

CLAUDE.md

This file provides guidance to Claude Code (claude.ai/code) when working with code in this repository.

Project Overview

Real-time Indian stock market tracking application with watchlist management, price alerts, and market scanning.

Project Structure

This is a monorepo with two main parts:

- `client/` - React frontend (Vite + TypeScript)
- `server/` - Express backend (runs locally with nodemon, deployed with PM2)

Commands

Development (Run Both Concurrently)

```
# Terminal 1: Start backend server on port 3001
cd server && npm run dev

# Terminal 2: Start frontend dev server with Vite
cd client && npm run dev
```

Production

```
# Root: Install all dependencies and build client
npm run build

# Root: Start production server (uses PM2)
npm start
```

Client Commands

```
cd client
npm run dev           # Vite dev server
npm run build        # TypeScript check + production build
npm run lint         # ESLint
npm run preview      # Preview production build
```

Server Commands

```
cd server
npm run dev          # Nodemon development server
npm start            # PM2 production runtime
npm run logs         # View PM2 logs
```

Architecture

Tech Stack

Frontend:

- React 19 with TypeScript
- Vite (build tool)
- Tailwind CSS 4 with shadcn/ui components (New York style)
- Zustand for state management (with localStorage persistence)
- React Query (@tanstack/react-query) for data fetching and caching
- Recharts for stock charts

- React Router for routing
- Axios for API calls

Backend:

- Express 5 server
- MongoDB with Mongoose (for user auth and wishlist persistence)
- JWT authentication with cookies
- PM2 for process management in production
- 60-second cache layer for stock quotes and charts

Key Directories

```
client/src/
├── components/           # React components
│   ├── ui/              # shadcn/ui base components (Button, Card, Dialog, etc.)
│   ├── StockCard.tsx   # Individual stock display card
│   ├── StockChart.tsx  # Recharts-based price chart
│   ├── MarketScanner.tsx # Real-time market scanning with alerts
│   ├── SearchDialog.tsx # Stock symbol search
│   └── AlertsPanel.tsx  # Price alert notifications
├── hooks/
│   ├── useStocks.ts     # React Query hooks for all data fetching
│   └── useDebounce.ts   # Debounce utility hook
├── services/
│   └── stockApi.ts      # Axios API client with endpoints
├── store/
│   └── useStockStore.ts # Zustand global state (watchlist, stocks, alerts)
├── types/stock.ts      # TypeScript interfaces
├── lib/
│   ├── constants.ts    # Config (polling intervals, thresholds, formatters)
│   └── timeUtils.ts     # Market hours detection
└── contexts/
    └── AuthContext.tsx  # Authentication context

server/
├── server.js           # Main Express app with all API endpoints
├── authMiddleware.js   # JWT authentication middleware
├── db.js               # MongoDB connection
└── models/
    ├── User.js        # User model (email, password)
    └── WishlistItem.js # Watchlist persistence model
```

Configuration (client/src/lib/constants.ts)

- `POLLING_INTERVAL` : 20 seconds (watchlist auto-refresh during market hours)
- `MARKET_SCANNER_INTERVAL` : 60 seconds
- `ALERT_THRESHOLD` : ₹10 price change from previous close
- `CHART_INTERVALS` : 1d, 5d, 1mo, 6mo, ytd, 1y, 5y
- `DEFAULT_STOCKS` : RELIANCE.NS, TCS.NS, INFY.NS, HDFCBANK.NS, ICICIBANK.NS

Stock Symbol Format

Indian stocks require exchange suffix:

- `.NS` = National Stock Exchange (NSE)
- `.BO` = Bombay Stock Exchange (BSE)

Examples: `RELIANCE.NS`, `TCS.NS`, `TATAMOTORS.BO`