Final Project Report

Project Title:
MarketWatch – Real-Time Stock & Crypto Tracker

Name:
Pranay Gunvant Gedam

Internship Domain:
Web Development Internship

Name:
Pranay Gunvant Gedam

Organization:
Prism Studio

1. Introduction

In today's fast-paced financial world, investors and traders demand real-time insights into cryptocurrency and stock prices. MarketWatch is a web-based application that serves this purpose by providing live market data, price trends, and user watchlist functionality. It's a pure frontend project developed using HTML, Tailwind CSS, and JavaScript, with data fetched from the CoinGecko public API.

2. Objectives

• To design and build a responsive dashboard showing real-time crypto prices.

• To provide interactive features like search, dark mode, and coin-specific charts.

• To implement a localStorage-powered watchlist system.

• To allow users to view historical price trends with interactive graphs.

• To improve frontend skills by integrating APIs, chart libraries, and UI components.

3. Tools & Technologies Used

- HTML5 for markup and page structure

- Tailwind CSS for styling

- JavaScript (Vanilla) for logic

- Chart.js for rendering charts

- CoinGecko API for real-time data

- LocalStorage for theme/watchlist persistence

- GitHub Pages for hosting

4. Project Structure

```
marketwatch/
├── index.html        → Main Dashboard
├── watchlist.html    → User Watchlist
├── details.html      → Coin-specific charts
├── style.css         → Custom Tailwind styles
├── script.js         → App logic
├── assets/           → Screenshots and icons
└── README.md         → Overview file
```

5. Key Features

- Live Dashboard: Auto-refreshing crypto data every 60 seconds

- Search & Filter: Easily locate specific assets

- Watchlist: Persistent user selection stored in browser

- Historical Charts: View trends over 1D, 7D, 30D, 90D

- Responsive Layout: Fully usable on mobile and desktop

- Dark Mode: Toggle with persistent setting across sessions

6. Dark Mode Integration

Implemented using Tailwind's dark variant and `localStorage`. Ensures consistent user experience across pages.

7. Testing

Tested on various screen sizes and conditions including API failures, empty watchlist, and theme persistence.

8. Challenges Faced

- Managing asynchronous API calls for dynamic chart rendering

- Syncing watchlist and dark mode across multiple pages

- Building a clean responsive UI within limited time

9. Learning Outcomes

- Advanced JavaScript and API integration

- State management with localStorage

- Chart rendering using Chart.js

- Tailwind CSS layout utilities

- Real-world frontend project architecture

10. Future Enhancements

- Add stock data API alongside crypto

- User login with Firebase for cloud watchlist

- Alerts/notifications for price thresholds

- Global search with category filters

11. Conclusion

This internship project helped me build a fully functional, real-time market tracking application with modern frontend technologies. I enhanced my coding skills and gained experience in building user-focused web applications using open APIs and client-side logic.