**STOCKWATCH FINANCIAL APPLICATION REPORT**

Introduction:

StockWatch is a comprehensive web-based stock portfolio management application designed to provide users with real-time market data, portfolio tracking, and trading capabilities. Built using Streamlit, the application serves as a unified platform for managing investment portfolios while delivering an intuitive and responsive user experience.

Usability Goals:

**Goal 1: Intuitive Navigation**

Objective: Enable users to navigate between different sections effortlessly without confusion.

Implementation:

* Implemented a persistent sidebar navigation with clearly labeled sections (Dashboard, Market, Trade, Transactions, News)
* Used consistent visual hierarchy and button styling
* Maintained active page state indication to help users understand their current location
* Applied the principle of progressive disclosure by organizing complex features into logical sections

Success Metrics: Users can navigate to any section within 2 clicks, with navigation patterns remaining consistent throughout the application.

**Goal 2: Real-Time Data Accessibility**

Objective: Provide immediate access to current market information without overwhelming the user.

Implementation:

* Integrated Finnhub API for live stock quotes and market data
* Implemented progress indicators during data loading to manage user expectations
* Used color-coded indicators (green/red) for immediate visual feedback on stock performance

Success Metrics: Market data refreshes within 3 seconds, with visual indicators showing data freshness and loading states.

**Goal 3: Error Prevention and Recovery**

Objective: Minimize user errors and provide clear recovery paths when errors occur.

Implementation:

* Added comprehensive input validation for trading forms (minimum values, required fields)
* Implemented position checking before allowing sell orders
* Provided fallback data sources and graceful degradation when APIs are unavailable
* Used confirmation dialogs for critical actions like trade execution

Success Metrics: Error rates reduced through validation, with clear error messages and recovery options provided when issues occur.

**Goal 4: Visual Clarity and Feedback**

Objective: Ensure users receive immediate feedback for their actions and can easily interpret data visualizations.

Implementation:

* Designed color-coded performance indicators throughout the interface
* Implemented real-time portfolio metrics with percentage changes
* Used interactive charts for portfolio performance visualization
* Provided geographic mapping for user location context

Success Metrics: Users can interpret key metrics within 5 seconds of viewing any page, with immediate visual feedback for all interactive elements.

**Goal 5: Efficient Task Completion**

Objective: Enable users to complete common tasks (checking portfolio, placing trades, viewing news) quickly and efficiently.

Implementation:

* Streamlined trading workflow with single-form submission
* Organized dashboard to show most critical information first
* Implemented search functionality for stock lookup
* Used tabular layouts for easy scanning of transaction history

Success Metrics: Core tasks (portfolio check, trade placement, news browsing) can be completed in under 30 seconds each.

Design Process:

The design process for this portfolio management application began with identifying core user needs and structuring the interface around five main sections with logical user flows from overview to detailed analysis. After all page designs were identified, I moved on to deciding on color scheme using green for positive and red for negative performance indicators against neutral gray backgrounds, consistent typography hierarchy with h2 and h4 headings, and adequate white space to prevent information overload.

API Integration:

The primary API integration utilized Finnhub Stock Market Data to provide real-time stock quotes, company profiles, and market news through specific endpoints including /quote, /stock/profile2, and /news. Key challenges included API rate limits requiring caching solutions, incomplete stock profile data necessitating fallback mechanisms, and network reliability issues addressed through timeout handling and retry logic.

The secondary API integration leveraged ipinfo.io's IP geolocation service to provide geographic context for users, processing latitude and longitude coordinates from location strings with Miami, Florida as the default fallback location. IP-based location services have an inherent accuracy limitation providing approximate rather than precise positioning.

Interactive Widgets:

**Navigation Sidebar**

Purpose: Primary navigation between application sections

Features:

* Persistent visibility for constant access
* Active state indication for current page
* Icon placeholders for visual enhancement
* Responsive button layout

**Stock Selection Dropdown**

Purpose: Symbol selection for trading and analysis

Features:

* Searchable dropdown with company names
* Format function displaying both symbol and company name
* Integration with real-time data fetching

**Trading Form**

Purpose: Order placement with comprehensive validation

Features:

* Radio buttons for buy/sell selection
* Number inputs with validation for quantity and price
* Order type selection (Market, Limit, Stop)
* Real-time total calculation
* Confirmation checkbox for error prevention

**Interactive Charts**

Purpose: Portfolio performance visualization

Features:

* Line chart for historical portfolio value
* Interactive hover information
* Date-based x-axis for temporal context
* Simulated historical data for demonstration

**Search Interface**

Purpose: Stock filtering and discovery

Features:

* Real-time search filtering
* Placeholder text for user guidance
* Integration with stock grid display

**Geographic Map**

Purpose: User location visualization

Features:

* Interactive Streamlit map component
* Automatic zoom level for optimal viewing
* Real-time coordinate plotting
* Fallback location handling

**Data Tables**

Purpose: Transaction history and portfolio display

Features:

* Sortable columns for data organization
* Pagination for large datasets
* Formatted currency display
* Real-time price updates

HCI Principles:

The application organizes content around five distinct sections (Dashboard, Market, Trading, Transactions and News) that align with natural user workflows in portfolio management. The Dashboard prioritizes critical information like portfolio value and daily changes at the top, followed by performance charts and geographic mapping, creating a logical information hierarchy.

Clear and consistent navigation is maintained through persistent navigation elements that allow users to move between sections at any time without losing their progress. The navigation structure follows standard web conventions with consistent placement and styling across all pages.

The application provides comprehensive immediate feedback through multiple mechanisms including progress bars during data loading operations, real-time validation on trading forms, and visual indicators for stock performance using green/red color coding. When users interact with the system, they receive confirmation messages for trades, error messages in plain language when issues occur, and status updates during batch API calls.

The design maintains visual appeal through a clean, minimalist approach using adequate white space, a consistent color scheme limited to essential indicators (green for gains, red for losses, neutral grays for text), and professional typography with clear heading hierarchies. The layout employs consistent margin and padding patterns, uniform button styling across all pages, and standardized metric formatting ($X,XXX.XX) to create a cohesive visual experience.

Conclusion:

The development of StockWatch provided valuable insights into creating user-centered applications. One of the most challenging parts was finding an API that would not limit the amount of data calls. The first option (MarketStack) had a 100 call limit and within the first 30 minutes of coding, I had already consumed almost half of that. Overall, I believe the application is straightforward for financial industry users as it follows standard workflows in securities trading and portfolio management.

For future improvements, I would work a little more on the styling of the dashboard, perhaps with more data points to explore. I would also include some icons on the menu, some advanced charting capabilities, portfolio analytics, database integration and an authentication system for security purposes.