{

"name": "TradePro",

"version": "1.0.0",

"description": "Ultimate Day Trading Platform with Smart Insights",

"repository": {

"type": "git",

"url": "https://github.com/yourusername/tradepro"

},

"license": "MIT",

"author": "Your Name",

"features": {

"tradingInsights": {

"description": "Contextual help bubbles providing trading tips based on user experience level",

"components": [

{

"name": "TradingInsightBubble",

"description": "Tooltip component that displays insights contextually",

"props": {

"contextId": "Identifies which context to show insights for",

"position": "Tooltip position (top/bottom/left/right)",

"alwaysShow": "Whether to display as tooltip or permanent card",

"insightId": "Optional specific insight to display"

},

"path": "client/src/components/TradingInsightBubble.tsx"

},

{

"name": "TradingInsightsManager",

"description": "Context provider for managing insights across the app",

"functionality": [

"Track dismissed insights in localStorage",

"Filter insights by user skill level",

"Provide access to insights via context hooks"

],

"path": "client/src/components/TradingInsightsManager.tsx"

},

{

"name": "TradingInsightsSettings",

"description": "Settings interface for configuring trading insights",

"settings": [

"Toggle insights on/off",

"Set skill level (beginner/intermediate/advanced)",

"Set insight frequency (high/medium/low/off)",

"Toggle hover mode",

"Reset dismissed insights"

],

"path": "client/src/components/TradingInsightsSettings.tsx"

}

],

"dataModel": {

"sources": "client/src/data/tradingInsights.ts",

"categories": [

"order-entry",

"chart-analysis",

"risk-management",

"position-sizing",

"pattern-recognition",

"market-psychology",

"strategy",

"general"

],

"skillLevels": [

"beginner",

"intermediate",

"advanced"

]

}

},

"core": {

"marketData": {

"description": "Real-time price updates via WebSockets",

"components": [

{

"name": "WebSocketProvider",

"path": "client/src/components/WebSocketProvider.tsx",

"functionality": "Manages WebSocket connections for real-time data"

},

{

"name": "MarketDataSlice",

"path": "client/src/store/slices/marketDataSlice.ts",

"functionality": "Redux state management for market data"

}

]

},

"orderEntry": {

"description": "Advanced order types with risk management",

"components": [

{

"name": "OrderEntryForm",

"path": "client/src/components/OrderEntry/index.tsx",

"functionality": "Form for creating and submitting orders"

},

{

"name": "OrdersSlice",

"path": "client/src/store/slices/ordersSlice.ts",

"functionality": "Redux state management for orders"

}

]

},

"portfolio": {

"description": "Position tracking and performance analysis",

"components": [

{

"name": "PortfolioOverview",

"path": "client/src/components/Portfolio/index.tsx",

"functionality": "Display of current positions and performance"

}

]

},

"watchlists": {

"description": "Custom symbol watchlists with sorting",

"components": [

{

"name": "WatchlistComponent",

"path": "client/src/components/Watchlist/index.tsx",

"functionality": "User-defined lists of stocks to monitor"

}

]

},

"charts": {

"description": "Technical analysis with multiple timeframes",

"components": [

{

"name": "StockChart",

"path": "client/src/components/Dashboard/StockChart.tsx",

"functionality": "Interactive price charts with timeframe selection"

},

{

"name": "TechnicalIndicators",

"path": "client/src/components/Charts/Indicators.tsx",

"functionality": "Technical analysis indicators (RSI, MACD, etc.)"

}

]

},

"journal": {

"description": "Trade journaling with performance metrics",

"components": [

{

"name": "JournalEntries",

"path": "client/src/components/Journal/index.tsx",

"functionality": "Log and review trading activity and notes"

}

]

},

"settings": {

"description": "User preferences and application configuration",

"components": [

{

"name": "SettingsPanel",

"path": "client/src/components/Settings/index.tsx",

"functionality": "Interface for all user settings including trading insights"

},

{

"name": "UserPreferencesSlice",

"path": "client/src/store/slices/userPreferencesSlice.ts",

"functionality": "Redux state management for user preferences"

}

]

}

}

},

"architecture": {

"frontend": {

"framework": "React with TypeScript",

"stateManagement": "Redux with Redux Toolkit",

"components": "Custom shadcn/ui components with Tailwind CSS",

"dataFetching": "TanStack Query (React Query)",

"routing": "wouter lightweight router",

"structure": {

"client/src": {

"components": "Reusable UI components",

"pages": "Main application pages/routes",

"store": "Redux store configuration and slices",

"lib": "Utility functions and API client",

"hooks": "Custom React hooks",

"data": "Static data and models"

}

}

},

"backend": {

"server": "Express.js with TypeScript",

"database": "PostgreSQL with Drizzle ORM",

"realtime": "WebSockets for live price updates",

"structure": {

"server": {

"index.ts": "Server entry point",

"routes.ts": "API route definitions",

"storage.ts": "Data access layer",

"db.ts": "Database connection",

"vite.ts": "Development server configuration"

},

"shared": {

"schema.ts": "Shared data models between client and server"

}

}

},

"crossPlatform": {

"desktop": {

"technology": "Electron",

"platforms": ["Windows", "macOS", "Linux"],

"structure": {

"electron": {

"main.ts": "Electron main process",

"preload.ts": "Preload scripts for renderer",

"update.ts": "Auto-update functionality"

}

}

},

"mobile": {

"technology": "Capacitor",

"platforms": ["iOS", "Android"],

"configuration": "capacitor.config.ts"

}

}

},

"database": {

"type": "PostgreSQL",

"orm": "Drizzle ORM",

"models": {

"users": "User accounts and authentication",

"userPreferences": "User-specific settings and preferences",

"instruments": "Financial instruments metadata",

"portfolios": "User portfolios",

"positions": "Current and historical positions",

"orders": "Trade orders",

"watchlists": "Custom symbol watchlists",

"watchlistItems": "Items within watchlists",

"journalEntries": "Trading journal entries",

"strategies": "Custom trading strategies"

},

"configuration": "drizzle.config.ts"

},

"deployment": {

"hosting": "Replit",

"database": "Neon PostgreSQL",

"environment": {

"variables": [

"DATABASE\_URL",

"PGHOST",

"PGUSER",

"PGPASSWORD",

"PGDATABASE",

"PGPORT"

],

"secrets": [

"OPENAI\_API\_KEY"

]

},

"ci": "GitHub Actions",

"scripts": {

"start": "npm run dev",

"build": "vite build",

"db:push": "drizzle-kit push:pg"

}

},

"dependencies": {

"frontend": [

"react",

"react-dom",

"redux",

"@reduxjs/toolkit",

"@tanstack/react-query",

"wouter",

"tailwindcss",

"recharts",

"zod",

"react-hook-form"

],

"backend": [

"express",

"drizzle-orm",

"@neondatabase/serverless",

"ws"

],

"crossPlatform": [

"@capacitor/core",

"@capacitor/ios",

"@capacitor/android",

"@capacitor/electron",

"electron"

]

},

"roadmap": {

"bugFixes": [

"Fix non-serializable dates in Redux store",

"Improve WebSocket reconnection logic",

"Handle edge cases in trading insights context"

],

"shortTerm": [

"Add context-specific insight triggers throughout the application",

"Implement an insight notification system for important alerts",

"Create a dedicated insights dashboard for all available trading tips"

],

"mediumTerm": [

"AI-powered trade analysis",

"Advanced alert system",

"Social trading integration"

],

"longTerm": [

"Backtesting engine",

"Enhanced trading insights with ML recommendations",

"Options analytics and strategy builder"

]

},

"documentation": {

"setup": "Follow instructions in README.md for local development setup",

"api": "API endpoints documented in server/routes.ts",

"architecture": "Architecture overview in docs/architecture.md"

},

"contributors": [

"Your Name <your.email@example.com>"

]

}