

Sprint 4 - Comprehensive Report

Whole Agile Development Process Overview

The project followed Agile methodology across four sprints. Sprint 1 was focused on UI/UX design for mobile and web. Sprint 2 introduced core functionalities including login, stock trading, and dashboards. Sprint 3 emphasized advanced features and usability improvements, such as dark mode, responsive interfaces, and stock search. Sprint 4 addressed remaining bugs and quality assurance, including API issues, database fixes, and UI inconsistencies. Each sprint ended with refinement of backlog and delivery of tested features.

Changes Done During All Sprints

Summary of changes completed per sprint:

Sprint 1

- Designed mobile & web UI (dark mode, responsiveness)
- Created base wireframes for main components

Sprint 2

- Implemented login system with JWT
- Created dashboard and portfolio view
- Enabled stock buy/sell functionality

Sprint 3

- Refined frontend performance
- Improved DOM update handling
- Handled DTO mismatches and added charts

Sprint 4

- Resolved API, fetch, DB connection issues
- Added test coverage and CI setup
- Integrated bug fixes into main backlog

Integration of Changes Into Product and Sprint Backlog

Each sprint's changes were documented and used to refine the product backlog. The team reviewed tasks at the end of each sprint, updated priorities, and added new entries to capture newly discovered needs or errors. These updates are reflected in backlog revisions and mini-backlogs.

Sprint	Change Implemented	Backlog Task ID	Integration Method
2	Added stock trading UI	2.3	Added subtask for form validation
3	Identified POST bug	3.8	Created refinement issue in Sprint 4
3	React state issue broke DOM updates	3.7	Added a new issue to Sprint 4
4	MongoDB connect failure	4.1	Bugfix task added to backlog
4	Fetch error due to wrong endpoint	3.6	Task marked as critical and resolved

Mapping of Tasks and User Stories

Each user story was mapped to one or more development tasks in the sprint backlog. For instance, the story for stock trading was implemented via three distinct tasks: UI handling, API binding, and form validation. Mapping ensured complete coverage.

User Story Summary	Related Tasks	Sprint
As a user, I want to trade stocks easily	2.3, 3.8, err3-3.8	2, 3, 4
As a user, I want responsive UI on mobile	1.1, 1.2, 1.3	1
As a user, I want to track portfolio performance	2.4, 3.3, 3.4	2, 3
As a user, I want secure login	2.1	2
As a developer, I want stable API communication	3.6, 3.7, 3.9, 4.1, err1-err5	3, 4

Backlog Evolution Table

Story Number	First Backlog	Refined Backlog
2.3	Buy/sell stock via button	Added form validation + confirmation modal
3.6	Fetch API data	Fixed endpoint and async bug
3.9	DTO mismatch fix	Schema normalized to match frontend
4.1	DB error on connect	Whitelist IP + fix URI

Quality Assurance and Testing Report

- Unit tests were written for login, search, and portfolio logic.
- Integration tests were done to verify backend communication (API, DB).
- Manual testing included all interfaces and workflows on mobile and web.
- Test coverage reached an estimated 85%.

- No critical bugs were found after testing.
- CI/CD pipeline included automated checks using GitHub Actions.

User Story	Interfaces	Test ID	Status	Testing Result
As a user, I want to register and log into the app	Web: Login, Register Mobile: Sign In, Sign Up	T1	Done	Passed
As a user, I want to manage my stock portfolio	Web: Portfolio Mobile: Profile, Portfolio	T2	Done	Passed
As a user, I want to buy/sell stocks easily	Web: Trade Details Mobile: Premium, Browse	T3	Done	Passed
As a user, I want to view detailed stock info	Web: Stock Details Mobile: Charts	T4	Done	Passed
As a user, I want the interface to be responsive	All Mobile & Web interfaces	T5	Done	Passed
As a user, I want to use dark mode and readable fonts	All Mobile & Web interfaces	T6	Done	Passed
As a user, I want to purchase a premium version	Web: Premium Mobile: Premium Page	T7	Done	Passed
As a user, I want to update my personal information	Web: Personal Info Mobile: Profile Page	T8	Done	Passed
As a user, I want my data to sync properly	Web: Portfolio, Home Mobile: Home Page, Events	T9	Done	Passed
As a user, I want consistent behavior on all devices	All Mobile & Web interfaces	T10	Done	Passed
As a developer, I want APIs to fetch/post correctly	Backend, APIs	T11	Issue	Updated
As a developer, I want the app to load fast	All Mobile & Web interfaces	T12	Done	Passed

Final Product Backlog (All Tasks Completed)

Task ID	Description	Status
1.1	Mobile UI Design	✅ Done
2.3	Stock Trading	✅ Done
3.6	Fix API Fetch	✅ Done
3.8	Fix POST request	✅ Done
3.9	Fix DTO mismatch	✅ Done
4.1	Fix DB connection	✅ Done