:TADAWUL',
          tickerSymbol: '4072',
          name: 'MBC Group',
          name_ar: 'Ù\205Ø¬Ù\205Ù\210Ø¹Ø© Ø¥Ù\205 Ø¨Ù\212 Ø³Ù\212',
          sector: 'Media and Entertainment',
          website: 'www.mbc.net'
      },
      {
          symbol: '4240:TADAWUL',
          tickerSymbol: '4240',
          name: 'Cenomi Retail',
          name_ar: 'Ø³Ù\212Ù\206Ù\210Ù\205Ù\212 Ù\204Ù\204Ø¨Ù\212Ø¹ Ø¨Ø§Ù\204ØªØ¬Ø²Ø¦Ø©',
          sector: 'Consumer Discretionary Distribution & Retail',
          website: 'www.cenomi.com'
      },
      {
          symbol: '4190:TADAWUL',
          tickerSymbol: '4190',
          name: 'Jarir',
          name_ar: 'Ø¬Ø±Ù\212Ø±',
          sector: 'Consumer Discretionary Distribution & Retail',
          website: ’www.jarir.com’
      },
      {
          symbol: ’4003:TADAWUL’,
          tickerSymbol: ’4003’,
          name: ’Extra’,
          name_ar: ’Ø¥Ù\203Ø³ØªØ±Ø§’,
          sector: ’Consumer Discretionary Distribution & Retail’,
          website: ’www.extra.com.sa’
      },
      {
          symbol: ’4050:TADAWUL’,
          tickerSymbol: ’4050’,
          name: ’SASCO’,
          name_ar: ’Ø³Ø§Ø³Ù\203Ù\210’,
          sector: ’Consumer Discretionary Distribution & Retail’,
          website: ’www.sasco.com.sa’
      },
      {
          symbol: ’4192:TADAWUL’,
          tickerSymbol: ’4192’,
          name: ’Alsaif Gallery’,
          name_ar: ’Ù\205Ø¹Ø±Ø¶ Ø§Ù\204Ø³Ù\212Ù\201’,
          sector: ’Consumer Discretionary Distribution & Retail’,
          website: ’www.alsaifgallery.com’
      },
      {
          symbol: ’4051:TADAWUL’,
          tickerSymbol: ’4051’,
          name: ’Baazeem’,
          name_ar: ’Ø¨Ø§Ø²Ù\212Ù\205’,
          sector: ’Consumer Discretionary Distribution & Retail’,
          website: ’www.baazeem.com.sa’
      },
      {
          symbol: ’4191:TADAWUL’,
          tickerSymbol: ’4191’,
          name: ’Abo Moati’,
          name_ar: ’Ø£Ø¨Ù\210 Ù\205Ø¹Ø•Ù\212’,
          sector: ’Consumer Discretionary Distribution & Retail’,
          website: ’www.abomoati.com.sa’
      },
      {
          symbol: ’4008:TADAWUL’,
          tickerSymbol: ’4008’,
          name: ’SACO’,
          name_ar: ’Ø³Ø§Ù\203Ù\210’,
          sector: ’Consumer Discretionary Distribution & Retail’,
          website: ’www.saco.com.sa’
      },
      {
          symbol: ’4001:TADAWUL’,
          tickerSymbol: ’4001’,
          name: ’A.Othaim Market’,
          name_ar: ’Ø£Ø³Ù\210Ø§Ù\202 Ø§Ù\204Ø¹Ø«Ù\212Ù\205’,
          sector: ’Consumer Staples Distribution & Retail’,
          website: ’www.othaimmarkets.com’
      },
      {
          symbol: ’4161:TADAWUL’,
          tickerSymbol: ’4161’,
          name: ’Bindawood’,
          name_ar: ’Ø¨Ù\206Ø¯Ø©’,
          sector: ’Consumer Staples Distribution & Retail’,
          website: ’www.bindawood.com’
      },
      {
          symbol: ’4162:TADAWUL’,
          tickerSymbol: ’4162’,
          name: ’Almunajem’,
          name_ar: ’Ø§Ù\204Ù\205Ù\206Ø¬Ù\205’,
          sector: ’Consumer Staples Distribution & Retail’,
          website: ’www.almunajem.com’
      },
      {
          symbol: ’4164:TADAWUL’,
          tickerSymbol: ’4164’,
          name: ’Nahdi’,
          name_ar: ’Ø§Ù\204Ù\206Ù\207Ø¯Ù\212’,
          sector: ’Consumer Staples Distribution & Retail’,
          website: ’www.nahdi.sa’
      },
      {
          symbol: ’4163:TADAWUL’,
          tickerSymbol: ’4163’,
          name: ’Aldawaa’,
          name_ar: 'Ø§Ù\204Ø¯Ù\210Ø§Ø¡',
          sector: 'Consumer Staples Distribution & Retail',
          website: 'www.aldawaa.com.sa'
      },
      {
          symbol: '4160:TADAWUL',
          tickerSymbol: '4160',
          name: 'Thimar',
          name_ar: 'Ø«Ù\205Ø§Ø±',
          sector: 'Consumer Staples Distribution & Retail',
          website: 'www.thimar.com.sa'
      },
      {
          symbol: '4006:TADAWUL',
          tickerSymbol: '4006',
          name: 'Farm Superstores',
          name_ar: 'Ø£Ø³Ù\210Ø§Ù\202 Ø§Ù\204Ù\205Ø²Ø±Ø¹Ø©',
          sector: 'Consumer Staples Distribution & Retail',
          website: 'www.farm.com.sa'
      },
      {
          symbol: '4061:TADAWUL',
          tickerSymbol: '4061',
          name: 'Anaam Holding',
          name_ar: 'Ø£Ù\206Ø¹Ø§Ù\205',
          sector: 'Consumer Staples Distribution & Retail',
          website: 'www.anaam.com.sa'
      },
      {
          symbol: '6001:TADAWUL',
          tickerSymbol: '6001',
          name: 'HB',
          name_ar: 'Ù\207Ø¨',
          sector: 'Food & Beverages',
          website: 'www.hb.com.sa'
      },
      {
          symbol: '2282:TADAWUL',
          tickerSymbol: '2282',
          name: 'Naqi',
          name_ar: 'Ù\206Ù\202Ù\212',
          sector: 'Food & Beverages',
          website: 'www.naqi.com.sa'
      },
      {
          symbol: '2283:TADAWUL',
          tickerSymbol: '2283',
          name: 'First Mills',
          name_ar: 'Ø§Ù\204Ù\205Ø•Ø§Ø–Ù\206 Ø§Ù\204Ø£Ù\210Ù\204Ù\211',
          sector: 'Food & Beverages',
          website: 'www.firstmills.com.sa'
      },
      {
          symbol: '2284:TADAWUL',
          tickerSymbol: '2284',
          name: 'Modern Mills',
          name_ar: 'Ø§Ù\204Ù\205Ø•Ø§Ø–Ù\206 Ø§Ù\204Ø–Ø¯Ù\212Ø«Ø©',
          sector: 'Food & Beverages',
          website: 'www.modernmills.com.sa'
      },
      {
          symbol: '2285:TADAWUL',
          tickerSymbol: '2285',
          name: 'Arabian Mills',
          name_ar: 'Ø§Ù\204Ù\205Ø•Ø§Ø–Ù\206 Ø§Ù\204Ø¹Ø±Ø¨Ù\212Ø©',
          sector: 'Food & Beverages',
          website: 'www.arabianmills.com.sa'
      },
      {
          symbol: '2286:TADAWUL',
          tickerSymbol: '2286',
          name: 'Fourth Milling',
          name_ar: 'Ø§Ù\204Ù\205Ø•Ø§Ø–Ù\206 Ø§Ù\204Ø±Ø§Ø¨Ø¹Ø©',
          sector: 'Food & Beverages',
          website: 'www.fourthmilling.com.sa'
      },
      {
          symbol: '6010:TADAWUL',
          tickerSymbol: '6010',
          name: 'NADEC',
          name_ar: 'Ù\206Ø§Ø¯Ù\203',
          sector: 'Food & Beverages',
          website: 'www.nadec.com.sa'
      },
      {
          symbol: '6020:TADAWUL',
          tickerSymbol: ’6020’,
          name: ’GACO’,
          name_ar: ’Ø¬Ù\203Ù\210’,
          sector: ’Food & Beverages’,
          website: ’www.gaco.com.sa’
      },
      {
          symbol: ’6040:TADAWUL’,
          tickerSymbol: ’6040’,
          name: ’TADCO’,
          name_ar: ’ØªØ§Ø¯Ù\203Ù\210’,
          sector: ’Food & Beverages’,
          website: ’www.tadco.com.sa’
      },
      {
          symbol: ’6050:TADAWUL’,
          tickerSymbol: ’6050’,
          name: ’SFICO’,
          name_ar: ’ØµÙ\201Ù\203Ù\210’,
          sector: ’Food & Beverages’,
          website: ’www.sfico.com.sa’
      },
      {
          symbol: ’6060:TADAWUL’,
          tickerSymbol: ’6060’,
          name: ’Sharqiyah Dev’,
          name_ar: ’Ø§Ù\204ØªÙ\206Ù\205Ù\212Ø© Ø§Ù\204Ø´Ø±Ù\202Ù\212Ø©’,
          sector: ’Food & Beverages’,
          website: ’www.sharqiyahdev.com.sa’
      },
      {
          symbol: ’6070:TADAWUL’,
          tickerSymbol: ’6070’,
          name: ’Aljouf’,
          name_ar: ’Ø§Ù\204Ø¬Ù\210Ù\201’,
          sector: ’Food & Beverages’,
          website: ’www.aljouf.com.sa’
      },
      {
          symbol: ’6090:TADAWUL’,
          tickerSymbol: ’6090’,
          name: ’Jazadco’,
          name_ar: ’Ø¬Ø§Ø²Ø§Ø¯Ù\203Ù\210’,
          sector: ’Food & Beverages’,
          website: ’www.jazadco.com’
      },
      {
          symbol: ’2281:TADAWUL’,
          tickerSymbol: ’2281’,
          name: ’Tanmiah’,
          name_ar: ’ØªÙ\206Ù\205Ù\212Ø©’,
          sector: ’Food & Beverages’,
          website: ’www.tanmiah.com’
      },
      {
          symbol: ’2050:TADAWUL’,
          tickerSymbol: ’2050’,
          name: ’Savola Group’,
          name_ar: ’Ù\205Ø¬Ù\205Ù\210Ø¹Ø© ØµØ§Ù\201Ù\210Ù\204Ø§’,
          sector: ’Food & Beverages’,
          website: ’www.savola.com’
      },
      {
          symbol: ’2100:TADAWUL’,
          tickerSymbol: ’2100’,
          name: ’Wafrah’,
          name_ar: ’Ù\210Ù\201Ø±Ø©’,
          sector: ’Food & Beverages’,
          website: ’www.wafrah.com.sa’
      },
      {
          symbol: ’2270:TADAWUL’,
          tickerSymbol: ’2270’,
          name: ’SADAFCO’,
          name_ar: ’Ø³Ø¯Ø§Ù\201Ù\203Ù\210’,
          sector: ’Food & Beverages’,
          website: ’www.sadafco.com’
      },
      {
          symbol: ’2280:TADAWUL’,
          tickerSymbol: ’2280’,
          name: ’Almarai’,
          name_ar: ’Ø§Ù\204Ù\205Ø±Ø§Ø¹Ù\212’,
          sector: ’Food & Beverages’,
          website: ’www.almarai.com’
      },
      {
          symbol: ’4080:TADAWUL’,
          tickerSymbol: ’4080’,
          name: ’Sinad Holding’,
          name_ar: ’Ø³Ù\206Ø§Ø¯ Ø§Ù\204Ù\202Ø§Ø¨Ø¶Ø©’,
          sector: ’Food & Beverages’,
          website: ’www.sinad.com.sa’
      },
      {
          symbol: ’4014:TADAWUL’,
          tickerSymbol: ’4014’,
          name: ’Equipment House’,
          name_ar: ’Ø¨Ù\212Øª Ø§Ù\204Ù\205Ø¹Ø¯Ø§Øª’,
          sector: ’Health Care Equipment & Services’,
          website: ’www.equipmenthouse.com.sa’
      },
      {
          symbol: ’4017:TADAWUL’,
          tickerSymbol: ’4017’,
          name: ’Fakeeh Care’,
          name_ar: ’Ù\201Ù\202Ù\212Ù\207 Ù\203Ù\212Ø±’,
          sector: ’Health Care Equipment & Services’,
          website: ’www.fakeeh.care’
      },
      {
          symbol: ’2140:TADAWUL’,
          tickerSymbol: ’2140’,
          name: ’Ayyan’,
          name_ar: ’Ø¹Ù\212Ø§Ù\206’,
          sector: ’Health Care Equipment & Services’,
          website: ’www.ayyan.com.sa’
      },
      {
          symbol: ’4013:TADAWUL’,
          tickerSymbol: ’4013’,
          name: ’Sulaiman Alhabib’,
          name_ar: ’Ø³Ù\204Ù\212Ù\205Ø§Ù\206 Ø§Ù\204Ø-Ø¨Ù\212Ø¨’,
          sector: ’Health Care Equipment & Services’,
          website: ’www.hmg.com.sa’
      },
      {
          symbol: ’4005:TADAWUL’,
          tickerSymbol: ’4005’,
          name: ’Care’,
          name_ar: ’Ø±Ø¹Ø§Ù\212Ø©’,
          sector: ’Health Care Equipment & Services’,
          website: ’www.care.com.sa’
      },
      {
          symbol: ’4002:TADAWUL’,
          tickerSymbol: ’4002’,
          name: ’Mouwasat’,
          name_ar: ’Ø§Ù\204Ù\205Ù\210Ø§Ø³Ø§Ø©’,
          sector: ’Health Care Equipment & Services’,
          website: ’www.mouwasat.com’
      },
      {
          symbol: ’4004:TADAWUL’,
          tickerSymbol: ’4004’,
          name: ’Dallah Health’,
          name_ar: ’Ø¯Ù\204Ù\207 Ø§Ù\204ØµØ-Ù\212Ø©’,
          sector: ’Health Care Equipment & Services’,
          website: ’www.dallah.com.sa’
      },
      {
          symbol: ’4007:TADAWUL’,
          tickerSymbol: ’4007’,
          name: ’Alhammadi’,
          name_ar: ’Ø§Ù\204Ø-Ù\205Ø¯Ù\212’,
          sector: ’Health Care Equipment & Services’,
          website: ’www.alhammadi.com.sa’
      },
      {
          symbol: ’4009:TADAWUL’,
          tickerSymbol: ’4009’,
          name: ’Saudi German Health’,
          name_ar: ’Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212 Ø§Ù\204Ø£Ù\204Ù\205Ø§Ù\206Ù\212 Ø§Ù\204ØµØ-Ù\212Ø©’,
          sector: ’Health Care Equipment & Services’,
          website: ’www.sgh.com.sa’
      },
      {
          symbol: ’2230:TADAWUL’,
          tickerSymbol: ’2230’,
          name: ’Chemical’,
          name_ar: ’Ø§Ù\204Ù\203Ù\212Ù\205Ù\212Ø§Ø¦Ù\212Ø©’,
          sector: ’Health Care Equipment & Services’,
          website: ’www.chemical.com.sa’
      },
      {
          symbol: ’2070:TADAWUL’,
          tickerSymbol: ’2070’,
          name: ’SPIMACO’,
          name_ar: ’Ø³Ø¨Ù\212Ù\205Ø§Ù\203Ù\210’,
          sector: ’Pharma, Biotech & Life Sciences’,
          website: ’www.spimaco.com.sa’
      },
      {
          symbol: ’4015:TADAWUL’,
          tickerSymbol: ’4015’,
          name: ’Jamjoom Pharma’,
          name_ar: ’Ø¬Ù\205Ø¬Ù\210Ù\205 Ù\201Ø§Ø±Ù\205Ø§’,
          sector: ’Pharma, Biotech & Life Sciences’,
          website: ’www.jamjoompharma.com’
      },
      {
          symbol: ’4016:TADAWUL’,
          tickerSymbol: ’4016’,
          name: ’Avalon Pharma’,
          name_ar: ’Ø£Ù\201Ø§Ù\204Ù\210Ù\206 Ù\201Ø§Ø±Ù\205Ø§’,
          sector: ’Pharma, Biotech & Life Sciences’,
          website: ’www.avalonpharma.com’
      },
      {
          symbol: ’1010:TADAWUL’,
          tickerSymbol: ’1010’,
          name: ’Riyad Bank’,
          name_ar: ’Ø¨Ù\206Ù\203 Ø§Ù\204Ø±Ù\212Ø§Ø¶’,
          sector: ’Banks’,
          website: ’www.riyadbank.com’
      },
      {
          symbol: ’1020:TADAWUL’,
          tickerSymbol: ’1020’,
          name: ’BJAZ’,
          name_ar: ’Ø¨Ù\206Ù\203 Ø§Ù\204Ø¬Ø²Ù\212Ø±Ø©’,
          sector: ’Banks’,
          website: ’www.bjaz.com.sa’
      },
      {
          symbol: ’1030:TADAWUL’,
          tickerSymbol: ’1030’,
          name: ’SAIB’,
          name_ar: ’Ø§Ù\204Ø¨Ù\206Ù\203 Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212 Ù\204Ù\204Ø§Ø³ØªØ«Ù\205Ø§Ø±’,
          sector: ’Banks’,
          website: ’www.saib.com.sa’
      },
      {
          symbol: ’1050:TADAWUL’,
          tickerSymbol: ’1050’,
          name: ’BSF’,
          name_ar: ’Ø§Ù\204Ø¨Ù\206Ù\203 Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212 Ø§Ù\204Ù\201Ø±Ù\206Ø³Ù\212’,
          sector: ’Banks’,
          website: ’www.bsf.com.sa’
      },
      {
          symbol: ’1060:TADAWUL’,
          tickerSymbol: ’1060’,
          name: ’SAB’,
          name_ar: ’Ø§Ù\204Ø¨Ù\206Ù\203 Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212 Ø§Ù\204Ø¨Ø±Ù\212Ø•Ø§Ù\206Ù\212’,
          sector: ’Banks’,
          website: ’www.sabb.com’
      },
      {
          symbol: ’1080:TADAWUL’,
          tickerSymbol: ’1080’,
          name: ’ANB’,
          name_ar: ’Ø§Ù\204Ø¨Ù\206Ù\203 Ø§Ù\204Ø¹Ø±Ø¨Ù\212 Ø§Ù\204Ù\210Ø•Ù\206Ù\212’,
          sector: ’Banks’,
          website: ’www.anb.com.sa’
      },
      {
          symbol: ’1120:TADAWUL’,
          tickerSymbol: ’1120’,
          name: ’Alrajhi’,
          name_ar: ’Ù\205ØµØ±Ù\201 Ø§Ù\204Ø±Ø§Ø¬Ø–Ù\212’,
          sector: ’Banks’,
          website: ’www.alrajhibank.com.sa’
      },
      {
          symbol: ’1140:TADAWUL’,
          tickerSymbol: ’1140’,
          name: ’Albilad’,
          name_ar: 'Ø¨Ù\206Ù\203 Ø§Ù\204Ø¨Ù\204Ø§Ø¯',
          sector: 'Banks',
          website: 'www.bankalbilad.com'
      },
      {
          symbol: '1150:TADAWUL',
          tickerSymbol: '1150',
          name: 'Alinma',
          name_ar: 'Ø¨Ù\206Ù\203 Ø§Ù\204Ø¥Ù\206Ù\205Ø§Ø¡',
          sector: 'Banks',
          website: 'www.alinma.com'
      },
      {
          symbol: '1180:TADAWUL',
          tickerSymbol: '1180',
          name: 'SNB',
          name_ar: 'Ø§Ù\204Ø¨Ù\206Ù\203 Ø§Ù\204Ø£Ù\207Ù\204Ù\212 Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212',
          sector: 'Banks',
          website: 'www.alahli.com'
      },
      {
          symbol: '1183:TADAWUL',
          tickerSymbol: '1183',
          name: 'SHL',
          name_ar: 'Ø´Ù\204Ù\207',
          sector: 'Financial Services',
          website: 'www.shl.com.sa'
      },
      {
          symbol: '1182:TADAWUL',
          tickerSymbol: '1182',
          name: 'Amlak',
          name_ar: 'Ø£Ù\205Ù\204Ø§Ù\203',
          sector: 'Financial Services',
          website: 'www.amlak.com.sa'
      },
      {
          symbol: '4081:TADAWUL',
          tickerSymbol: '4081',
          name: 'Nayifat',
          name_ar: 'Ù\206Ø§Ù\212Ù\201Ø§Øª',
          sector: 'Financial Services',
          website: 'www.nayifat.com.sa'
      },
      {
          symbol: '1111:TADAWUL',
          tickerSymbol: '1111',
          name: 'Tadawul Group',
          name_ar: 'Ù\205Ø¬Ù\205Ù\210Ø¹Ø© ØªØ¯Ø§Ù\210Ù\204',
          sector: 'Financial Services',
          website: 'www.tadawul.com.sa'
      },
      {
          symbol: '4082:TADAWUL',
          tickerSymbol: '4082',
          name: 'MRNA',
          name_ar: 'Ù\205Ø±Ù\206Ø©',
          sector: 'Financial Services',
          website: 'www.mrna.com.sa'
      },
      {
          symbol: '4130:TADAWUL',
          tickerSymbol: '4130',
          name: 'Albaha',
          name_ar: 'Ø§Ù\204Ø¨Ø§ØØ©',
          sector: 'Financial Services',
          website: 'www.albaha.com.sa'
      },
      {
          symbol: '2120:TADAWUL',
          tickerSymbol: '2120',
          name: 'SAIC',
          name_ar: 'Ø³Ù\212Ø³Ù\203Ù\210',
          sector: 'Financial Services',
          website: 'www.saic.com.sa'
      },
      {
          symbol: '4280:TADAWUL',
          tickerSymbol: '4280',
          name: 'Kingdom',
          name_ar: 'Ø§Ù\204Ù\205Ù\205Ù\204Ù\203Ø©',
          sector: 'Financial Services',
          website: 'www.kingdom.com.sa'
      },
      {
          symbol: '8313:TADAWUL',
          tickerSymbol: '8313',
          name: 'Rasan',
          name_ar: 'Ø±Ø³Ø§Ù\206',
          sector: 'Insurance',
          website: 'www.rasan.com.sa'
      },
      {
          symbol: '8010:TADAWUL',
          tickerSymbol: '8010',
          name: 'Tawuniya',
          name_ar: 'Ø§Ù\204ØªØ¹Ø§Ù\210Ù\206Ù\212Ø©',
          sector: 'Insurance',
          website: 'www.tawuniya.com.sa'
      },
      {
          symbol: '8020:TADAWUL',
          tickerSymbol: '8020',
          name: 'Malath Insurance',
          name_ar: 'Ù\205Ù\204Ø§Ø° Ù\204Ù\204ØªØ£Ù\205Ù\212Ù\206',
          sector: 'Insurance',
          website: 'www.malath.com.sa'
      },
      {
          symbol: '8030:TADAWUL',
          tickerSymbol: '8030',
          name: 'Medgulf',
          name_ar: 'Ù\205Ù\212Ø¯ØºÙ\204Ù\201',
          sector: 'Insurance',
          website: 'www.medgulf.com.sa'
      },
      {
          symbol: '8060:TADAWUL',
          tickerSymbol: '8060',
          name: 'Walaa',
          name_ar: 'Ù\210Ù\204Ø§Ø¡',
          sector: 'Insurance',
          website: 'www.walaa.com.sa'
      },
      {
          symbol: '8040:TADAWUL',
          tickerSymbol: '8040',
          name: 'Allianz SF',
          name_ar: 'Ø£Ù\204Ù\212Ø§Ù\206Ø² Ø¥Ø³ Ø¥Ù\201',
          sector: 'Insurance',
          website: 'www.allianz.com.sa'
      },
      {
          symbol: '8070:TADAWUL',
          tickerSymbol: '8070',
          name: 'Arabian Shield',
          name_ar: 'Ø§Ù\204Ø¯Ø±Ø¹ Ø§Ù\204Ø¹Ø±Ø¨Ù\212',
          sector: 'Insurance',
          website: 'www.arabianshield.com.sa'
      },
      {
          symbol: '8050:TADAWUL',
          tickerSymbol: '8050',
          name: 'Salama',
          name_ar: 'Ø³Ù\204Ø§Ù\205Ø©',
          sector: 'Insurance',
          website: 'www.salama.com.sa'
      },
      {
          symbol: '8100:TADAWUL',
          tickerSymbol: '8100',
          name: 'SAICO',
          name_ar: 'Ø³Ø§Ù\212Ù\203Ù\210',
          sector: 'Insurance',
          website: 'www.saico.com.sa'
      },
      {
          symbol: '8012:TADAWUL',
          tickerSymbol: '8012',
          name: 'Jazira Takaful',
          name_ar: 'Ø§Ù\204Ø¬Ø²Ù\212Ø±Ø© ØªÙ\203Ø§Ù\201Ù\204',
          sector: 'Insurance',
          website: 'www.jaziratakaful.com.sa'
      },
      {
          symbol: '8120:TADAWUL',
          tickerSymbol: '8120',
          name: 'Gulf Union Alahlia',
          name_ar: 'Ø§Ù\204Ø®Ù\204Ù\212Ø¬ Ø§Ù\204Ø£Ù\207Ù\204Ù\212Ø©',
          sector: 'Insurance',
          website: 'www.gulfunion.com.sa'
      },
      {
          symbol: ’8150:TADAWUL’,
          tickerSymbol: ’8150’,
          name: ’ACIG’,
          name_ar: ’Ø§Ù\204Ù\205Ø¬Ù\205Ù\210Ø¹Ø© Ø§Ù\204Ù\205ØªØ-Ø¯Ø©’,
          sector: ’Insurance’,
          website: ’www.acig.com.sa’
      },
      {
          symbol: ’8160:TADAWUL’,
          tickerSymbol: ’8160’,
          name: ’AICC’,
          name_ar: ’Ø£Ù\203Ø³Ø§’,
          sector: ’Insurance’,
          website: ’www.aicc.com.sa’
      },
      {
          symbol: ’8170:TADAWUL’,
          tickerSymbol: ’8170’,
          name: ’Aletihad’,
          name_ar: ’Ø§Ù\204Ø§ØªØ-Ø§Ø¯’,
          sector: ’Insurance’,
          website: ’www.aletihad.com.sa’
      },
      {
          symbol: ’8180:TADAWUL’,
          tickerSymbol: ’8180’,
          name: ’Alsagr Insurance’,
          name_ar: ’Ø§Ù\204ØµÙ\202Ø± Ù\204Ù\204ØªØ£Ù\205Ù\212Ù\206’,
          sector: ’Insurance’,
          website: ’www.alsagr.com.sa’
      },
      {
          symbol: ’8190:TADAWUL’,
          tickerSymbol: ’8190’,
          name: ’UCA’,
          name_ar: ’Ø§Ù\204Ù\205ØªØ-Ø¯Ø© Ù\204Ù\204ØªØ£Ù\205Ù\212Ù\206’,
          sector: ’Insurance’,
          website: ’www.uca.com.sa’
      },
      {
          symbol: ’8200:TADAWUL’,
          tickerSymbol: ’8200’,
          name: ’Saudi Re’,
          name_ar: ’Ø¥Ø¹Ø§Ø¯Ø© Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212Ø©’,
          sector: ’Insurance’,
          website: ’www.saudire.com’
      },
      {
          symbol: ’8210:TADAWUL’,
          tickerSymbol: ’8210’,
          name: ’Bupa Arabia’,
          name_ar: ’Ø¨Ù\210Ø¨Ø§ Ø§Ù\204Ø¹Ø±Ø¨Ù\212Ø©’,
          sector: ’Insurance’,
          website: ’www.bupa.com.sa’
      },
      {
          symbol: ’8230:TADAWUL’,
          tickerSymbol: ’8230’,
          name: ’Alrajhi Takaful’,
          name_ar: ’ØªÙ\203Ø§Ù\201Ù\204 Ø§Ù\204Ø±Ø§Ø¬Ø-Ù\212’,
          sector: ’Insurance’,
          website: ’www.alrajhitakaful.com’
      },
      {
          symbol: ’8240:TADAWUL’,
          tickerSymbol: ’8240’,
          name: ’Chubb’,
          name_ar: ’ØªØ´Ø¨’,
          sector: ’Insurance’,
          website: ’www.chubb.com.sa’
      },
      {
          symbol: ’8250:TADAWUL’,
          tickerSymbol: ’8250’,
          name: ’GIG’,
          name_ar: ’Ø¬Ù\212 Ø¢Ù\212 Ø¬Ù\212’,
          sector: ’Insurance’,
          website: ’www.gig.com.sa’
      },
      {
          symbol: ’8260:TADAWUL’,
          tickerSymbol: ’8260’,
          name: ’Gulf General’,
          name_ar: ’Ø§Ù\204Ø®Ù\204Ù\212Ø¬Ù\212Ø© Ø§Ù\204Ø¹Ø§Ù\205Ø©’,
          sector: ’Insurance’,
          website: 'www.gulfgeneral.com.sa'
      },
      {
          symbol: '8270:TADAWUL',
          tickerSymbol: '8270',
          name: 'Buruj',
          name_ar: 'Ø¨Ø±Ù\210Ø¬',
          sector: 'Insurance',
          website: 'www.buruj.com.sa'
      },
      {
          symbol: '8280:TADAWUL',
          tickerSymbol: '8280',
          name: 'Liva',
          name_ar: 'Ù\204Ù\212Ù\201Ø§',
          sector: 'Insurance',
          website: 'www.liva.com.sa'
      },
      {
          symbol: '8300:TADAWUL',
          tickerSymbol: '8300',
          name: 'Wataniya',
          name_ar: 'Ø§Ù\204Ù\210Ø•Ù\206Ù\212Ø©',
          sector: 'Insurance',
          website: 'www.wataniya.com.sa'
      },
      {
          symbol: '8310:TADAWUL',
          tickerSymbol: '8310',
          name: 'Amana Insurance',
          name_ar: 'Ø£Ù\205Ø§Ù\206Ø© Ù\204Ù\204ØªØ£Ù\205Ù\212Ù\206',
          sector: 'Insurance',
          website: 'www.amana.com.sa'
      },
      {
          symbol: '8311:TADAWUL',
          tickerSymbol: '8311',
          name: 'Enaya',
          name_ar: 'Ø¹Ù\206Ø§Ù\212Ø©',
          sector: 'Insurance',
          website: 'www.enaya.com.sa'
      },
      {
          symbol: '7010:TADAWUL',
          tickerSymbol: '7010',
          name: 'STC',
          name_ar: 'Ø¥Ø³ ØªÙ\212 Ø³Ù\212',
          sector: 'Telecommunication Services',
          website: 'www.stc.com.sa'
      },
      {
          symbol: '7020:TADAWUL',
          tickerSymbol: '7020',
          name: 'Etihad Etisalat',
          name_ar: 'Ø§ØªØ–Ø§Ø¯ Ø§ØªØµØ§Ù\204Ø§Øª',
          sector: 'Telecommunication Services',
          website: 'www.mobily.com.sa'
      },
      {
          symbol: '7030:TADAWUL',
          tickerSymbol: '7030',
          name: 'Zain KSA',
          name_ar: 'Ø²Ù\212Ù\206 Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212Ø©',
          sector: 'Telecommunication Services',
          website: 'www.sa.zain.com'
      },
      {
          symbol: '7040:TADAWUL',
          tickerSymbol: '7040',
          name: 'Atheeb Telecom',
          name_ar: 'Ø§ØªØµØ§Ù\204Ø§Øª Ø¹Ø°Ù\212Ø¨',
          sector: 'Telecommunication Services',
          website: 'www.atheeb.com.sa'
      },
      {
          symbol: '2080:TADAWUL',
          tickerSymbol: '2080',
          name: 'GASCO',
          name_ar: 'ØºØ§Ø²Ù\203Ù\210',
          sector: 'Utilities',
          website: 'www.gasco.com.sa'
      },
      {
          symbol: '5110:TADAWUL',
          tickerSymbol: '5110',
          name: 'Saudi Electricity',
          name_ar: 'Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212Ø© Ù\204Ù\204Ù\203Ù\207Ø±Ø¨Ø§Ø¡',
          sector: 'Utilities',
          website: 'www.se.com.sa'
      },
      {
          symbol: '2084:TADAWUL',
          tickerSymbol: '2084',
          name: 'Miahona',
          name_ar: 'Ù\205Ù\212Ø§Ù\207Ù\206Ø§',
          sector: 'Utilities',
          website: 'www.miahona.com.sa'
      },
      {
          symbol: '2081:TADAWUL',
          tickerSymbol: '2081',
          name: 'AWPT',
          name_ar: 'Ø£Ù\212 Ø¯Ø¨Ù\204Ù\212Ù\210 Ø¨Ù\212 ØªÙ\212',
          sector: 'Utilities',
          website: 'www.awpt.com.sa'
      },
      {
          symbol: '2082:TADAWUL',
          tickerSymbol: '2082',
          name: 'ACWA Power',
          name_ar: 'Ø£Ù\203Ù\210Ø§ Ø¨Ø§Ù\210Ø±',
          sector: 'Utilities',
          website: 'www.acwapower.com'
      },
      {
          symbol: '2083:TADAWUL',
          tickerSymbol: '2083',
          name: 'Marafiq',
          name_ar: 'Ù\205Ø±Ø§Ù\201Ù\202',
          sector: 'Utilities',
          website: 'www.marafiq.com.sa'
      },
      {
          symbol: '4330:TADAWUL',
          tickerSymbol: '4330',
          name: 'Riyad REIT',
          name_ar: 'Ø§Ù\204Ø±Ù\212Ø§Ø¶ Ø±Ù\212Øª',
          sector: 'REITs',
          website: 'www.riyadreit.com'
      },
      {
          symbol: '4331:TADAWUL',
          tickerSymbol: '4331',
          name: 'Aljazira REIT',
          name_ar: 'Ø§Ù\204Ø¬Ø²Ù\212Ø±Ø© Ø±Ù\212Øª',
          sector: 'REITs',
          website: 'www.aljazirareit.com'
      },
      {
          symbol: '4332:TADAWUL',
          tickerSymbol: '4332',
          name: 'Jadwa REIT Alharamain',
          name_ar: 'Ø¬Ø¯Ù\210Ù\211 Ø±Ù\212Øª Ø§Ù\204ØØ±Ù\205Ù\212Ù\206',
          sector: 'REITs',
          website: 'www.jadwareit.com'
      },
      {
          symbol: '4333:TADAWUL',
          tickerSymbol: '4333',
          name: 'Taleem REIT',
          name_ar: 'ØªØ¹Ù\204Ù\212Ù\205 Ø±Ù\212Øª',
          sector: 'REITs',
          website: 'www.taleemreit.com'
      },
      {
          symbol: '4334:TADAWUL',
          tickerSymbol: '4334',
          name: 'Al Maather REIT',
          name_ar: 'Ø§Ù\204Ù\205Ø¹Ø°Ø± Ø±Ù\212Øª',
          sector: 'REITs',
          website: 'www.almaatherreit.com'
      },
      {
          symbol: '4335:TADAWUL',
          tickerSymbol: '4335',
          name: 'Musharaka REIT',
          name_ar: 'Ù\205Ø´Ø§Ø±Ù\203Ø© Ø±Ù\212Øª',
          sector: 'REITs',
          website: 'www.musharakareit.com'
      },
      {
          symbol: '4336:TADAWUL',
          tickerSymbol: ’4336’,
          name: ’Mulkia REIT’,
          name_ar: ’Ù\205Ù\204Ù\203Ù\212Ø© Ø±Ù\212Øª’,
          sector: ’REITs’,
          website: ’www.mulkiareit.com’
      },
      {
          symbol: ’4337:TADAWUL’,
          tickerSymbol: ’4337’,
          name: ’SICO Saudi REIT’,
          name_ar: ’Ø³Ù\212Ù\203Ù\210 Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212Ø© Ø±Ù\212Øª’,
          sector: ’REITs’,
          website: ’www.sicoreit.com’
      },
      {
          symbol: ’4338:TADAWUL’,
          tickerSymbol: ’4338’,
          name: ’Alahli REIT 1’,
          name_ar: ’Ø§Ù\204Ø£Ù\207Ù\204Ù\212 Ø±Ù\212Øª 1’,
          sector: ’REITs’,
          website: ’www.alahlireit.com’
      },
      {
          symbol: ’4344:TADAWUL’,
          tickerSymbol: ’4344’,
          name: ’Sedco Capital REIT’,
          name_ar: ’Ø³Ø¯Ù\203Ù\210 Ù\203Ø§Ø¨Ù\212ØªØ§Ù\204 Ø±Ù\212Øª’,
          sector: ’REITs’,
          website: ’www.sedcoreit.com’
      },
      {
          symbol: ’4339:TADAWUL’,
          tickerSymbol: ’4339’,
          name: ’Derayah REIT’,
          name_ar: ’Ø¯Ø±Ø§Ù\212Ø© Ø±Ù\212Øª’,
          sector: ’REITs’,
          website: ’www.derayahreit.com’
      },
      {
          symbol: ’4340:TADAWUL’,
          tickerSymbol: ’4340’,
          name: ’Al Rajhi REIT’,
          name_ar: ’Ø§Ù\204Ø±Ø§Ø¬Ø-Ù\212 Ø±Ù\212Øª’,
          sector: ’REITs’,
          website: ’www.alrajhireit.com’
      },
      {
          symbol: ’4345:TADAWUL’,
          tickerSymbol: ’4345’,
          name: ’Alinma Retail REIT’,
          name_ar: ’Ø§Ù\204Ø¥Ù\206Ù\205Ø§Ø¡ Ø±Ù\212Øª Ù\204Ù\204ØªØ¬Ø²Ø¦Ø©’,
          sector: ’REITs’,
          website: ’www.alinmareit.com’
      },
      {
          symbol: ’4342:TADAWUL’,
          tickerSymbol: ’4342’,
          name: ’Jadwa REIT Saudi’,
          name_ar: ’Ø¬Ø¯Ù\210Ù\211 Ø±Ù\212Øª Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212Ø©’,
          sector: ’REITs’,
          website: ’www.jadwareit.com’
      },
      {
          symbol: ’4346:TADAWUL’,
          tickerSymbol: ’4346’,
          name: ’MEFIC REIT’,
          name_ar: ’Ù\205Ù\212Ù\201Ù\203 Ø±Ù\212Øª’,
          sector: ’REITs’,
          website: ’www.meficreit.com’
      },
      {
          symbol: ’4347:TADAWUL’,
          tickerSymbol: ’4347’,
          name: ’Bonyan REIT’,
          name_ar: ’Ø¨Ù\206Ù\212Ø§Ù\206 Ø±Ù\212Øª’,
          sector: ’REITs’,
          website: ’www.bonyanreit.com’
      },
      {
          symbol: ’4348:TADAWUL’,
          tickerSymbol: ’4348’,
          name: ’Alkhabeer REIT’,
          name_ar: ’Ø§Ù\204Ø®Ø¨Ù\212Ø± Ø±Ù\212Øª’,
          sector: ’REITs’,
          website: ’www.alkhabeerreit.com’
      },
      {
          symbol: ’4349:TADAWUL’,
          tickerSymbol: ’4349’,
          name: ’Alinma Hospitality REIT’,
          name_ar: ’Ø§Ù\204Ø¥Ù\206Ù\205Ø§Ø¡ Ø±Ù\212Øª Ù\204Ù\204Ø¶Ù\212Ø§Ù\201Ø©’,
          sector: ’REITs’,
          website: ’www.alinmareit.com’
      },
      {
          symbol: ’4350:TADAWUL’,
          tickerSymbol: ’4350’,
          name: ’Alistithmar REIT’,
          name_ar: ’Ø§Ù\204Ø§Ø³ØªØ«Ù\205Ø§Ø± Ø±Ù\212Øª’,
          sector: ’REITs’,
          website: ’www.alistithmarreit.com’
      },
      {
          symbol: ’4020:TADAWUL’,
          tickerSymbol: ’4020’,
          name: ’Alakaria’,
          name_ar: ’Ø§Ù\204Ø¹Ù\202Ø§Ø±Ù\212Ø©’,
          sector: ’Real Estate Mgmt & Dev\’t’,
          website: ’www.alakaria.com.sa’
      },
      {
          symbol: ’4090:TADAWUL’,
          tickerSymbol: ’4090’,
          name: ’Taiba’,
          name_ar: ’Ø•Ù\212Ø¨Ø©’,
          sector: ’Real Estate Mgmt & Dev\’t’,
          website: ’www.taiba.com.sa’
      },
      {
          symbol: ’4100:TADAWUL’,
          tickerSymbol: ’4100’,
          name: ’MCDC’,
          name_ar: ’Ù\205Ù\203Ø©’,
          sector: ’Real Estate Mgmt & Dev\’t’,
          website: ’www.mcdc.com.sa’
      },
      {
          symbol: ’4150:TADAWUL’,
          tickerSymbol: ’4150’,
          name: ’ARDCO’,
          name_ar: ’Ø§Ù\204Ø£Ø±Ø¶’,
          sector: ’Real Estate Mgmt & Dev\’t’,
          website: ’www.ardco.com.sa’
      },
      {
          symbol: ’4220:TADAWUL’,
          tickerSymbol: ’4220’,
          name: ’Emaar EC’,
          name_ar: ’Ø¥Ø¹Ù\205Ø§Ø± Ø§Ù\204Ù\205Ø¯Ù\212Ù\206Ø© Ø§Ù\204Ø§Ù\202ØªØµØ§Ø¯Ù\212Ø©’,
          sector: ’Real Estate Mgmt & Dev\’t’,
          website: ’www.emaar.com.sa’
      },
      {
          symbol: ’4250:TADAWUL’,
          tickerSymbol: ’4250’,
          name: ’Jabal Omar’,
          name_ar: ’Ø¬Ø¨Ù\204 Ø¹Ù\205Ø±’,
          sector: ’Real Estate Mgmt & Dev\’t’,
          website: ’www.jabalomar.com.sa’
      },
      {
          symbol: ’4300:TADAWUL’,
          tickerSymbol: ’4300’,
          name: ’Dar Alarkan’,
          name_ar: ’Ø¯Ø§Ø± Ø§Ù\204Ø£Ø±Ù\203Ø§Ù\206’,
          sector: ’Real Estate Mgmt & Dev\’t’,
          website: ’www.daralarkan.com’
      },
      {
          symbol: ’4310:TADAWUL’,
          tickerSymbol: ’4310’,
          name: ’KEC’,
          name_ar: ’Ù\203Ù\212Ù\203’,
          sector: ’Real Estate Mgmt & Dev\’t’,
          website: ’www.kec.com.sa’
      },
      {
          symbol: ’4320:TADAWUL’,
          tickerSymbol: ’4320’,
          name: ’Alandalus’,
          name_ar: ’Ø§Ù\204Ø£Ù\206Ø¯Ù\204Ø³’,
          sector: ’Real Estate Mgmt & Dev\’t’,
          website: ’www.alandalus.com.sa’
      },
      {
          symbol: ’4321:TADAWUL’,
          tickerSymbol: ’4321’,
          name: ’Cenomi Centers’,
          name_ar: ’Ø³Ù\212Ù\206Ù\210Ù\205Ù\212 Ø³Ù\206ØªØ±Ø²’,
          sector: ’Real Estate Mgmt & Dev\’t’,
          website: ’www.cenomicenters.com’
      },
      {
          symbol: ’4322:TADAWUL’,
          tickerSymbol: ’4322’,
          name: ’Retal’,
          name_ar: ’Ø±ØªØ§Ù\204’,
          sector: ’Real Estate Mgmt & Dev\’t’,
          website: ’www.retal.com.sa’
      },
      {
          symbol: ’4323:TADAWUL’,
          tickerSymbol: ’4323’,
          name: ’Sumou’,
          name_ar: ’Ø³Ù\205Ù\210’,
          sector: ’Real Estate Mgmt & Dev\’t’,
          website: ’www.sumou.com.sa’
      },
      {
          symbol: ’4230:TADAWUL’,
          tickerSymbol: ’4230’,
          name: ’Red Sea’,
          name_ar: ’Ø§Ù\204Ø¨Ø-Ø± Ø§Ù\204Ø£Ø-Ù\205Ø±’,
          sector: ’Real Estate Mgmt & Dev\’t’,
          website: ’www.redsea.com.sa’
      },
      {
          symbol: ’7200:TADAWUL’,
          tickerSymbol: ’7200’,
          name: ’MIS’,
          name_ar: ’Ø¥Ù\205 Ø¢Ù\212 Ø¥Ø³’,
          sector: ’Software & Services’,
          website: ’www.mis.com.sa’
      },
      {
          symbol: ’7201:TADAWUL’,
          tickerSymbol: ’7201’,
          name: ’Arab Sea’,
          name_ar: ’Ø§Ù\204Ø¨Ø-Ø± Ø§Ù\204Ø¹Ø±Ø¨Ù\212’,
          sector: ’Software & Services’,
          website: ’www.arabsea.com.sa’
      },
      {
          symbol: ’7202:TADAWUL’,
          tickerSymbol: ’7202’,
          name: ’Solutions’,
          name_ar: ’Ø-Ù\204Ù\210Ù\204’,
          sector: ’Software & Services’,
          website: ’www.solutions.com.sa’
      },
      {
          symbol: ’7203:TADAWUL’,
          tickerSymbol: ’7203’,
          name: ’ELM’,
          name_ar: ’Ø¹Ù\204Ù\205’,
          sector: ’Software & Services’,
          website: ’www.elm.com.sa’
      },
      {
          symbol: ’7204:TADAWUL’,
          tickerSymbol: ’7204’,
          name: ’2P’,
          name_ar: ’ØªÙ\210 Ø¨Ù\212’,
          sector: ’Software & Services’,
          website: ’www.2p.com.sa’
      },
      {
          symbol: ’4165:TADAWUL’,
          tickerSymbol: ’4165’,
          name: ’Almajed Oud’,
          name_ar: ’Ø§Ù\204Ù\205Ø¬Ø¯ Ù\204Ù\204Ø¹Ù\210Ø¯’,
          sector: ’Household & Personal Products’,
          website: ’www.almajedoud.com.sa’
      },
      {
          symbol: ’9406:TADAWUL’,
          tickerSymbol: ’9406’,
          name: ’Albilad MSCI’,
          name_ar: 'Ø§Ù\204Ø¨Ù\204Ø§Ø¯ Ø¥Ù\205 Ø¥Ø³ Ø³Ù\212 Ø¢Ù\212',
          sector: 'Index Funds',
          website: 'www.albiladmsci.com'
      },
      {
          symbol: '9402:TADAWUL',
          tickerSymbol: '9402',
          name: 'Alawwal MT30',
          name_ar: 'Ø§Ù\204Ø£Ù\210Ù\204 Ø¥Ù\205 ØªÙ\212 30',
          sector: 'Index Funds',
          website: 'www.alawwalmt30.com'
      },
      {
          symbol: '9403:TADAWUL',
          tickerSymbol: '9403',
          name: 'Albilad Sukuk',
          name_ar: 'Ø§Ù\204Ø¨Ù\204Ø§Ø¯ ØµÙ\203Ù\210Ù\203',
          sector: 'Index Funds',
          website: 'www.albiladsukuk.com'
      },
      {
          symbol: '4701:TADAWUL',
          tickerSymbol: '4701',
          name: 'ADITF',
          name_ar: 'Ø£Ø¯Ù\212ØªÙ\201',
          sector: 'Index Funds',
          website: 'www.aditf.com.sa'
      },
      {
          symbol: '9400:TADAWUL',
          tickerSymbol: '9400',
          name: 'Yaqeen 30',
          name_ar: 'Ù\212Ù\202Ù\212Ù\206 30',
          sector: 'Index Funds',
          website: 'www.yaqeen30.com'
      },
      {
          symbol: '9401:TADAWUL',
          tickerSymbol: '9401',
          name: 'Yaqeen Petrochemical',
          name_ar: 'Ù\212Ù\202Ù\212Ù\206 Ù\204Ù\204Ø¨ØªØ±Ù\210Ù\203Ù\212Ù\205Ø§Ù\210Ù\212Ø§Øª',
          sector: 'Index Funds',
          website: 'www.yaqeenpetrochemical.com'
      },
      {
          symbol: '9404:TADAWUL',
          tickerSymbol: '9404',
          name: 'Alinma Sukuk',
          name_ar: 'Ø§Ù\204Ø¥Ù\206Ù\205Ø§Ø¡ ØµÙ\203Ù\210Ù\203',
          sector: 'Index Funds',
          website: 'www.alinmasukuk.com'
      },
      {
          symbol: '9405:TADAWUL',
          tickerSymbol: '9405',
          name: 'Albilad Gold',
          name_ar: 'Ø§Ù\204Ø¨Ù\204Ø§Ø¯ Ø°Ù\207Ø¨',
          sector: 'Index Funds',
          website: 'www.albiladgold.com'
      },
      {
          symbol: '4700:TADAWUL',
          tickerSymbol: '4700',
          name: 'Alkhabeer Income',
          name_ar: 'Ø§Ù\204Ø®Ø¨Ù\212Ø± Ø¯Ø®Ù\204',
          sector: 'Index Funds',
          website: 'www.alkhabeerincome.com'
      },
      {
          symbol: '9407:TADAWUL',
          tickerSymbol: '9407',
          name: 'Albilad US Tech',
          name_ar: 'Ø§Ù\204Ø¨Ù\204Ø§Ø¯ Ø§Ù\204ØªÙ\203Ù\206Ù\210Ù\204Ù\210Ø¬Ù\212Ø§ Ø§Ù\204Ø£Ù\205Ø±Ù\212Ù\203Ù
\212Ø©',
          sector: 'Index Funds',
          website: 'www.albiladustech.com'
      },
      {
          symbol: '9408:TADAWUL',
          tickerSymbol: '9408',
          name: 'Albilad Saudi Growth',
          name_ar: 'Ø§Ù\204Ø¨Ù\204Ø§Ø¯ Ø§Ù\204Ù\206Ù\205Ù\210 Ø§Ù\204Ø³Ø¹Ù\210Ø¯Ù\212',
          sector: 'Index Funds',
          website: 'www.albiladsaudigrowth.com'
      },
      {
          symbol: ’4702:TADAWUL’,
          tickerSymbol: ’4702’,
          name: ’Alkhabeer Income 2030’,
          name_ar: ’Ø§Ù\204Ø®Ø¨Ù\212Ø± Ø¯Ø®Ù\204 2030’,
          sector: ’Index Funds’,
          website: ’www.alkhabeerincome2030.com’
      },
      {
          symbol: ’4703:TADAWUL’,
          tickerSymbol: ’4703’,
          name: ’Sedco Multi Asset’,
          name_ar: ’Ø³Ø¯Ù\203Ù\210 Ù\205ØªØ¹Ø¯Ø¯ Ø§Ù\204Ø£ØµÙ\210Ù\204’,
          sector: ’Index Funds’,
          website: ’www.sedcomultiasset.com’
      },
      {
          symbol: ’9410:TADAWUL’,
          tickerSymbol: ’9410’,
          name: ’Albilad Hong Kong China’,
          name_ar: ’Ø§Ù\204Ø¨Ù\204Ø§Ø¯ Ù\207Ù\210Ù\206Øº Ù\203Ù\210Ù\206Øº ØµÙ\212Ù\206’,
          sector: ’Index Funds’,
          website: ’www.albiladhongkongchina.com’
      },
      {
          symbol: ’9411:TADAWUL’,
          tickerSymbol: ’9411’,
          name: ’SAB HK’,
          name_ar: ’Ø³Ø§Ø¨ Ù\207Ù\210Ù\206Øº Ù\203Ù\210Ù\206Øº’,
          sector: ’Index Funds’,
          website: ’www.sabhk.com’
      }
  ]
  // Helper functions
export function findSaudiSymbol(query: string) {
  const normalizedQuery = query.toLowerCase();

  return SAUDI_SYMBOLS.find(stock =>
    stock.symbol.toLowerCase().includes(normalizedQuery) ||
    stock.tickerSymbol.toLowerCase().includes(normalizedQuery) ||
    stock.name.toLowerCase().includes(normalizedQuery) ||
    stock.name_ar.includes(query)
  );
}

export function getSectorCompanies(sector: string) {
  return SAUDI_SYMBOLS.filter(stock =>
    stock.sector.toLowerCase() === sector.toLowerCase()
  );
}

export function formatSaudiSymbol(symbol: string | number) {
  const stock = SAUDI_SYMBOLS.find(s =>
    s.tickerSymbol === symbol.toString() ||
    s.symbol === symbol.toString()
  );

  if (!stock) return ‘${symbol}:TADAWUL‘;
  return stock.symbol;
}

export function searchStocks(query: string) {
  const normalizedQuery = query.toLowerCase();
  return SAUDI_SYMBOLS.filter(stock =>
    stock.symbol.toLowerCase().includes(normalizedQuery) ||
    stock.tickerSymbol.toLowerCase().includes(normalizedQuery) ||
    stock.name.toLowerCase().includes(normalizedQuery) ||
    stock.name_ar.includes(query) ||
    stock.sector.toLowerCase().includes(normalizedQuery) ||
    stock.website.toLowerCase().includes(normalizedQuery)
  );
}


export function getAllSymbols() {
  return SAUDI_SYMBOLS.map(stock => stock.symbol);
}
```

```
// // hooks/useMarketSocket.ts

// import { useEffect, useRef, useState } from ’react’;
// import { WS_URL, API_KEY, MarketQuote } from ’@/app/config/market’;

// export function useMarketSocket(symbols: string[]) {
//   const [quotes, setQuotes] = useState<Record<string, MarketQuote>>({});
//   const ws = useRef<WebSocket | null>(null);

//   useEffect(() => {
//     if (!symbols.length) return;

//     // Create WebSocket connection
//     ws.current = new WebSocket(‘${WS_URL}?apikey=${API_KEY}‘);

//     ws.current.onopen = () => {
//       console.log(’WebSocket Connected’);
//       if (ws.current?.readyState === WebSocket.OPEN) {
//         // Subscribe to symbols
//         ws.current.send(JSON.stringify({
//           action: ’subscribe’,
//           params: {
//             symbols: symbols.join(’,’)
//           }
//         }));
//       }
//     };

//     ws.current.onmessage = (event) => {
//       const data = JSON.parse(event.data);
//       if (data.event === ’price’) {
//         setQuotes(prev => ({
//           ...prev,
//           [data.symbol]: {
//             ...prev[data.symbol],
//             symbol: data.symbol,
//             name: prev[data.symbol]?.name || data.symbol,
//             price: parseFloat(data.price),
//             timestamp: data.timestamp,
//             change: parseFloat(data.change) || 0,
//             percentChange: parseFloat(data.percent_change) || 0,
//             volume: parseInt(data.volume) || 0
//           }
//         }));
//       }
//     };

//     ws.current.onerror = (error) => {
//       console.error(’WebSocket error:’, error);
//     };

//     return () => {
//       if (ws.current) {
//         ws.current.close();
//       }
//     };
//   }, [symbols]);

//   return quotes;
// }
// // hooks/useSaudiMarket.ts
// import { useState, useEffect } from ’react’;

// interface SaudiStock {
//   symbol: string;
//   name: string;
//   currency: string;
//   exchange: string;
//   mic_code: string;
//   country: string;
//   type: string;
// }

// interface StockQuote {
//   symbol: string;
//   price: string;
//   change: string;
//   percent_change: string;
//   volume: string;
//   timestamp: string;
// }

// interface RealTimeData {
//   price: number;
//   change: number;
//   percentChange: number;
//   volume: number;
//   timestamp: string;
// }

// interface StockData {
//   stock: SaudiStock;
//   quote?: StockQuote;
//   realTimeData?: RealTimeData;
// }

// interface WebSocketMessage {
//   event: string;
//   symbol: string;
//   price: string;
//   change: string;
//   percent_change: string;
//   volume: string;
//   timestamp: string;
// }

// export function useSaudiMarket() {
//   const [stocks, setStocks] = useState<StockData[]>([]);
//   const [loading, setLoading] = useState(true);
//   const [error, setError] = useState<string | null>(null);
//   const [selectedTimeframe, setSelectedTimeframe] = useState(’1day’);

//   const fetchSaudiMarketData = async () => {
//     try {
//       setLoading(true);
//       // Fetch Saudi stocks list
//       const stocksResponse = await fetch(
//         `https://api.twelvedata.com/stocks?country=Saudi Arabia&apikey=${process.env.NEXT_PUBLIC_TWELVE_DATA
_API_KEY}`
//       );

//       if (!stocksResponse.ok) {
//         throw new Error(’Failed to fetch stocks list’);
//       }

//       const stocksData = await stocksResponse.json();
//       const stocksList: SaudiStock[] = stocksData.data || stocksData;

//       // Take first 20 stocks to avoid rate limits
//       const stocksToFetch = stocksList.slice(0, 20);
//       const symbols = stocksToFetch.map(stock => stock.symbol).join(’,’);

//       // Fetch quotes for all stocks at once
//       const quotesResponse = await fetch(
//         `https://api.twelvedata.com/quote?symbol=${symbols}&apikey=${process.env.NEXT_PUBLIC_TWELVE_DATA_API
_KEY}`
//       );

//       if (!quotesResponse.ok) {
//         throw new Error(’Failed to fetch quotes’);
//       }

//       const quotesData = await quotesResponse.json();
//       let quotesArray: StockQuote[];

//       if (Array.isArray(quotesData)) {
```

```
//         quotesArray = quotesData;
//       } else if (quotesData.data) {
//         quotesArray = Array.isArray(quotesData.data) ? quotesData.data : [quotesData.data];
//       } else {
//         quotesArray = [quotesData];
//       }

//       // Combine stocks with their quotes
//       const combinedData = stocksToFetch.map(stock => {
//         const quote = quotesArray.find(q => q.symbol === stock.symbol);
//         return {
//           stock,
//           quote,
//           realTimeData: quote ? {
//             price: parseFloat(quote.price),
//             change: parseFloat(quote.change),
//             percentChange: parseFloat(quote.percent_change),
//             volume: parseInt(quote.volume),
//             timestamp: quote.timestamp
//           } : undefined
//         };
//       });

//       setStocks(combinedData);
//       setupWebSocket(symbols.split(’,’));
//       setError(null);
//     } catch (err) {
//       console.error(’Error fetching market data:’, err);
//       setError(err instanceof Error ? err.message : ’An error occurred’);
//     } finally {
//       setLoading(false);
//     }
//   };

//   const setupWebSocket = (symbols: string[]) => {
//     const ws = new WebSocket(
//       ‘wss://ws.twelvedata.com/v1/quotes/price?apikey=${process.env.NEXT_PUBLIC_TWELVE_DATA_API_KEY}‘
//     );

//     ws.onopen = () => {
//       ws.send(JSON.stringify({
//         action: ’subscribe’,
//         params: {
//           symbols: symbols.join(’,’)
//         }
//       }));
//     };

//     ws.onmessage = (event: MessageEvent) => {
//       try {
//         const data = JSON.parse(event.data) as WebSocketMessage;
//         if (data.event === ’price’) {
//           setStocks(prev => prev.map(stock => {
//             if (stock.stock.symbol === data.symbol) {
//               return {
//                 ...stock,
//                 realTimeData: {
//                   price: parseFloat(data.price),
//                   change: parseFloat(data.change),
//                   percentChange: parseFloat(data.percent_change),
//                   volume: parseInt(data.volume),
//                   timestamp: data.timestamp
//                 }
//               };
//             }
//             return stock;
//           }));
//         }
//       } catch (error) {
//         console.error(’WebSocket message parsing error:’, error);
//       }
//     };

//     ws.onerror = (error: Event) => {
//       console.error(’WebSocket error:’, error);
//     };

//     return () => {
//       if (ws.readyState === WebSocket.OPEN) {
//         ws.close();
//       }
//     };
//   };

//   useEffect(() => {
//     fetchSaudiMarketData();
//     // Refresh data every minute
//     const interval = setInterval(fetchSaudiMarketData, 60000);
//     return () => clearInterval(interval);
//   }, [selectedTimeframe]);

//   const changeTimeframe = (timeframe: string) => {
//     setSelectedTimeframe(timeframe);
//   };

//   return {
//     stocks,
//     loading,
//     error,
//     selectedTimeframe,
//     changeTimeframe,
//     refreshData: fetchSaudiMarketData
//   };
// }
// /app/lib/context/financeContextUtils.ts

import { FinancialMetadata } from './types';

export class FinanceContextAnalyzer {
  private static readonly FINANCIAL_INSTRUMENTS = [
    'stock', 'bond', 'etf', 'option', 'future', 'forex', 'crypto',
    'dividend', 'mutual fund', 'index', 'commodity'
  ];

  private static readonly TECHNICAL_INDICATORS = [
    'moving average', 'MACD', 'RSI', 'bollinger', 'volume', 'momentum',
    'support', 'resistance', 'trend', 'volatility'
  ];

  private static readonly FUNDAMENTAL_FACTORS = [
    'earnings', 'revenue', 'profit', 'margin', 'PE ratio', 'market cap',
    'dividend yield', 'cash flow', 'debt', 'assets'
  ];

  private static readonly MARKET_KEYWORDS = [
    'bull', 'bear', 'volatile', 'correction', 'crash', 'rally',
    'uptick', 'downturn', 'market', 'trade', 'buy', 'sell'
  ];

  public static analyzeFinancialMessage(message: string): FinancialMetadata {
    const words = message.toLowerCase().split(/\s+/);

    return {
      instruments: this.extractFinancialInstruments(message),
      companies: this.extractCompanies(message),
      markets: this.extractMarkets(message),
      metrics: this.extractMetrics(message),
      timeframes: this.extractTimeframes(message),
      sentiment: this.analyzeMarketSentiment(message),
      confidenceScore: this.calculateConfidenceScore(message),
      riskLevel: this.assessRiskLevel(message),
      technicalIndicators: this.extractTechnicalIndicators(message),
      fundamentalFactors: this.extractFundamentalFactors(message)
    };
  }

  private static extractFinancialInstruments(message: string): string[] {
    const text = message.toLowerCase();
    return this.FINANCIAL_INSTRUMENTS.filter(instrument =>
      text.includes(instrument)
    );
  }

  private static extractCompanies(message: string): string[] {
    // Look for potential company names (capitalized words)
    const companies = new Set<string>();
    const words = message.split(/\s+/);

    for (let i = 0; i < words.length; i++) {
      const word = words[i];
      if (word.length > 1 &&
          word[0] === word[0].toUpperCase() &&
          word[1] === word[1].toLowerCase()) {
        companies.add(word);
      }
      // Check for stock symbols (uppercase letters)
      if (word.match(/^\$?[A-Z]{1,5}$/)) {
        companies.add(word);
      }
    }

    return Array.from(companies);
  }

  private static extractMarkets(message: string): string[] {
    const commonMarkets = [
      'NYSE', 'NASDAQ', 'DOW', 'S&P', 'FTSE', 'market',
      'exchange', 'trading'
    ];

    return commonMarkets.filter(market =>
      message.toLowerCase().includes(market.toLowerCase())
    );
  }

  private static extractMetrics(message: string): string[] {
    const commonMetrics = [
      'price', 'volume', 'volatility', 'return', 'yield',
      'ratio', 'growth', 'loss', 'gain', 'percentage'
    ];
    return commonMetrics.filter(metric =>
      message.toLowerCase().includes(metric)
    );
  }

  private static extractTimeframes(message: string): string[] {
    const timeframes = [
      ’day’, ’week’, ’month’, ’year’, ’quarter’,
      ’short-term’, ’long-term’, ’intraday’
    ];

    return timeframes.filter(timeframe =>
      message.toLowerCase().includes(timeframe)
    );
  }

  private static analyzeMarketSentiment(message: string): ’bullish’ | ’bearish’ | ’neutral’ {
    const text = message.toLowerCase();

    const bullishWords = [’buy’, ’bull’, ’up’, ’growth’, ’positive’, ’increase’];
    const bearishWords = [’sell’, ’bear’, ’down’, ’decline’, ’negative’, ’decrease’];

    let sentiment = 0;

    bullishWords.forEach(word => {
      if (text.includes(word)) sentiment++;
    });

    bearishWords.forEach(word => {
      if (text.includes(word)) sentiment--;
    });

    if (sentiment > 0) return ’bullish’;
    if (sentiment < 0) return ’bearish’;
    return ’neutral’;
  }

  private static calculateConfidenceScore(message: string): number {
    let score = 0.5; // Base confidence

    // Increase confidence based on presence of specific data
    if (this.extractMetrics(message).length > 0) score += 0.1;
    if (this.extractTechnicalIndicators(message).length > 0) score += 0.1;
    if (this.extractFundamentalFactors(message).length > 0) score += 0.1;

    // Cap the score at 1.0
    return Math.min(score, 1.0);
  }

  private static assessRiskLevel(message: string): ’low’ | ’medium’ | ’high’ {
    const text = message.toLowerCase();

    const highRiskWords = [’volatile’, ’risky’, ’speculation’, ’aggressive’];
    const lowRiskWords = [’safe’, ’conservative’, ’stable’, ’defensive’];

    let riskScore = 0;

    highRiskWords.forEach(word => {
      if (text.includes(word)) riskScore++;
    });

    lowRiskWords.forEach(word => {
      if (text.includes(word)) riskScore--;
    });

    if (riskScore > 0) return ’high’;
    if (riskScore < 0) return ’low’;
    return ’medium’;
  }

  private static extractTechnicalIndicators(message: string): string[] {
    return this.TECHNICAL_INDICATORS.filter(indicator =>
      message.toLowerCase().includes(indicator)
    );
  }

  private static extractFundamentalFactors(message: string): string[] {
    return this.FUNDAMENTAL_FACTORS.filter(factor =>
      message.toLowerCase().includes(factor)
    );
  }
}
```

```
// /app/lib/context/types.ts

export interface FinancialMetadata {
    instruments?: string[];              // Mentioned financial instruments
    companies?: string[];               // Mentioned companies
    markets?: string[];                 // Mentioned markets/exchanges
    metrics?: string[];                 // Financial metrics discussed
    timeframes?: string[];             // Timeframes mentioned
    sentiment?: 'bullish' | 'bearish' | 'neutral';
    confidenceScore?: number;          // Confidence in financial analysis
    riskLevel?: 'low' | 'medium' | 'high';
    technicalIndicators?: string[];    // Technical analysis indicators
    fundamentalFactors?: string[];     // Fundamental analysis factors
  }

  export interface Message {
    id: string;
    role: 'user' | 'assistant';
    content: string;
    timestamp: number;
    metadata: FinancialMetadata;
  }

  export interface MarketContext {
    currentMarketHours: boolean;
    marketConditions: 'bull' | 'bear' | 'neutral';
    volatilityLevel: 'low' | 'medium' | 'high';
    majorIndices: {
      [key: string]: {
        value: number;
        change: number;
        changePercent: number;
      };
    };
  }

  export interface ConversationContext {
    conversationId: string;
    messages: Message[];
    activeInstruments: string[];
    activeTechnicalAnalysis: boolean;
    fundamentalAnalysisActive: boolean;
    marketContext?: MarketContext;
    lastUpdated: number;
    userPreferences?: {
      preferredMarkets?: string[];
      riskTolerance?: 'conservative' | 'moderate' | 'aggressive';
      preferredTimeframe?: 'short' | 'medium' | 'long';
      technicalAnalysisLevel?: 'basic' | 'intermediate' | 'advanced';
    };
  }

  export interface ContextWindow {
    maxSize: number;
    currentSize: number;
    priorityThreshold: number;
    recentMessages: Message[];
    marketSnapshots: MarketContext[];
    keyInsights: string[];
  }

  export type ContextPriority = 'critical' | 'high' | 'medium' | 'low';

  export interface ContextScore {
    messageId: string;
    marketRelevance: number;
    timeRelevance: number;
    instrumentRelevance: number;
    userPriorityScore: number;
    finalScore: number;
  }
```

```
// /app/lib/context/financeContextManager.ts

import {
  Message,
  ConversationContext,
  ContextWindow,
  ContextScore,
  ContextPriority,
  MarketContext,
  FinancialMetadata
} from ’./types’;

export class FinanceContextManager {
  private readonly maxWindowSize: number;
  private contexts: Map<string, ConversationContext>;
  private activeWindow: ContextWindow;
  private marketContext: MarketContext;

  constructor(maxWindowSize: number = 4096) {
    this.maxWindowSize = maxWindowSize;
    this.contexts = new Map();
    this.activeWindow = {
      maxSize: maxWindowSize,
      currentSize: 0,
      priorityThreshold: 0.7,
      recentMessages: [],
      marketSnapshots: [],
      keyInsights: []
    };
    this.marketContext = {
      currentMarketHours: false,
      marketConditions: ’neutral’,
      volatilityLevel: ’low’,
      majorIndices: {}
    };
  }

  private calculateFinancialContextScore(message: Message, currentTime: number): ContextScore {
    const timeRelevance = Math.exp(-(currentTime - message.timestamp) / (1000 * 60 * 60));
    const metadata = message.metadata;

    let marketRelevance = 0.5;
    if (metadata) {
      if (metadata.instruments?.length) marketRelevance += 0.1;
      if (metadata.markets?.length) marketRelevance += 0.1;
      if (metadata.technicalIndicators?.length) marketRelevance += 0.1;
      if (metadata.fundamentalFactors?.length) marketRelevance += 0.1;
    }

    let instrumentRelevance = 0.5;
    if (metadata?.instruments?.length) {
      instrumentRelevance = 0.7;
      if (this.isActiveInstrument(metadata.instruments[0])) {
        instrumentRelevance = 0.9;
      }
    }

    let userPriorityScore = 0.5;
    if (metadata) {
      if (metadata.riskLevel === ’high’) userPriorityScore += 0.2;
      if (metadata.confidenceScore && metadata.confidenceScore > 0.8) userPriorityScore += 0.1;
    }

    return {
      messageId: message.id,
      marketRelevance,
      timeRelevance,
      instrumentRelevance,
      userPriorityScore,
      finalScore: (
        marketRelevance * 0.3 +
        timeRelevance * 0.2 +
        instrumentRelevance * 0.3 +
        userPriorityScore * 0.2
      )
    };
  }

  private isActiveInstrument(instrument: string): boolean {
    const contexts = Array.from(this.contexts.values());
    return contexts.some(context =>
      context.activeInstruments.includes(instrument)
    );
  }

  public addMessage(conversationId: string, message: Message): void {
    let context = this.contexts.get(conversationId);

    if (!context) {
      context = {
        conversationId,
        messages: [],
        activeInstruments: [],
        activeTechnicalAnalysis: false,
        fundamentalAnalysisActive: false,
        lastUpdated: Date.now(),
        userPreferences: {
          preferredMarkets: [],
          riskTolerance: ’moderate’,
          preferredTimeframe: ’medium’,
          technicalAnalysisLevel: ’basic’
        }
      };
      this.contexts.set(conversationId, context);
    }

    if (message.metadata?.instruments) {
      context.activeInstruments = Array.from(new Set([
        ...context.activeInstruments,
        ...message.metadata.instruments
      ]));
    }

    if (message.metadata?.technicalIndicators?.length) {
      context.activeTechnicalAnalysis = true;
    }
    if (message.metadata?.fundamentalFactors?.length) {
      context.fundamentalAnalysisActive = true;
    }

    context.messages.push(message);
    context.lastUpdated = Date.now();
    this.updateActiveWindow(context);
  }

  private updateActiveWindow(context: ConversationContext): void {
    const currentTime = Date.now();

    const scoredMessages = context.messages
      .map(msg => ({
        message: msg,
        score: this.calculateFinancialContextScore(msg, currentTime)
      }))
      .sort((a, b) => b.score.finalScore - a.score.finalScore);

    this.activeWindow.recentMessages = scoredMessages
      .filter(scored => scored.score.finalScore > this.activeWindow.priorityThreshold)
      .map(scored => scored.message)
      .slice(0, Math.floor(this.maxWindowSize / 2));

    this.activeWindow.keyInsights = this.extractFinancialInsights(
      scoredMessages.slice(0, 5).map(scored => scored.message)
    );

    if (this.shouldTakeMarketSnapshot(context)) {
      this.activeWindow.marketSnapshots.push({ ...this.marketContext });
    }
  }

  private shouldTakeMarketSnapshot(context: ConversationContext): boolean {
    const latestMessage = context.messages[context.messages.length - 1];
    return Boolean(
      latestMessage.metadata?.markets?.length ||
      (this.marketContext.volatilityLevel === ’high’ && Math.random() > 0.7)
    );
  }

  private extractFinancialInsights(messages: Message[]): string[] {
    const insights = new Set<string>();

    messages.forEach(message => {
      const metadata = message.metadata;
      if (metadata) {
        if (metadata.technicalIndicators?.length) {
          insights.add(‘Technical Analysis: ${metadata.technicalIndicators.join(’, ’)}‘);
        }

        if (metadata.fundamentalFactors?.length) {
          insights.add(‘Fundamental Factors: ${metadata.fundamentalFactors.join(’, ’)}‘);
        }

        if (metadata.sentiment) {
          insights.add(`Market Sentiment: ${metadata.sentiment}`);
        }

        if (metadata.riskLevel) {
          insights.add(`Risk Level: ${metadata.riskLevel}`);
        }
      }
    });

    return Array.from(insights);
  }

  public getContextForPrompt(conversationId: string): string {
    const context = this.contexts.get(conversationId);
    if (!context) return '';

    const relevantMessages = this.activeWindow.recentMessages
      .map(msg => {
        const metadata = msg.metadata;
        return `${msg.role}: ${msg.content}
        ${metadata ? `
        Instruments: ${metadata.instruments?.join(', ') || 'none'}
        Technical Indicators: ${metadata.technicalIndicators?.join(', ') || 'none'}
        Sentiment: ${metadata.sentiment || 'neutral'}
        Risk Level: ${metadata.riskLevel || 'medium'}` : ''}`;
      })
      .join('\n\n');

    const marketContext = this.activeWindow.marketSnapshots.length > 0
      ? `\nCurrent Market Context:
        Market Hours: ${this.marketContext.currentMarketHours ? 'Open' : 'Closed'}
        Conditions: ${this.marketContext.marketConditions}
        Volatility: ${this.marketContext.volatilityLevel}`
      : '';

    const insights = this.activeWindow.keyInsights.length > 0
      ? `\nKey Insights:\n${this.activeWindow.keyInsights.join('\n')}`
      : '';

    return `${marketContext}\n${insights}\n\nRecent Discussion:\n${relevantMessages}`;
  }

  public updateMarketContext(newContext: Partial<MarketContext>): void {
    this.marketContext = { ...this.marketContext, ...newContext };
  }

  public getActiveInstruments(conversationId: string): string[] {
    return this.contexts.get(conversationId)?.activeInstruments || [];
  }

  public getCurrentMarketContext(): MarketContext {
    return this.marketContext;
  }

  public getKeyInsights(conversationId: string): string[] {
    return this.activeWindow.keyInsights;
  }
}
```

```
// /app/lib/context/financeAnalyzer.ts

export class FinanceAnalyzer {
    // Known financial instruments and companies
    private static readonly KNOWN_STOCKS = new Set([
      'AAPL', 'MSFT', 'GOOGL', 'AMZN', 'META', 'TSLA', 'NVDA', 'JPM', 'BAC', 'WFC',
      'apple', 'microsoft', 'google', 'amazon', 'meta', 'tesla', 'nvidia', 'jpmorgan', 'bank of america'
    ]);

    // Technical analysis terms
    private static readonly TECHNICAL_INDICATORS = new Set([
      'RSI', 'MACD', 'moving average', 'bollinger bands', 'support', 'resistance',
      'volume', 'momentum', 'trend', 'breakout', 'consolidation', 'divergence',
      'fibonacci', 'chart pattern', 'candlestick', 'oscillator', 'volatility'
    ]);

    // Fundamental analysis terms
    private static readonly FUNDAMENTAL_FACTORS = new Set([
      'PE ratio', 'P/E', 'EPS', 'revenue', 'earnings', 'profit margin', 'cash flow',
      'debt', 'assets', 'liabilities', 'market cap', 'dividend', 'book value',
      'ROE', 'ROI', 'EBITDA', 'gross margin', 'operating margin', 'net margin'
    ]);

    // Market sectors
    private static readonly SECTORS = new Set([
      'technology', 'finance', 'healthcare', 'energy', 'consumer', 'industrial',
      'utilities', 'materials', 'real estate', 'telecommunications'
    ]);

    // Economic indicators
    private static readonly ECONOMIC_INDICATORS = new Set([
      'GDP', 'inflation', 'interest rates', 'unemployment', 'CPI', 'PPI',
      'retail sales', 'housing starts', 'PMI', 'trade balance', 'fed', 'fomc'
    ]);

    // Risk-related terms
    private static readonly RISK_TERMS = {
      high: ['volatile', 'risky', 'speculative', 'aggressive', 'unstable', 'uncertain'],
      medium: ['moderate', 'balanced', 'mixed', 'neutral'],
      low: ['safe', 'stable', 'conservative', 'defensive', 'reliable']
    };

    // Sentiment terms
    private static readonly SENTIMENT_TERMS = {
      bullish: ['up', 'rise', 'grow', 'positive', 'bullish', 'outperform', 'buy', 'strong'],
      bearish: ['down', 'fall', 'decline', 'negative', 'bearish', 'underperform', 'sell', 'weak'],
      neutral: ['stable', 'steady', 'hold', 'neutral', 'balanced']
    };

    public static analyzeFinancialContext(message: string): {
      confidence: number;
      instruments: string[];
      technicalIndicators: string[];
      fundamentalFactors: string[];
      sectors: string[];
      economicIndicators: string[];
      riskLevel: 'low' | 'medium' | 'high';
      sentiment: 'bullish' | 'bearish' | 'neutral';
      analysisType: 'technical' | 'fundamental' | 'mixed' | 'general';
    } {
      const text = message.toLowerCase();
      const words = text.split(/\s+/);

      // Extract found items
      const instruments = this.extractInstruments(text);
      const technical = this.extractTechnicalIndicators(text);
      const fundamental = this.extractFundamentalFactors(text);
      const sectors = this.extractSectors(text);
      const economic = this.extractEconomicIndicators(text);

      // Determine analysis type
      const analysisType = this.determineAnalysisType(technical.length, fundamental.length);

      // Calculate risk level
      const riskLevel = this.assessRiskLevel(text);

      // Determine sentiment
      const sentiment = this.determineSentiment(text);

      // Calculate confidence score based on multiple factors
      const confidence = this.calculateConfidence({
        instruments: instruments.length,
        technical: technical.length,
        fundamental: fundamental.length,
        sectors: sectors.length,
        economic: economic.length,
        hasKnownTerms: this.hasKnownFinancialTerms(text),
        messageLength: words.length,
        isQuestionFormat: text.includes('?'),
        hasNumbers: /\d+/.test(text)
      });

      return {
        confidence,
        instruments,
        technicalIndicators: technical,
        fundamentalFactors: fundamental,
        sectors,
        economicIndicators: economic,
        riskLevel,
        sentiment,
        analysisType
      };
    }

    private static extractInstruments(text: string): string[] {
      return Array.from(this.KNOWN_STOCKS).filter(stock =>
        text.includes(stock.toLowerCase())
      );
    }

    private static extractTechnicalIndicators(text: string): string[] {
      return Array.from(this.TECHNICAL_INDICATORS).filter(indicator =>
        text.includes(indicator.toLowerCase())
      );
    }

    private static extractFundamentalFactors(text: string): string[] {
      return Array.from(this.FUNDAMENTAL_FACTORS).filter(factor =>
        text.includes(factor.toLowerCase())
      );
    }

    private static extractSectors(text: string): string[] {
      return Array.from(this.SECTORS).filter(sector =>
        text.includes(sector.toLowerCase())
      );
    }

    private static extractEconomicIndicators(text: string): string[] {
      return Array.from(this.ECONOMIC_INDICATORS).filter(indicator =>
        text.includes(indicator.toLowerCase())
      );
    }

    private static assessRiskLevel(text: string): 'low' | 'medium' | 'high' {
      let riskScore = 0;

      // Check for risk-related terms
      for (const term of this.RISK_TERMS.high) {
        if (text.includes(term)) riskScore += 2;
      }
      for (const term of this.RISK_TERMS.low) {
        if (text.includes(term)) riskScore -= 2;
      }

      // Adjust based on other factors
      if (text.includes('hedge') || text.includes('diversif')) riskScore -= 1;
      if (text.includes('leverage') || text.includes('margin')) riskScore += 1;

      if (riskScore > 2) return 'high';
      if (riskScore < -1) return 'low';
      return 'medium';
    }

    private static determineSentiment(text: string): 'bullish' | 'bearish' | 'neutral' {
      let sentimentScore = 0;

      // Check sentiment terms
      for (const term of this.SENTIMENT_TERMS.bullish) {
        if (text.includes(term)) sentimentScore += 1;
      }
      for (const term of this.SENTIMENT_TERMS.bearish) {
        if (text.includes(term)) sentimentScore -= 1;
      }

      // Additional context analysis
      if (text.includes('buy') || text.includes('long')) sentimentScore += 1;
      if (text.includes('sell') || text.includes('short')) sentimentScore -= 1;

      if (sentimentScore > 1) return 'bullish';
      if (sentimentScore < -1) return 'bearish';
      return 'neutral';
    }

    private static determineAnalysisType(technicalCount: number, fundamentalCount: number): 'technical' | 'fundamental' | 'mixed' | 'general' {
      if (technicalCount > 0 && fundamentalCount > 0) return 'mixed';
      if (technicalCount > 0) return 'technical';
      if (fundamentalCount > 0) return 'fundamental';
      return 'general';
    }

    private static hasKnownFinancialTerms(text: string): boolean {
      const financialTerms = [
        'market', 'stock', 'trade', 'invest', 'price', 'value',
        'portfolio', 'analysis', 'performance', 'return'
      ];
      return financialTerms.some(term => text.includes(term));
    }

    private static calculateConfidence(factors: {
      instruments: number;
      technical: number;
      fundamental: number;
      sectors: number;
      economic: number;
      hasKnownTerms: boolean;
      messageLength: number;
      isQuestionFormat: boolean;
      hasNumbers: boolean;
    }): number {
      let confidence = 0.5; // Base confidence

      // Adjust based on identified elements
      if (factors.instruments > 0) confidence += 0.15;
      if (factors.technical > 0) confidence += 0.1;
      if (factors.fundamental > 0) confidence += 0.1;
      if (factors.sectors > 0) confidence += 0.05;
      if (factors.economic > 0) confidence += 0.05;

      // Adjust for question quality
      if (factors.hasKnownTerms) confidence += 0.1;
      if (factors.isQuestionFormat) confidence += 0.05;
      if (factors.hasNumbers) confidence += 0.05;

      // Penalize very short messages
      if (factors.messageLength < 3) confidence -= 0.2;

      // Penalize if no specific elements found
      if (factors.instruments + factors.technical + factors.fundamental + factors.sectors + factors.economic === 0) {
        confidence -= 0.3;
      }

      // Ensure confidence stays within bounds
      return Math.max(0.1, Math.min(0.95, confidence));
    }
  }
```

```
// app/lib/pdf-utils.ts
import { encode } from 'base64-js';

export async function convertPDFToBase64(file: File): Promise<string> {
  return new Promise((resolve, reject) => {
    const reader = new FileReader();
    reader.onload = () => {
      const arrayBuffer = reader.result as ArrayBuffer;
      const uint8Array = new Uint8Array(arrayBuffer);
      const base64String = encode(uint8Array);
      resolve(base64String);
    };
    reader.onerror = reject;
    reader.readAsArrayBuffer(file);
  });
}

export interface PDFMessage {
  role: 'user';
  content: {
    type: 'document' | 'text';
    source?: {
      type: 'base64';
      media_type: 'application/pdf';
      data: string;
    };
    text?: string;
  }[];
}

export function createPDFMessage(base64Data: string, query: string): PDFMessage {
  return {
    role: 'user',
    content: [
      {
        type: 'document',
        source: {
          type: 'base64',
          media_type: 'application/pdf',
          data: base64Data
        }
      },
      {
        type: 'text',
        text: query
      }
    ]
  };
}
```

```
import {
  BedrockAgentRuntimeClient,
  RetrieveCommand,
  RetrieveCommandInput,
} from "@aws-sdk/client-bedrock-agent-runtime";
import { type ClassValue, clsx } from "clsx";
import { twMerge } from "tailwind-merge";

console.log("ð\237\224\221 Have AWS AccessKey?", !!process.env.BAWS_ACCESS_KEY_ID);
console.log("ð\237\224\221 Have AWS Secret?", !!process.env.BAWS_SECRET_ACCESS_KEY);

const bedrockClient = new BedrockAgentRuntimeClient({
  region: "us-east-1", // Make sure this matches your Bedrock region
  credentials: {
    accessKeyId: process.env.BAWS_ACCESS_KEY_ID!,
    secretAccessKey: process.env.BAWS_SECRET_ACCESS_KEY!,
  },
});

export function cn(...inputs: ClassValue[]) {
  return twMerge(clsx(inputs));
}

export interface RAGSource {
  id: string;
  fileName: string;
  snippet: string;
  score: number;
}

export async function retrieveContext(
  query: string,
  knowledgeBaseId: string,
  n: number = 3,
): Promise<{
  context: string;
  isRagWorking: boolean;
  ragSources: RAGSource[];
}> {
  try {
    if (!knowledgeBaseId) {
      console.error("knowledgeBaseId is not provided");
      return {
        context: "",
        isRagWorking: false,
        ragSources: [],
      };
    }

    const input: RetrieveCommandInput = {
      knowledgeBaseId: knowledgeBaseId,
      retrievalQuery: { text: query },
      retrievalConfiguration: {
        vectorSearchConfiguration: { numberOfResults: n },
      },
    };

    const command = new RetrieveCommand(input);
    const response = await bedrockClient.send(command);

    // Parse results
    const rawResults = response?.retrievalResults || [];
    const ragSources: RAGSource[] = rawResults
      .filter((res: any) => res.content && res.content.text)
      .map((result: any, index: number) => {
        const uri = result?.location?.s3Location?.uri || "";
        const fileName = uri.split("/").pop() || `Source-${index}.txt`;

        return {
          id:
            result.metadata?.["x-amz-bedrock-kb-chunk-id"] || `chunk-${index}`,
          fileName: fileName.replace(/_/g, " ").replace(".txt", ""),
          snippet: result.content?.text || "",
          score: result.score || 0,
        };
      })
      .slice(0, 1);

    console.log("ð\237\224\215 Parsed RAG Sources:", ragSources); // Debug log

    const context = rawResults
      .filter((res: any) => res.content && res.content.text)
      .map((res: any) => res.content.text)
      .join("\n\n");

    return {
      context,
      isRagWorking: true,
      ragSources,
    };
  } catch (error) {
    console.error("RAG Error:", error);
    return { context: "", isRagWorking: false, ragSources: [] };
  }
}
```

```
// // lib/api/market.ts

// const TWELVE_DATA_API_KEY = process.env.NEXT_PUBLIC_TWELVE_DATA_API_KEY;
// const BASE_URL = 'https://api.twelvedata.com';

// export const DEFAULT_SYMBOLS = [
//   'AAPL', 'AMZN', 'MSFT', 'NVDA', 'TSLA',
//   'GOOGL', 'V', 'TSM', 'BRK.A', 'WMT'
// ];

// export async function getQuotes(symbols: string[] = DEFAULT_SYMBOLS) {
//   try {
//     const response = await fetch(
//       `${BASE_URL}/quote?symbol=${symbols.join(',')}&apikey=${TWELVE_DATA_API_KEY}`
//     );
//     const data = await response.json();
//     return Array.isArray(data) ? data : [data];
//   } catch (error) {
//     console.error('Error fetching quotes:', error);
//     return [];
//   }
// }

// export async function getTimeSeries(
//   symbol: string,
//   interval: string = '1day',
//   outputsize: number = 252 // One year of trading days
// ) {
//   try {
//     const response = await fetch(
//       `${BASE_URL}/time_series?symbol=${symbol}&interval=${interval}&outputsize=${outputsize}&apikey=${TWELV
E_DATA_API_KEY}`
//     );
//     const data = await response.json();
//     return data.values || [];
//   } catch (error) {
//     console.error('Error fetching time series:', error);
//     return [];
//   }
// }
// // services/saudi-market.ts

// import { API_KEY, API_URL } from ’@/app/config/market’;

// export interface SaudiSymbol {
//   symbol: string;
//   name: string;
//   currency: string;
//   exchange: string;
//   mic_code: string;
//   type: string;
//   logo_url?: string; // Will be added from another API or local storage
// }

// export async function fetchSaudiSymbols(): Promise<SaudiSymbol[]> {
//   try {
//     // Fetch all stocks with Saudi Arabia filter
//     const response = await fetch(
//       ‘${API_URL}/stocks?country=Saudi Arabia&apikey=${API_KEY}‘
//     );

//     if (!response.ok) {
//       throw new Error(’Failed to fetch Saudi stocks’);
//     }

//     const data = await response.json();
//     const stocks = Array.isArray(data) ? data : data.data || [];

//     // Map and add logo URLs
//     // We’ll need to either:
//     // 1. Use another API to get logos
//     // 2. Store logos locally
//     // 3. Generate placeholder logos
//     return stocks.map(stock => ({
//       ...stock,
//       logo_url: ‘/logos/${stock.symbol}.png‘ // This path will need to be adjusted
//     }));

//   } catch (error) {
//     console.error(’Error fetching Saudi symbols:’, error);
//     throw error;
//   }
// }

// // Function to get logo URL for a symbol
// export async function getCompanyLogo(symbol: string): Promise<string> {
//   // We can implement this in several ways:
//   // 1. Use a financial API that provides logos
//   // 2. Scrape from Tadawul website
//   // 3. Store logos locally
//   // 4. Generate placeholder logos

//   // For now, return a placeholder
//   return ‘https://ui-avatars.com/api/?name=${encodeURIComponent(symbol)}&background=random‘;
// }

// // Function to get company details
// export interface CompanyDetails {
//   symbol: string;
//   name: string;
//   sector: string;
//   industry: string;
//   website?: string;
//   description?: string;
//   market_cap?: number;
//   employees?: number;
//   ceo?: string;
//   headquarters?: string;
//   founded?: string;
// }

// export async function getCompanyDetails(symbol: string): Promise<CompanyDetails | null> {
//   try {
//     // This would need to be implemented with actual data source
//     // Could be from Tadawul API or another financial data provider
//     return null;
//   } catch (error) {
//     console.error(‘Error fetching details for ${symbol}:‘, error);
//     return null;
//   }
// }
// /app/lib/services/saudiFundamentals.ts

import { API_KEY, API_URL, SAUDI_SYMBOLS } from '@/app/config/market';

// Type Definitions
export interface FiftyTwoWeek {
  high: number;
  low: number;
  high_date: string;
  low_date: string;
  high_change: number;
  low_change: number;
  range: string;
}

export interface FinancialRatios {
  quick_ratio: number;
  current_ratio: number;
  debt_to_equity: number;
  gross_margin: number;
  operating_margin: number;
  net_profit_margin: number;
  return_on_assets: number;
  return_on_equity: number;
  asset_turnover: number;
}

export interface ValuationMetrics {
  price_earnings: number;
  price_to_book: number;
  price_to_sales: number;
  ev_to_ebitda: number;
  enterprise_value: number;
  market_cap: number;
}

export interface FundamentalData {
  symbol: string;
  name: string;
  sector: string;
  earnings_per_share: number;
  dividend_yield: number;
  book_value: number;
  beta: number;
  fifty_two_week: FiftyTwoWeek;
  financial_ratios: FinancialRatios;
  valuation_metrics: ValuationMetrics;
  growth_metrics?: {
    revenue_growth: number;
    earnings_growth: number;
    dividend_growth: number;
  };
}

export interface FinancialStatement {
  symbol: string;
  period: string;
  date: string;
  currency: string;
  // Income Statement
  revenue: number;
  gross_profit: number;
  operating_income: number;
  net_income: number;
  eps: number;
  ebitda: number;
  // Balance Sheet
  total_assets: number;
  total_liabilities: number;
  total_equity: number;
  cash_and_equivalents: number;
  total_debt: number;
  // Cash Flow
  cash_flow_operations: number;
  cash_flow_investing: number;
  cash_flow_financing: number;
  free_cash_flow: number;
  // Year over Year changes
  yoy_revenue_growth?: number;
  yoy_earnings_growth?: number;
  yoy_fcf_growth?: number;
}

export interface SectorMetrics {
  sector: string;
  average_pe: number;
  average_pb: number;
  average_dividend_yield: number;
  average_market_cap: number;
  best_performer: {
    symbol: string;
    name: string;
    return: number;
  };
  worst_performer: {
    symbol: string;
    name: string;
    return: number;
  };
  total_market_cap: number;
  companies_count: number;
}

export class SaudiFundamentals {
  private static readonly DEFAULT_TIMEOUT = 10000; // 10 seconds
  private static readonly CACHE_DURATION = 300000; // 5 minutes
  private static cache: Map<string, { data: any; timestamp: number }> = new Map();

  private static safeNumber(value: any, defaultValue: number = 0): number {
    if (value === null || value === undefined) return defaultValue;
    const parsed = parseFloat(value);
    return isNaN(parsed) ? defaultValue : parsed;
  }

  private static safeDate(date: any): string {
    if (!date) return '';
    try {
      const parsed = new Date(date);
      return parsed.toISOString();
    } catch {
      return '';
    }
  }

  private static async fetchWithTimeout(url: string, timeout: number = this.DEFAULT_TIMEOUT): Promise<Response>
 {
    const controller = new AbortController();
    const id = setTimeout(() => controller.abort(), timeout);

    try {
      const response = await fetch(url, { signal: controller.signal });
      clearTimeout(id);
      return response;
    } catch (error) {
      clearTimeout(id);
      throw error;
    }
  }

  private static async fetchWithRetry(url: string, retries = 3): Promise<any> {
    const cacheKey = url;
    const cached = this.cache.get(cacheKey);

    if (cached && Date.now() - cached.timestamp < this.CACHE_DURATION) {
      return cached.data;
    }

    let lastError: Error | null = null;

    for (let i = 0; i < retries; i++) {
      try {
        const response = await this.fetchWithTimeout(url);
        if (!response.ok) throw new Error(`HTTP error! status: ${response.status}`);

        const data = await response.json();
        this.cache.set(cacheKey, { data, timestamp: Date.now() });
        return data;
      } catch (error) {
        lastError = error instanceof Error ? error : new Error('Unknown error occurred');
        if (i < retries - 1) await new Promise(resolve => setTimeout(resolve, 1000 * (i + 1)));
      }
    }

    throw lastError || new Error('Failed to fetch data');
  }

  private static formatSymbol(symbol: string): string {
    return symbol.replace(':TADAWUL', '') + '.SAU';
  }

  static async getFundamentals(symbol: string): Promise<FundamentalData | null> {
    try {
      const formattedSymbol = this.formatSymbol(symbol);
      const stockInfo = SAUDI_SYMBOLS.find(s => s.symbol === symbol);

      if (!stockInfo) return null;

      const endpoints = {
        fundamentals: `${API_URL}/fundamentals?symbol=${formattedSymbol}&apikey=${API_KEY}`,
        quote: `${API_URL}/quote?symbol=${formattedSymbol}&apikey=${API_KEY}`,
        earnings: `${API_URL}/earnings?symbol=${formattedSymbol}&apikey=${API_KEY}`
      };

      const [fundamentals, quote, earnings] = await Promise.all([
        this.fetchWithRetry(endpoints.fundamentals),
        this.fetchWithRetry(endpoints.quote),
        this.fetchWithRetry(endpoints.earnings)
      ]);

      const valuationMetrics: ValuationMetrics = {
        price_earnings: this.safeNumber(fundamentals.pe_ratio),
        price_to_book: this.safeNumber(fundamentals.price_to_book_ratio),
        price_to_sales: this.safeNumber(fundamentals.price_to_sales_ratio),
        ev_to_ebitda: this.safeNumber(fundamentals.ev_to_ebitda),
        enterprise_value: this.safeNumber(fundamentals.enterprise_value),
        market_cap: this.safeNumber(fundamentals.market_cap)
      };

      return {
        symbol,
        name: stockInfo.name,
        sector: stockInfo.sector,
        earnings_per_share: this.safeNumber(earnings.eps),
        dividend_yield: this.safeNumber(fundamentals.dividend_yield),
        book_value: this.safeNumber(fundamentals.book_value),
        beta: this.safeNumber(fundamentals.beta),
        fifty_two_week: {
          high: this.safeNumber(quote.fifty_two_week?.high),
          low: this.safeNumber(quote.fifty_two_week?.low),
          high_date: this.safeDate(quote.fifty_two_week?.high_date),
          low_date: this.safeDate(quote.fifty_two_week?.low_date),
          high_change: this.safeNumber(quote.fifty_two_week?.high_change),
          low_change: this.safeNumber(quote.fifty_two_week?.low_change),
          range: quote.fifty_two_week?.range || 'N/A'
        },
        financial_ratios: {
          quick_ratio: this.safeNumber(fundamentals.quick_ratio),
          current_ratio: this.safeNumber(fundamentals.current_ratio),
          debt_to_equity: this.safeNumber(fundamentals.debt_to_equity),
          gross_margin: this.safeNumber(fundamentals.gross_margin),
          operating_margin: this.safeNumber(fundamentals.operating_margin),
          net_profit_margin: this.safeNumber(fundamentals.net_profit_margin),
          return_on_assets: this.safeNumber(fundamentals.roa),
          return_on_equity: this.safeNumber(fundamentals.roe),
          asset_turnover: this.safeNumber(fundamentals.asset_turnover)
        },
        valuation_metrics,
        growth_metrics: {
          revenue_growth: this.safeNumber(fundamentals.revenue_growth),
          earnings_growth: this.safeNumber(earnings.growth),
          dividend_growth: this.safeNumber(fundamentals.dividend_growth)
        }
      };
    } catch (error) {
      console.error('Error fetching fundamentals for', symbol, ':', error);
      return null;
    }
  }

  static async getFinancialStatements(
    symbol: string,
    period: 'quarterly' | 'annual' = 'annual'
  ): Promise<FinancialStatement[]> {
    try {
      const formattedSymbol = this.formatSymbol(symbol);
      const url = `${API_URL}/financial_statement?symbol=${formattedSymbol}&period=${period}&apikey=${API_KEY}`
;

      const response = await this.fetchWithRetry(url);
      if (!response.financials?.length) return [];

      return response.financials.map((statement: any, index: number, array: any[]) => {
        const current = {
          symbol,
          period,
          date: statement.date || '',
          currency: statement.currency || 'SAR',
          revenue: this.safeNumber(statement.revenue),
          gross_profit: this.safeNumber(statement.gross_profit),
          operating_income: this.safeNumber(statement.operating_income),
          net_income: this.safeNumber(statement.net_income),
          eps: this.safeNumber(statement.eps),
          ebitda: this.safeNumber(statement.ebitda),
          total_assets: this.safeNumber(statement.total_assets),
          total_liabilities: this.safeNumber(statement.total_liabilities),
          total_equity: this.safeNumber(statement.total_equity),
          cash_and_equivalents: this.safeNumber(statement.cash_and_equivalents),
          total_debt: this.safeNumber(statement.total_debt),
          cash_flow_operations: this.safeNumber(statement.cash_flow_operating),
          cash_flow_investing: this.safeNumber(statement.cash_flow_investing),
          cash_flow_financing: this.safeNumber(statement.cash_flow_financing),
          free_cash_flow: this.safeNumber(statement.free_cash_flow)
        };

        // Calculate year-over-year growth metrics if previous period exists
        if (index < array.length - 1) {
          const previous = array[index + 1];
          current.yoy_revenue_growth = this.calculateGrowth(current.revenue, previous.revenue);
          current.yoy_earnings_growth = this.calculateGrowth(current.net_income, previous.net_income);
          current.yoy_fcf_growth = this.calculateGrowth(current.free_cash_flow, previous.free_cash_flow);
        }

        return current;
      });
    } catch (error) {
      console.error('Error fetching financial statements for', symbol, ':', error);
      return [];
    }
  }

  static async getSectorComparison(sector: string): Promise<SectorMetrics> {
    try {
      const sectorCompanies = SAUDI_SYMBOLS.filter(stock => stock.sector === sector);
      const fundamentalsPromises = sectorCompanies.map(company => this.getFundamentals(company.symbol));
      const results = await Promise.allSettled(fundamentalsPromises);

      const validResults = results
        .filter((result): result is PromiseFulfilledResult<FundamentalData | null> =>
          result.status === 'fulfilled' && result.value !== null
        )
        .map(result => result.value!);

      if (!validResults.length) throw new Error('No valid fundamental data found for sector');

      const returns = validResults.map(company => ({
        symbol: company.symbol,
        name: company.name,
        return: company.growth_metrics?.earnings_growth || 0
      }));

      returns.sort((a, b) => b.return - a.return);

      return {
        sector,
        average_pe: this.calculateAverage(validResults, r => r.valuation_metrics.price_earnings),
        average_pb: this.calculateAverage(validResults, r => r.valuation_metrics.price_to_book),
        average_dividend_yield: this.calculateAverage(validResults, r => r.dividend_yield),
        average_market_cap: this.calculateAverage(validResults, r => r.valuation_metrics.market_cap),
        best_performer: returns[0],
        worst_performer: returns[returns.length - 1],
        total_market_cap: validResults.reduce((sum, company) => sum + company.valuation_metrics.market_cap, 0),
        companies_count: validResults.length
      };
    } catch (error) {
      console.error('Error fetching sector comparison:', error);
      throw error;
    }
  }

  private static calculateGrowth(current: number, previous: number): number {
    if (!previous) return 0;
    return ((current - previous) / Math.abs(previous)) * 100;
  }

  private static calculateAverage(data: any[], selector: (item: any) => number): number {
    const values = data.map(selector).filter(v => !isNaN(v) && v !== null);
    return values.length ? values.reduce((a, b) => a + b, 0) / values.length : 0;
  }
}
```

```
// /app/api/chat/route.ts
import Anthropic from "@anthropic-ai/sdk";
import { z } from "zod";
import crypto from "crypto";
import { FinanceAnalyzer } from "@/app/lib/context/financeAnalyzer";
import { FinanceContextManager } from "@/app/lib/context/financeContextManager";
import { FinanceContextAnalyzer } from "@/app/lib/context/financeContextUtils";
import { SaudiMarketHandler } from "@/app/market/saudiMarketHandler";
import { SaudiFundamentals } from '@/app/lib/services/saudiFundamentals';


// Initialize Anthropic client
const anthropic = new Anthropic({
  apiKey: process.env.ANTHROPIC_API_KEY,
  defaultHeaders: {
    "anthropic-beta": "pdfs-2024-09-25"
  }
});

// Initialize singleton context manager
const contextManager = new FinanceContextManager();

// Enhanced response schema with Saudi market data
const responseSchema = z.object({
  response: z.string(),
  thinking: z.string(),
  user_mood: z.enum(["positive", "neutral", "negative", "curious", "frustrated", "confused"]),
  suggested_questions: z.array(z.string()),
  financial_context: z.object({
    instruments_mentioned: z.array(z.string()),
    market_sentiment: z.enum(["bullish", "bearish", "neutral"]),
    risk_level: z.enum(["low", "medium", "high"]),
    confidence_score: z.number(),
    technical_indicators: z.array(z.string()),
    fundamental_factors: z.array(z.string()),
    sectors: z.array(z.string()),
    economic_indicators: z.array(z.string()),
    analysis_type: z.enum(["technical", "fundamental", "mixed", "general"]),
    market_data: z.object({
      quotes: z.array(z.object({
        symbol: z.string(),
        name: z.string(),
        price: z.number().nonnegative(),
        change: z.number().default(0),
        percent_change: z.number().default(0),
        volume: z.number().nonnegative().default(0),
        timestamp: z.string().optional(),
        sector: z.string().optional()
      })).default([]),
      market_details: z.object({
        exchange: z.string(),
        currency: z.string(),
        trading_hours: z.string(),
        data_delay: z.string()
      }).optional(),
      recent_performance: z.array(z.any()).optional()
    }).optional().nullable(),
    key_insights: z.array(z.string())
  }),
  debug: z.object({
    context_used: z.boolean(),
    market_data_used: z.boolean(),
    context_confidence: z.number(),
    analysis_quality: z.string()
  })
});

export async function POST(req: Request) {
  try {
    const { messages, pdfData, conversationId = crypto.randomUUID() } = await req.json();
    const latestMessage = messages[messages.length - 1].content;

    // Analyze financial context
    const baseAnalysis = FinanceAnalyzer.analyzeFinancialContext(latestMessage);
    const contextAnalysis = FinanceContextAnalyzer.analyzeFinancialMessage(latestMessage);

    // Analyze Saudi market specific companies and sectors
    const saudiMarketAnalysis = await SaudiMarketHandler.analyzeMentionedCompanies(latestMessage);

    // Fetch financial data for mentioned companies
    const financialDataPromises = saudiMarketAnalysis.companies.map(company =>
      SaudiFundamentals.getFundamentals(company.symbol)
    );
    const financialDataResults = await Promise.all(financialDataPromises);

    // Combine all analyses
    const combinedAnalysis = {
      ...baseAnalysis,
      confidence: Math.max(baseAnalysis.confidence, contextAnalysis.confidenceScore || 0),
      sentiment: contextAnalysis.sentiment || baseAnalysis.sentiment,
      technicalIndicators: [
        ...new Set([
          ...baseAnalysis.technicalIndicators,
          ...(contextAnalysis.technicalIndicators || [])
        ])
      ],
      fundamentalFactors: [
        ...new Set([
          ...baseAnalysis.fundamentalFactors,
          ...(contextAnalysis.fundamentalFactors || [])
        ])
      ],
      sectors: Array.from(saudiMarketAnalysis.sectors)
    };

    // Update context manager
    const messageContext = {
      id: crypto.randomUUID(),
      role: 'user',
      content: latestMessage,
      timestamp: Date.now(),
      metadata: {
        instruments: combinedAnalysis.instruments,
        sentiment: combinedAnalysis.sentiment,
        riskLevel: combinedAnalysis.riskLevel,
        confidenceScore: combinedAnalysis.confidence,
        technicalIndicators: combinedAnalysis.technicalIndicators,
        fundamentalFactors: combinedAnalysis.fundamentalFactors,
        topics: combinedAnalysis.sectors,
        markets: combinedAnalysis.instruments
      }
    };

    contextManager.addMessage(conversationId, messageContext);
    const historicalContext = contextManager.getContextForPrompt(conversationId);

    // Get market data for mentioned companies
    let marketData = null;
    if (saudiMarketAnalysis.companies.length > 0) {
      const companyData = await Promise.all(
        saudiMarketAnalysis.companies.slice(0, 3).map(company =>
          SaudiMarketHandler.getCompanyData(company.symbol)
        )
      );

      // Integrate financial data with market data
      marketData = {
        quotes: saudiMarketAnalysis.companies.map((company, index) => ({
          ...company,
          fundamentals: financialDataResults[index]
        })),
        market_details: companyData[0]?.market_details,
        recent_performance: companyData[0]?.recent_performance
      };
    }

    // Build system context based on analysis
    let systemContext = [
      "You are a highly knowledgeable Saudi market financial advisor and analyst.",
      combinedAnalysis.analysisType === 'technical' && "Focus on technical analysis and chart patterns.",
      combinedAnalysis.analysisType === 'fundamental' && "Focus on fundamental analysis and company metrics.",
      combinedAnalysis.riskLevel === 'high' && "Be sure to emphasize risk factors and potential downsides.",
      historicalContext && `Previous context: ${historicalContext}`,
      marketData && `Current market data: ${JSON.stringify(marketData)}`,
      `Current market sentiment: ${combinedAnalysis.sentiment}`,
      `Analysis confidence: ${(combinedAnalysis.confidence * 100).toFixed(1)}%`
    ].filter(Boolean).join("\n");

    // Prepare message content
    const messageContent = pdfData ? [{
      role: 'user',
      content: [
        {
          type: 'document',
          source: {
            type: 'base64',
            media_type: 'application/pdf',
            data: pdfData,
          },
        },
        {
          type: 'text',
          text: latestMessage,
        },
      ],
    }] : [{
      role: 'user',
      content: [{
        type: 'text',
        text: `${systemContext}\n\nUser query: ${latestMessage}`
      }],
    }];

    // Get response from Claude
    const response = await anthropic.messages.create({
      model: "claude-3-5-sonnet-20241022",
      max_tokens: 4096,
      messages: messageContent,
    });

    // Generate suggested questions based on context
    const suggestedQuestions = generateFinancialQuestions(
      combinedAnalysis,
      contextManager.getKeyInsights(conversationId),
      saudiMarketAnalysis
    );

    // Calculate analysis quality
    const analysisQuality = calculateAnalysisQuality(
      combinedAnalysis,
      historicalContext,
      saudiMarketAnalysis
    );

    // Prepare response data
    const responseData = {
      response: response.content[0].text,
      thinking: "Analyzing Saudi market data and financial context",
      user_mood: determineUserMood(latestMessage, combinedAnalysis),
      suggested_questions: suggestedQuestions,
      financial_context: {
        instruments_mentioned: combinedAnalysis.instruments,
        market_sentiment: combinedAnalysis.sentiment,
        risk_level: combinedAnalysis.riskLevel,
        confidence_score: combinedAnalysis.confidence,
        technical_indicators: combinedAnalysis.technicalIndicators,
        fundamental_factors: combinedAnalysis.fundamentalFactors,
        sectors: combinedAnalysis.sectors,
        economic_indicators: combinedAnalysis.economicIndicators,
        analysis_type: combinedAnalysis.analysisType,
        market_data: marketData,
        key_insights: contextManager.getKeyInsights(conversationId)
      },
      debug: {
        context_used: !!historicalContext,
        market_data_used: !!marketData,
        context_confidence: contextManager.getActiveInstruments(conversationId).length > 0 ? 0.9 : 0.5,
        analysis_quality: analysisQuality
      }
    };

    // Validate response
    const validatedResponse = responseSchema.parse(responseData);

    return new Response(JSON.stringify({
      id: crypto.randomUUID(),
      conversationId,
      ...validatedResponse,
    }), {
      status: 200,
      headers: { "Content-Type": "application/json" }
    });

  } catch (error) {
    console.error("Error in chat handler:", error);
    return new Response(
      JSON.stringify({
        response: "An error occurred while processing your Saudi market query. Please try again.",
        thinking: "Error in analysis",
        user_mood: "frustrated",
        suggested_questions: [
          "Would you like to view the current market overview?",
          "Should we look at a specific sector instead?",
          "Would you like to know about top performing stocks?"
        ],
        financial_context: {
          instruments_mentioned: [],
          market_sentiment: "neutral",
          risk_level: "medium",
          confidence_score: 0,
          technical_indicators: [],
          fundamental_factors: [],
          sectors: [],
          economic_indicators: [],
          analysis_type: "general",
          market_data: null,
          key_insights: []
        },
        debug: {
          context_used: false,
          market_data_used: false,
          context_confidence: 0,
          analysis_quality: "error"
        }
      }),
      {
        status: 500,
        headers: { "Content-Type": "application/json" }
      }
    );
  }
}

function generateFinancialQuestions(
  analysis: any,
  insights: string[],
  marketAnalysis: { companies: any[], sectors: Set<string> }
): string[] {
  const questions = [];

  // Add Saudi market specific questions
  if (marketAnalysis.companies.length > 0) {
    const company = marketAnalysis.companies[0];
    questions.push(
      `What’s your analysis of ${company.name}’s recent performance?`,
      `How does ${company.name} compare to its sector peers?`
    );
  }

  if (marketAnalysis.sectors.size > 0) {
    const sector = Array.from(marketAnalysis.sectors)[0];
    questions.push(
      `What’s your outlook for the ${sector} sector?`,
      `Which companies in the ${sector} sector show the most promise?`
    );
  }

  // Add analysis-type specific questions
  if (analysis.technicalIndicators.length > 0) {
    questions.push(
      `How do the ${analysis.technicalIndicators[0]} levels compare to historical trends?`
    );
  }

  if (analysis.fundamentalFactors.length > 0) {
    questions.push(
      `How do these ${analysis.fundamentalFactors[0]} metrics compare to sector averages?`
    );
  }

  // Add general market questions if needed
  if (questions.length < 3) {
    questions.push(
      "What are the key factors driving the Saudi market today?",
      "Which sectors are showing the strongest momentum?",
      "How might current economic conditions affect the market?"
    );
  }

  // Return top 3 most relevant questions
  return questions.slice(0, 3);
}

function determineUserMood(
  message: string,
  analysis: any
): "positive" | "neutral" | "negative" | "curious" | "frustrated" | "confused" {
  const text = message.toLowerCase();

  if (text.includes(’help’) || text.includes(’?’)) return ’curious’;
  if (text.includes(’wrong’) || text.includes(’error’)) return ’frustrated’;
  if (text.includes(’not sure’) || text.includes(’confused’)) return ’confused’;

  if (analysis.confidence > 0.8) {
    return analysis.sentiment === ’bullish’ ? ’positive’ :
           analysis.sentiment === ’bearish’ ? ’negative’ : ’neutral’;
  }

  return analysis.confidence < 0.5 ? ’curious’ : ’neutral’;
}

function calculateAnalysisQuality(
  analysis: any,
  historicalContext: string | null,
  marketAnalysis: { companies: any[], sectors: Set<string> }
): string {
  const factors = {
    hasCompanies: marketAnalysis.companies.length > 0,
    hasSectors: marketAnalysis.sectors.size > 0,
    hasTechnical: analysis.technicalIndicators.length > 0,
    hasFundamental: analysis.fundamentalFactors.length > 0,
    hasHistory: !!historicalContext,
    highConfidence: analysis.confidence > 0.7
  };

  const score = Object.values(factors).filter(Boolean).length;

  if (score >= 5) return ’excellent’;
  if (score >= 3) return ’good’;
  if (score >= 1) return ’fair’;
  return ’limited’;
}
```

```
// /app/lib/market/saudiMarketHandler.ts

import { API_KEY, API_URL, SAUDI_SYMBOLS, SAUDI_MARKET } from '@/app/config/market';
import { SaudiFundamentals } from '@/app/lib/services/saudiFundamentals';

interface SaudiQuote {
  symbol: string;
  name: string;
  name_ar?: string;
  price: number;
  change: number;
  percent_change: number;
  volume: number;
  timestamp: string;
  sector?: string;

  fundamentals?: {
    price_earnings: number;
    earnings_per_share: number;
    dividend_yield: number;
    market_cap: number;
    financial_ratios: {
      quick_ratio: number;
      current_ratio: number;
      debt_to_equity: number;
      net_profit_margin: number;
    };
  };
  sector_comparison?: {
    sector_avg_pe: number;
    sector_avg_dividend_yield: number;
    sector_position: 'above_average' | 'below_average' | 'average';
  };
}


export class SaudiMarketHandler {
  private static formatTadawulSymbol(symbol: string): string {
    // Remove any existing suffixes
    const cleanSymbol = symbol.replace(/\.(SAU|TADAWUL)$/, '');
    // Add Tadawul suffix
    return `${cleanSymbol}:Tadawul`;
  }

  private static safeParseNumber(value: any): number {
    if (value === null || value === undefined) return 0;
    const parsed = parseFloat(value);
    return isNaN(parsed) ? 0 : parsed;
  }

  private static async fetchWithRetry(url: string, retries = 3): Promise<any> {
    let lastError;
    for (let i = 0; i < retries; i++) {
      try {
        const response = await fetch(url);
        if (!response.ok) {
          console.log('Response not OK:', await response.text());
          throw new Error(`HTTP error! status: ${response.status}`);
        }
        const data = await response.json();
        console.log('Fetched data:', data); // Debug log
        return data;
      } catch (error) {
        console.error(`Attempt ${i + 1} failed:`, error);
        lastError = error;
        await new Promise(resolve => setTimeout(resolve, 1000 * (i + 1)));
      }
    }
    throw lastError;
  }

  private static formatTimestamp(timestamp: number | string): string {
    if (typeof timestamp === 'number') {
      // Convert UNIX timestamp to ISO string
      return new Date(timestamp * 1000).toISOString();
    }
    return timestamp || new Date().toISOString();
  }

  private static safeParseQuote(data: any, stockInfo: any): SaudiQuote {
    const quote = {
      symbol: stockInfo.tickerSymbol,
      name: stockInfo.name,
      name_ar: stockInfo.name_ar,
      price: this.safeParseNumber(data.close || data.price),
      change: this.safeParseNumber(data.change),
      percent_change: this.safeParseNumber(data.percent_change),
      volume: this.safeParseNumber(data.volume),
      timestamp: this.formatTimestamp(data.timestamp),
      sector: stockInfo.sector,
      trading_info: {
        open: this.safeParseNumber(data.open),
        high: this.safeParseNumber(data.high),
        low: this.safeParseNumber(data.low),
        previous_close: this.safeParseNumber(data.previous_close),
        is_market_open: data.is_market_open || false,
        fifty_two_week: data.fifty_two_week || {
          low: 0,
          high: 0,
          low_change: 0,
          high_change: 0,
          low_change_percent: 0,
          high_change_percent: 0,
          range: ’N/A’
        }
      }
    };
    console.log(’Parsed quote:’, quote); // Debug log
    return quote;
  }

  static async getQuote(query: string): Promise<SaudiQuote | null> {
    try {
      const stockInfo = SAUDI_SYMBOLS.find(stock =>
        stock.tickerSymbol === query ||
        stock.name.toLowerCase().includes(query.toLowerCase()) ||
        (stock.name_ar && stock.name_ar.includes(query))
      );

      if (!stockInfo) {
        console.log(’Stock not found for query:’, query);
        return null;
      }

      const formattedSymbol = this.formatTadawulSymbol(stockInfo.tickerSymbol);
      const url = `${API_URL}/quote?symbol=${formattedSymbol}&apikey=${API_KEY}`;
      console.log(’Fetching URL:’, url); // Debug log

      const data = await this.fetchWithRetry(url);
      return this.safeParseQuote(data, stockInfo);
    } catch (error) {
      console.error(`Error fetching quote for ${query}:`, error);
      return null;
    }
  }

  static async getQuotes(queries: string[]): Promise<SaudiQuote[]> {
    try {
      const validStocks = queries
        .map(query => SAUDI_SYMBOLS.find(stock =>
          stock.tickerSymbol === query ||
          stock.name.toLowerCase().includes(query.toLowerCase()) ||
          (stock.name_ar && stock.name_ar.includes(query))
        ))
        .filter((stock): stock is typeof SAUDI_SYMBOLS[0] => stock !== undefined);

      if (!validStocks.length) return [];

      const symbols = validStocks
        .map(stock => this.formatTadawulSymbol(stock.tickerSymbol))
        .join(’,’);

      console.log(’Fetching quotes for symbols:’, symbols); // Debug log

      const url = `${API_URL}/quote?symbol=${symbols}&apikey=${API_KEY}`;
      const data = await this.fetchWithRetry(url);
      const quotesData = Array.isArray(data) ? data : [data];

      return quotesData.map((quote, index) =>
        this.safeParseQuote(quote, validStocks[index])
      );
    } catch (error) {
      console.error(’Error fetching Saudi market quotes:’, error);
      return [];
    }
  }

  static async getCompanyData(query: string): Promise<any> {
    try {
      const stockInfo = SAUDI_SYMBOLS.find(stock =>
        stock.tickerSymbol === query ||
        stock.name.toLowerCase().includes(query.toLowerCase()) ||
        (stock.name_ar && stock.name_ar.includes(query))
      );

      if (!stockInfo) return null;

      const formattedSymbol = this.formatTadawulSymbol(stockInfo.tickerSymbol);
      const [quote, timeSeries] = await Promise.all([
        this.getQuote(stockInfo.tickerSymbol),
        this.fetchWithRetry(
          `${API_URL}/time_series?symbol=${formattedSymbol}&interval=1day&outputsize=5&apikey=${API_KEY}`
        )
      ]);

      return {
        quote,
        recent_performance: timeSeries.values || [],
        market_details: {
          ...SAUDI_MARKET,
          sector: stockInfo.sector
        }
      };
    } catch (error) {
      console.error('Error fetching Saudi company data:', error);
      return null;
    }
  }

  static async analyzeMentionedCompanies(text: string): Promise<{
    companies: SaudiQuote[];
    sectors: Set<string>;
  }> {
    try {
      const mentionedStocks = SAUDI_SYMBOLS.filter(stock =>
        text.toLowerCase().includes(stock.name.toLowerCase()) ||
        text.includes(stock.tickerSymbol) ||
        (stock.name_ar && text.includes(stock.name_ar))
      );

      console.log('Found mentioned stocks:', mentionedStocks); // Debug log

      const quotes = await this.getQuotes(
        mentionedStocks.map(stock => stock.tickerSymbol)
      );

      const sectors = new Set(
        mentionedStocks.map(stock => stock.sector)
      );

      return {
        companies: quotes,
        sectors
      };
    } catch (error) {
      console.error('Error analyzing mentioned companies:', error);
      return {
        companies: [],
        sectors: new Set()
      };
    }
  }

  static async getMarketOverview(): Promise<any> {
    try {
      // Key market indicators
      const mainSymbols = [
        '1120', // Al Rajhi Bank
        '2222', // Saudi Aramco
        '7010', // STC
        '2010'  // SABIC
      ];

      const quotes = await this.getQuotes(mainSymbols);

      return {
        quotes,
        market_info: SAUDI_MARKET,
        timestamp: new Date().toISOString()
      };
    } catch (error) {
      console.error('Error fetching market overview:', error);
      return null;
    }
  }
}
```

```
@tailwind base;
@tailwind components;
@tailwind utilities;

@layer base {
    :root {
        --background: 0 0% 100%;
        --foreground: 0 0% 3.9%;
        --card: 0 0% 100%;
        --card-foreground: 0 0% 3.9%;
        --popover: 0 0% 100%;
        --popover-foreground: 0 0% 3.9%;
        --primary: 0 0% 9%;
        --primary-foreground: 0 0% 98%;
        --secondary: 0 0% 96.1%;
        --secondary-foreground: 0 0% 9%;
        --muted: 0 0% 96.1%;
        --muted-foreground: 0 0% 45.1%;
        --accent: 0 0% 96.1%;
        --accent-foreground: 0 0% 9%;
        --destructive: 0 84.2% 60.2%;
        --destructive-foreground: 0 0% 98%;
        --border: 0 0% 89.8%;
        --input: 0 0% 89.8%;
        --ring: 0 0% 3.9%;
        --radius: 0.75rem;
        --chart-1: 12 76% 61%;
        --chart-2: 173 58% 39%;
        --chart-3: 197 37% 24%;
        --chart-4: 43 74% 66%;
        --chart-5: 27 87% 67%;
    }

    .dark {
        --background: 0 0% 3.9%;
        --foreground: 0 0% 98%;
        --card: 0 0% 3.9%;
        --card-foreground: 0 0% 98%;
        --popover: 0 0% 3.9%;
        --popover-foreground: 0 0% 98%;
        --primary: 0 0% 98%;
        --primary-foreground: 0 0% 9%;
        --secondary: 0 0% 14.9%;
        --secondary-foreground: 0 0% 98%;
        --muted: 0 0% 14.9%;
        --muted-foreground: 0 0% 63.9%;
        --accent: 0 0% 14.9%;
        --accent-foreground: 0 0% 98%;
        --destructive: 0 62.8% 30.6%;
        --destructive-foreground: 0 0% 98%;
        --border: 0 0% 14.9%;
        --input: 0 0% 14.9%;
        --ring: 0 0% 83.1%;
        --chart-1: 220 70% 50%;
        --chart-2: 160 60% 45%;
        --chart-3: 30 80% 55%;
        --chart-4: 280 65% 60%;
        --chart-5: 340 75% 55%;
    }
}

@layer base {
    * {
        @apply border-border;
    }
    body {
        @apply bg-background text-foreground;
    }
}
```

```
#!/bin/bash

# Function to print usage
print_usage() {
    echo "Usage: $0 <output_pdf> <directory1> [<directory2> ...]"
    echo "Example: $0 all_code.pdf /path/to/project1 /path/to/project2"
}

# Check if at least two arguments are provided
if [ $# -lt 2 ]; then
    print_usage
    exit 1
fi

# Output PDF file name
OUTPUT="$1"
shift

# Temporary PS file
TEMP_PS="temp.ps"

# Array of file extensions to include
EXTENSIONS=("c" "cpp" "h" "hpp" "py" "java" "js" "html" "css" "php" "rb" "go" "rs" "swift" "kt" "scala" "pl" "s
h" "sql" "js" "ts")

# Function to get the appropriate language for enscript
get_language() {
    case "$1" in
        c|h|cpp|hpp) echo "c" ;;
        py) echo "python" ;;
        java) echo "java" ;;
        js|ts) echo "javascript" ;;
        html|css) echo "html" ;;
        php) echo "php" ;;
        rb) echo "ruby" ;;
        go) echo "go" ;;
        rs) echo "rust" ;;
        swift) echo "swift" ;;
        kt) echo "kotlin" ;;
        scala) echo "scala" ;;
        pl) echo "perl" ;;
        sh) echo "bash" ;;
        sql) echo "sql" ;;
        json) echo "javascript" ;;
        *) echo "text" ;;
    esac
}

# Function to generate the find command for a single directory
generate_find_command() {
    local dir="$1"
    local cmd="find \"$dir\" \( -type d \( -name node_modules -o -name .next \) -prune \) -o \( -type f \("
    for i in "${!EXTENSIONS[@]}"; do
        if [ $i -ne 0 ]; then
            cmd+=" -o"
        fi
        cmd+=" -name \"*.${EXTENSIONS[$i]}\""
    done
    cmd+=" \) -print \)"
    echo "$cmd"
}

# Clear the temporary PS file if it exists
> "$TEMP_PS"

# Process each directory
for dir in "$@"; do
    if [ ! -d "$dir" ]; then
        echo "Warning: $dir is not a valid directory. Skipping."
        continue
    fi

    echo "Processing directory: $dir"

    # Generate and execute the find command
    eval "$(generate_find_command "$dir")" | while read -r file; do
        echo "Converting $file"
        # Get the file extension
        ext="${file##*.}"
        # Get the appropriate language for enscript
        lang=$(get_language "$ext")
        # Use the appropriate syntax highlighting based on the file extension
        enscript -p - --highlight="$lang" --color=1 -fCourier8 --header="$file|Page \$% of \$=" "$file" >> "$TE
MP_PS"
    done
done
```

```
# Check if any files were processed
if [ ! -s "$TEMP_PS" ]; then
    echo "Error: No files were found or processed. The output PDF will not be created."
    rm "$TEMP_PS"
    exit 1
fi

# Convert PostScript to PDF
ps2pdf "$TEMP_PS" "$OUTPUT"

# Remove temporary PostScript file
rm "$TEMP_PS"

echo "Conversion complete. Output saved as $OUTPUT"
```

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

// NOTE: This file should not be edited
// see https://nextjs.org/docs/basic-features/typescript for more information.
```

```
import type { Config } from "tailwindcss";

const config = {
  darkMode: ["class"],
  content: [
    "./pages/**/*.{ts,tsx}",
    "./components/**/*.{ts,tsx}",
    "./app/**/*.{ts,tsx}",
    "./src/**/*.{ts,tsx}",
  ],
  prefix: "",
  theme: {
    container: {
      center: true,
      padding: "2rem",
      screens: {
        "2xl": "1400px",
      },
    },
    extend: {
      colors: {
        border: "hsl(var(--border))",
        input: "hsl(var(--input))",
        ring: "hsl(var(--ring))",
        background: "hsl(var(--background))",
        foreground: "hsl(var(--foreground))",
        primary: {
          DEFAULT: "hsl(var(--primary))",
          foreground: "hsl(var(--primary-foreground))",
        },
        secondary: {
          DEFAULT: "hsl(var(--secondary))",
          foreground: "hsl(var(--secondary-foreground))",
        },
        destructive: {
          DEFAULT: "hsl(var(--destructive))",
          foreground: "hsl(var(--destructive-foreground))",
        },
        muted: {
          DEFAULT: "hsl(var(--muted))",
          foreground: "hsl(var(--muted-foreground))",
        },
        accent: {
          DEFAULT: "hsl(var(--accent))",
          foreground: "hsl(var(--accent-foreground))",
        },
        popover: {
          DEFAULT: "hsl(var(--popover))",
          foreground: "hsl(var(--popover-foreground))",
        },
        card: {
          DEFAULT: "hsl(var(--card))",
          foreground: "hsl(var(--card-foreground))",
        },
      },
      borderRadius: {
        lg: "var(--radius)",
        md: "calc(var(--radius) - 2px)",
        sm: "calc(var(--radius) - 4px)",
      },
      keyframes: {
        "accordion-down": {
          from: { height: "0" },
          to: { height: "var(--radix-accordion-content-height)" },
        },
        "accordion-up": {
          from: { height: "var(--radix-accordion-content-height)" },
          to: { height: "0" },
        },
        fadeIn: {
          "0%": { opacity: "0" },
          "100%": { opacity: "1" },
        },
        fadeInUp: {
          "0%": { opacity: "0", transform: "translateY(8px)" },
          "100%": { opacity: "1", transform: "translateY(0)" },
        },
        shimmer: {
          "100%": { transform: "translateX(100%)" },
        },
      },
      animation: {
        "accordion-down": "accordion-down 0.2s ease-out",
        "accordion-up": "accordion-up 0.2s ease-out",
        "fade-in": "fadeIn 0.5s ease-out forwards",
        "fade-in-up": "fadeInUp 0.5s ease-out forwards",
        "fade-in-up-fast": "fadeInUp 0.3s ease-out forwards",
        "fade-in-up-slow":
          "fadeInUp 0.8s cubic-bezier(0.4, 0, 0.2, 1) forwards",
        shimmer: "shimmer 2s linear infinite",
      },
    },
  },
  plugins: [require("tailwindcss-animate")],
} satisfies Config;

export default config;
```

```
// themes.ts

export const themes = {
  neutral: {
    light: {
      background: "0 0% 100%",
      foreground: "0 0% 3.9%",
      card: "0 0% 100%",
      "card-foreground": "0 0% 3.9%",
      popover: "0 0% 100%",
      "popover-foreground": "0 0% 3.9%",
      primary: "0 0% 9%",
      "primary-foreground": "0 0% 98%",
      secondary: "0 0% 96.1%",
      "secondary-foreground": "0 0% 9%",
      muted: "0 0% 96.1%",
      "muted-foreground": "0 0% 45.1%",
      accent: "0 0% 96.1%",
      "accent-foreground": "0 0% 9%",
      destructive: "0 84.2% 60.2%",
      "destructive-foreground": "0 0% 98%",
      border: "0 0% 89.8%",
      input: "0 0% 89.8%",
      ring: "0 0% 3.9%",
      radius: "0.75rem",
      "chart-1": "12 76% 61%",
      "chart-2": "173 58% 39%",
      "chart-3": "197 37% 24%",
      "chart-4": "43 74% 66%",
      "chart-5": "27 87% 67%",
    },
    dark: {
      background: "0 0% 3.9%",
      foreground: "0 0% 98%",
      card: "0 0% 3.9%",
      "card-foreground": "0 0% 98%",
      popover: "0 0% 3.9%",
      "popover-foreground": "0 0% 98%",
      primary: "0 0% 98%",
      "primary-foreground": "0 0% 9%",
      secondary: "0 0% 14.9%",
      "secondary-foreground": "0 0% 98%",
      muted: "0 0% 14.9%",
      "muted-foreground": "0 0% 63.9%",
      accent: "0 0% 14.9%",
      "accent-foreground": "0 0% 98%",
      destructive: "0 62.8% 30.6%",
      "destructive-foreground": "0 0% 98%",
      border: "0 0% 14.9%",
      input: "0 0% 14.9%",
      ring: "0 0% 83.1%",
      radius: "0.75rem",
      "chart-1": "220 70% 50%",
      "chart-2": "160 60% 45%",
      "chart-3": "30 80% 55%",
      "chart-4": "280 65% 60%",
      "chart-5": "340 75% 55%",
    },
  },
  red: {
    light: {
      background: "0 0% 100%",
      foreground: "0 0% 3.9%",
      card: "0 0% 100%",
      "card-foreground": "0 0% 3.9%",
      popover: "0 0% 100%",
      "popover-foreground": "0 0% 3.9%",
      primary: "0 72.2% 50.6%",
      "primary-foreground": "0 85.7% 97.3%",
      secondary: "0 0% 96.1%",
      "secondary-foreground": "0 0% 9%",
      muted: "0 0% 96.1%",
      "muted-foreground": "0 0% 45.1%",
      accent: "0 0% 96.1%",
      "accent-foreground": "0 0% 9%",
      destructive: "0 84.2% 60.2%",
      "destructive-foreground": "0 0% 98%",
      border: "0 0% 89.8%",
      input: "0 0% 89.8%",
      ring: "0 72.2% 50.6%",
      radius: "0.75rem",
      "chart-1": "12 76% 61%",
      "chart-2": "173 58% 39%",
      "chart-3": "197 37% 24%",
      "chart-4": "43 74% 66%",
      "chart-5": "27 87% 67%",
    },
    dark: {
      background: "0 0% 3.9%",
      foreground: "0 0% 98%",
      card: "0 0% 3.9%",
      "card-foreground": "0 0% 98%",
      popover: "0 0% 3.9%",
      "popover-foreground": "0 0% 98%",
      primary: "0 72.2% 50.6%",
      "primary-foreground": "0 85.7% 97.3%",
      secondary: "0 0% 14.9%",
      "secondary-foreground": "0 0% 98%",
      muted: "0 0% 14.9%",
      "muted-foreground": "0 0% 63.9%",
      accent: "0 0% 14.9%",
      "accent-foreground": "0 0% 98%",
      destructive: "0 62.8% 30.6%",
      "destructive-foreground": "0 0% 98%",
      border: "0 0% 14.9%",
      input: "0 0% 14.9%",
      ring: "0 72.2% 50.6%",
      radius: "0.75rem",
      "chart-1": "220 70% 50%",
      "chart-2": "160 60% 45%",
      "chart-3": "30 80% 55%",
      "chart-4": "280 65% 60%",
      "chart-5": "340 75% 55%",
    },
  },
  violet: {
    light: {
      background: "0 0% 100%",
      foreground: "224 71.4% 4.1%",
      card: "0 0% 100%",
      "card-foreground": "224 71.4% 4.1%",
      popover: "0 0% 100%",
      "popover-foreground": "224 71.4% 4.1%",
      primary: "262.1 83.3% 57.8%",
      "primary-foreground": "210 20% 98%",
      secondary: "220 14.3% 95.9%",
      "secondary-foreground": "220.9 39.3% 11%",
      muted: "220 14.3% 95.9%",
      "muted-foreground": "220 8.9% 46.1%",
      accent: "220 14.3% 95.9%",
      "accent-foreground": "220.9 39.3% 11%",
      destructive: "0 84.2% 60.2%",
      "destructive-foreground": "210 20% 98%",
      border: "220 13% 91%",
      input: "220 13% 91%",
      ring: "262.1 83.3% 57.8%",
      radius: "0.75rem",
      "chart-1": "12 76% 61%",
      "chart-2": "173 58% 39%",
      "chart-3": "197 37% 24%",
      "chart-4": "43 74% 66%",
      "chart-5": "27 87% 67%",
    },
    dark: {
      background: "224 71.4% 4.1%",
      foreground: "210 20% 98%",
      card: "224 71.4% 4.1%",
      "card-foreground": "210 20% 98%",
      popover: "224 71.4% 4.1%",
      "popover-foreground": "210 20% 98%",
      primary: "263.4 70% 50.4%",
      "primary-foreground": "210 20% 98%",
      secondary: "215 27.9% 16.9%",
      "secondary-foreground": "210 20% 98%",
      muted: "215 27.9% 16.9%",
      "muted-foreground": "217.9 10.6% 64.9%",
      accent: "215 27.9% 16.9%",
      "accent-foreground": "210 20% 98%",
      destructive: "0 62.8% 30.6%",
      "destructive-foreground": "210 20% 98%",
      border: "215 27.9% 16.9%",
      input: "215 27.9% 16.9%",
      ring: "263.4 70% 50.4%",
      radius: "0.75rem",
      "chart-1": "220 70% 50%",
      "chart-2": "160 60% 45%",
      "chart-3": "30 80% 55%",
      "chart-4": "280 65% 60%",
      "chart-5": "340 75% 55%",
    },
  },
  blue: {
    light: {
      background: "0 0% 100%",
      foreground: "222.2 84% 4.9%",
      card: "0 0% 100%",
      "card-foreground": "222.2 84% 4.9%",
      popover: "0 0% 100%",
      "popover-foreground": "222.2 84% 4.9%",
      primary: "221.2 83.2% 53.3%",
      "primary-foreground": "210 40% 98%",
      secondary: "210 40% 96.1%",
      "secondary-foreground": "222.2 47.4% 11.2%",
      muted: "210 40% 96.1%",
      "muted-foreground": "215.4 16.3% 46.9%",
      accent: "210 40% 96.1%",
      "accent-foreground": "222.2 47.4% 11.2%",
      destructive: "0 84.2% 60.2%",
      "destructive-foreground": "210 40% 98%",
      border: "214.3 31.8% 91.4%",
      input: "214.3 31.8% 91.4%",
      ring: "221.2 83.2% 53.3%",
      radius: "0.75rem",
      "chart-1": "12 76% 61%",
      "chart-2": "173 58% 39%",
      "chart-3": "197 37% 24%",
      "chart-4": "43 74% 66%",
      "chart-5": "27 87% 67%",
    },
    dark: {
      background: "222.2 84% 4.9%",
      foreground: "210 40% 98%",
      card: "222.2 84% 4.9%",
      "card-foreground": "210 40% 98%",
      popover: "222.2 84% 4.9%",
      "popover-foreground": "210 40% 98%",
      primary: "217.2 91.2% 59.8%",
      "primary-foreground": "222.2 47.4% 11.2%",
      secondary: "217.2 32.6% 17.5%",
      "secondary-foreground": "210 40% 98%",
      muted: "217.2 32.6% 17.5%",
      "muted-foreground": "215 20.2% 65.1%",
      accent: "217.2 32.6% 17.5%",
      "accent-foreground": "210 40% 98%",
      destructive: "0 62.8% 30.6%",
      "destructive-foreground": "210 40% 98%",
      border: "217.2 32.6% 17.5%",
      input: "217.2 32.6% 17.5%",
      ring: "224.3 76.3% 48%",
      "chart-1": "220 70% 50%",
      "chart-2": "160 60% 45%",
      "chart-3": "30 80% 55%",
      "chart-4": "280 65% 60%",
      "chart-5": "340 75% 55%",
    },
  },
  tangerine: {
    light: {
      background: "0 0% 100%",
      foreground: "20 14.3% 4.1%",
      card: "0 0% 100%",
      "card-foreground": "20 14.3% 4.1%",
      popover: "0 0% 100%",
      "popover-foreground": "20 14.3% 4.1%",
      primary: "24.6 95% 53.1%",
      "primary-foreground": "60 9.1% 97.8%",
      secondary: "60 4.8% 95.9%",
      "secondary-foreground": "24 9.8% 10%",
      muted: "60 4.8% 95.9%",
      "muted-foreground": "25 5.3% 44.7%",
      accent: "60 4.8% 95.9%",
      "accent-foreground": "24 9.8% 10%",
      destructive: "0 84.2% 60.2%",
      "destructive-foreground": "60 9.1% 97.8%",
      border: "20 5.9% 90%",
      input: "20 5.9% 90%",
      ring: "24.6 95% 53.1%",
      radius: "0.75rem",
      "chart-1": "12 76% 61%",
      "chart-2": "173 58% 39%",
      "chart-3": "197 37% 24%",
      "chart-4": "43 74% 66%",
      "chart-5": "27 87% 67%",
    },
    dark: {
      background: "20 14.3% 4.1%",
      foreground: "60 9.1% 97.8%",
      card: "20 14.3% 4.1%",
      "card-foreground": "60 9.1% 97.8%",
      popover: "20 14.3% 4.1%",
      "popover-foreground": "60 9.1% 97.8%",
      primary: "20.5 90.2% 48.2%",
      "primary-foreground": "60 9.1% 97.8%",
      secondary: "12 6.5% 15.1%",
      "secondary-foreground": "60 9.1% 97.8%",
      muted: "12 6.5% 15.1%",
      "muted-foreground": "24 5.4% 63.9%",
      accent: "12 6.5% 15.1%",
      "accent-foreground": "60 9.1% 97.8%",
      destructive: "0 72.2% 50.6%",
      "destructive-foreground": "60 9.1% 97.8%",
      border: "12 6.5% 15.1%",
      input: "12 6.5% 15.1%",
      ring: "20.5 90.2% 48.2%",
      "chart-1": "220 70% 50%",
      "chart-2": "160 60% 45%",
      "chart-3": "30 80% 55%",
      "chart-4": "280 65% 60%",
      "chart-5": "340 75% 55%",
    },
  },
  emerald: {
    light: {
      background: "0 0% 100%",
      foreground: "240 10% 3.9%",
      card: "0 0% 100%",
      "card-foreground": "240 10% 3.9%",
      popover: "0 0% 100%",
      "popover-foreground": "240 10% 3.9%",
      primary: "142.1 76.2% 36.3%",
      "primary-foreground": "355.7 100% 97.3%",
      secondary: "240 4.8% 95.9%",
      "secondary-foreground": "240 5.9% 10%",
      muted: "240 4.8% 95.9%",
      "muted-foreground": "240 3.8% 46.1%",
      accent: "240 4.8% 95.9%",
      "accent-foreground": "240 5.9% 10%",
      destructive: "0 84.2% 60.2%",
      "destructive-foreground": "0 0% 98%",
      border: "240 5.9% 90%",
      input: "240 5.9% 90%",
      ring: "142.1 76.2% 36.3%",
      radius: "0.75rem",
      "chart-1": "12 76% 61%",
      "chart-2": "173 58% 39%",
      "chart-3": "197 37% 24%",
      "chart-4": "43 74% 66%",
      "chart-5": "27 87% 67%",
    },
    dark: {
      background: "20 14.3% 4.1%",
      foreground: "0 0% 95%",
      card: "24 9.8% 10%",
      "card-foreground": "0 0% 95%",
      popover: "0 0% 9%",
      "popover-foreground": "0 0% 95%",
      primary: "142.1 70.6% 45.3%",
      "primary-foreground": "144.9 80.4% 10%",
      secondary: "240 3.7% 15.9%",
      "secondary-foreground": "0 0% 98%",
      muted: "0 0% 15%",
      "muted-foreground": "240 5% 64.9%",
      accent: "12 6.5% 15.1%",
      "accent-foreground": "0 0% 98%",
      destructive: "0 62.8% 30.6%",
      "destructive-foreground": "0 85.7% 97.3%",
      border: "240 3.7% 15.9%",
      input: "240 3.7% 15.9%",
      ring: "142.4 71.8% 29.2%",
      "chart-1": "220 70% 50%",
      "chart-2": "160 60% 45%",
      "chart-3": "30 80% 55%",
      "chart-4": "280 65% 60%",
      "chart-5": "340 75% 55%",
    },
  },
  amber: {
    light: {
      background: "0 0% 100%",
      foreground: "20 14.3% 4.1%",
      card: "0 0% 100%",
      "card-foreground": "20 14.3% 4.1%",
      popover: "0 0% 100%",
      "popover-foreground": "20 14.3% 4.1%",
      primary: "47.9 95.8% 53.1%",
      "primary-foreground": "26 83.3% 14.1%",
      secondary: "60 4.8% 95.9%",
      "secondary-foreground": "24 9.8% 10%",
      muted: "60 4.8% 95.9%",
      "muted-foreground": "25 5.3% 44.7%",
      accent: "60 4.8% 95.9%",
      "accent-foreground": "24 9.8% 10%",
      destructive: "0 84.2% 60.2%",
      "destructive-foreground": "60 9.1% 97.8%",
      border: "20 5.9% 90%",
      input: "20 5.9% 90%",
      ring: "20 14.3% 4.1%",
      radius: "0.75rem",
      "chart-1": "12 76% 61%",
      "chart-2": "173 58% 39%",
      "chart-3": "197 37% 24%",
      "chart-4": "43 74% 66%",
      "chart-5": "27 87% 67%",
    },
    dark: {
      background: "20 14.3% 4.1%",
      foreground: "60 9.1% 97.8%",
      card: "20 14.3% 4.1%",
      "card-foreground": "60 9.1% 97.8%",
      popover: "20 14.3% 4.1%",
      "popover-foreground": "60 9.1% 97.8%",
      primary: "47.9 95.8% 53.1%",
      "primary-foreground": "26 83.3% 14.1%",
      secondary: "12 6.5% 15.1%",
      "secondary-foreground": "60 9.1% 97.8%",
      muted: "12 6.5% 15.1%",
      "muted-foreground": "24 5.4% 63.9%",
      accent: "12 6.5% 15.1%",
      "accent-foreground": "60 9.1% 97.8%",
      destructive: "0 62.8% 30.6%",
      "destructive-foreground": "60 9.1% 97.8%",
      border: "12 6.5% 15.1%",
      input: "12 6.5% 15.1%",
      ring: "35.5 91.7% 32.9%",
      "chart-1": "220 70% 50%",
      "chart-2": "160 60% 45%",
      "chart-3": "30 80% 55%",
      "chart-4": "280 65% 60%",
      "chart-5": "340 75% 55%",
    },
  },
};
```

## /Users/abdulazizdot/Desktop/customer-support-agent/lib/utils.ts

```
import { type ClassValue, clsx } from "clsx";
import { twMerge } from "tailwind-merge";

export function cn(...inputs: ClassValue[]) {
  return twMerge(clsx(inputs));
}
```

```
type Config = {
  includeLeftSidebar: boolean;
  includeRightSidebar: boolean;
};

const config: Config = {
  includeLeftSidebar: process.env.NEXT_PUBLIC_INCLUDE_LEFT_SIDEBAR === "true",
  includeRightSidebar: process.env.NEXT_PUBLIC_INCLUDE_RIGHT_SIDEBAR === "true",
};

export default config;
```