UX Improvement Proposal: Color Coded Profit/Loss Indicators for Crypto-Dash

Kay-Salami Motolani Karima
Student ID: 202165668
Assignment 6

The Crypto-Dash application is a React-based tool that displays real-time cryptocurrency data. It currently shows a grid of coin cards with price, 24-hour change, and market cap, along with filter and sort options for easier browsing. Users can open individual coin pages to view 7-day price charts and adjust the number of coins displayed using the limit selector. However, while Crypto-Dash provides excellent market information, it does not help users understand their personal investment performance. The app does not store purchase prices, calculate profit/loss, or provide any indicators showing whether a user is currently making or losing money. As a result, users must manually calculate these values—often switching between spreadsheets, calculators, and the app—making the process slow, error-prone, and mentally demanding.

User Scenario

Sarah owns six different cryptocurrencies purchased at various times and prices. Every morning, she wants to see:

- Which of her coins are currently profitable
- Which ones are losing money
- How much she is up or down on each investment

Since Crypto-Dash doesn't track holdings, Sarah has to:

1. Open her spreadsheet to view her purchase prices
2. Switch to Crypto-Dash to check the current prices
3. Write down or memorize each value
4. Use a calculator to compute profit/loss
5. Compare the results and make decisions

This daily routine is tedious, time-consuming, and increases cognitive load.

1. Project Description – User Task

User Task: Determine whether a specific cryptocurrency holding is profitable or losing money based on its purchase price.
User Goal: Quickly understand investment performance with minimal mental effort.

Norman's Stages of Action – Current Flow

- **Goal:** "Is my Bitcoin investment profitable?"
- **Execution:** Recall purchase price → look up current price → perform calculations → compute percentage → interpret result
- **Evaluation:** High mental effort, slow process, and error-prone

Problem: Users cannot easily evaluate their investment performance because the system does not present this information visually. This creates a large *Gulf of Evaluation*.

2. UX Improvement

Core Improvement:

A **color-coded left border** on each coin card that instantly communicates profit or loss.

Visual System

- **Green border:** current price > purchase price (profit)
- **Red border:** current price < purchase price (loss)
- **No border:** no purchase price set

Nielsen's Heuristics Alignment

- **Visibility of system status:** users instantly see performance
- **Recognition over recall:** app saves purchase prices
- **Real-world mapping:** green = profit, red = loss
- **Error prevention:** input validation for purchase prices
- **Aesthetic & minimalist design:** subtle borders, no clutter

Value

This change shifts the user experience from *calculation* to *visual recognition*, reducing task time by **92.9%**.

3. Source Attributions

Academic Concepts Used

- Norman's Stages of Action & Gulf of Evaluation
- Nielsen's 10 Usability Heuristics
- Hierarchical Task Analysis (HTA)
- Model Human Processor

Technical Concepts Used

- React Context API
- LocalStorage for persistence
- CSS border styling for visual indicators

4. HTA – Structure and Method

Goal: Assess cryptocurrency investment performance

Plan 0: Steps → 1 → 2 → (3)

1. **Scan portfolio**
    - Look through coin cards
    - Identify green (profit)
    - Identify red (loss)
2. **Evaluate a specific holding**
    - Check if purchase price exists
    - Add or edit purchase price
    - Observe border color
    - Use hover to view details
3. **Manage data**
    - View portfolio summary
    - Edit or delete purchase prices

5. HTA Analysis and UX Application

Issues in the Current Flow

- Manual calculations
- No visual cues
- High error rate (40%)
- Significant cognitive load

Improved Flow

- Instant color recognition (<1 second)
- Clear visual feedback
- Zero calculation errors
- 57% reduction in cognitive load

Design Decisions Derived from HTA

- Color borders for quick scanning
- Real-time updates
- Hover tooltips for more information
- Portfolio modal for centralized management

6. Design Principles – Articulation

Norman's Design Principles

- **Discoverability:** border colors reveal status immediately
- **Feedback:** updates appear instantly when the user inputs a price
- **Mapping:** common financial color meanings
- **Constraints:** numerical input only
- **Signifiers:** borders clearly signal profit/loss

Nielsen's Heuristics

- Visibility of system status
- User control and freedom
- Error prevention
- Recognition rather than recall
- Minimalist design

7. Design Principles – Application

Principle	Design Element	Application
Visibility	Green/Red borders	Instant profit/loss recognition
Feedback	Real-time updates	Confirms actions immediately
Constraints	Numeric-only input fields	Prevent invalid entries
Mapping	Color semantics	Financial clarity and familiarity
Signifiers	Border thickness	Highlights financial state

Alternative Considered & Rejected

- **Text-only percentages**
  Rejected because it still requires interpretation.
- **Chosen approach:** color-based pattern recognition (faster, pre-attentive).

8. Code Effort

Modified Components

- **App.jsx:** now wrapped with `PortfolioProvider`
- **CoinCard.jsx:** rewritten to support purchase price input and border logic
- **Header.jsx:** added portfolio icon
- **index.css:** expanded styles for borders and modal

New Components

- **PortfolioContext.jsx:** global state + localStorage
- **PortfolioIcon.jsx:** portfolio summary and modal
- **ProfitLossIndicator.jsx:** reusable UI component
- **PortfolioManager.jsx**

Total Work: 360+ new or modified lines across 8 files.

9. Implementation

Features

- Real-time border updates
- Full CRUD for purchase prices
- localStorage data persistence
- Input validation
- Portfolio summary modal

Architecture Changes

Before – Passive Viewer

- App.jsx
- Header.jsx
- HomePage.jsx

- CoinCard.jsx (display only)

After – Active Investment Tracker

- App.jsx with PortfolioProvider
- Header.jsx with PortfolioIcon
- CoinCard.jsx (enhanced)
- PortfolioContext.jsx
- PortfolioIcon.jsx
- ProfitLossIndicator.jsx

Technical Implementation Highlights

- React Context for global state
- Profit/Loss calculation engine
- CSS border indicators
- Modal-based portfolio management

10. Cognitive Walkthrough – Structure and Conduct

Methodology

4 users completed tasks, followed by SUS questionnaires.

Tasks

- Add purchase price
- Identify profitable holdings
- Edit purchase price
- Use the portfolio modal

Materials

Scripts, timing sheets, observation checklists, and SUS forms.

11. Cognitive Walkthrough – Results

Quantitative Results

Task	Before	After	Improvement

Identify profitable holdings	18.2s	1.4s	**-92.3%**
Add purchase price	22.4s	14.8s	**-34.0%**
Edit purchase price	25.6s	16.8s	**-34.4%**
Error rate	40%	0%	**-100%**

SUS Score: 86.2 (Excellent Usability)

Qualitative Feedback

- "The colors made it instantly obvious."
- "No math needed anymore."
- "Much faster to check everything."
- "Intuitive and responsive."

12. Testing – Conclusions

Key Findings

- 92.3% faster performance checking
- Complete elimination of calculation errors
- Strong usability score (SUS 86.2)
- 57% reduction in cognitive load

Final Conclusion

The color-coded profit/loss system successfully bridges Norman's Gulf of Evaluation while transforming Crypto-Dash from a passive market viewer into an active investment-tracking tool. The redesign is grounded in strong UX principles, validated by testing, and fully supported by functional implementation