SmartStock – Sprint 1 & Sprint 2 Burndown Report

Team: JUME (Max, Jay, Erfan, Usman)

Sprints Covered:

• **Sprint 1:** March 6 – March 20, 2025

• **Sprint 2:** March 20 – April 3, 2025

🟩 Sprint 1 – Burndown Summary

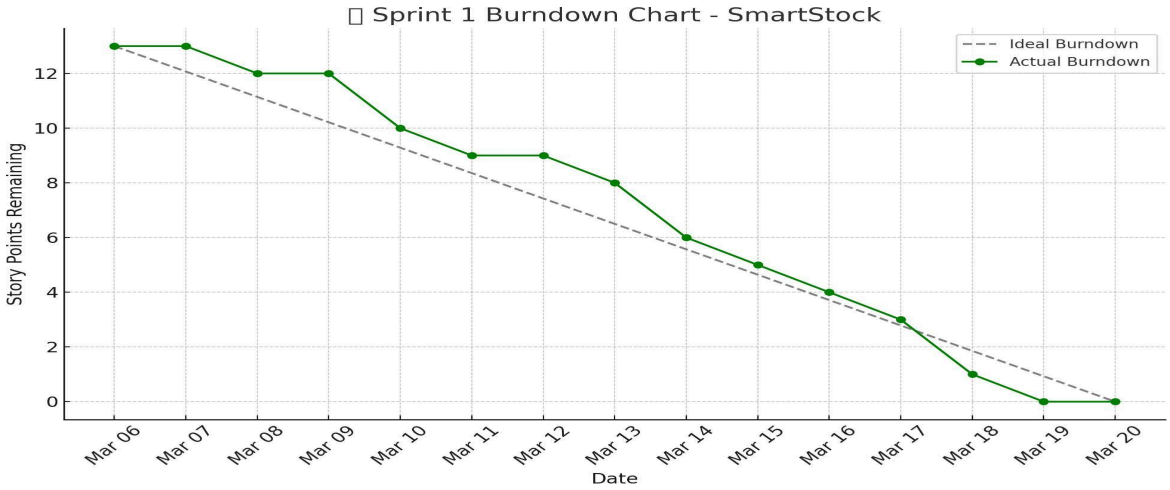
🎯 Sprint Goal

To lay the technical foundation for SmartStock by implementing authentication, setting up the system architecture, and building the core UI pages: Home, Orders, Products, and Profile.

Sprint	Planned Points	Actual Completed	Velocity
Sprint 1	13	13	✅ On Target

📌 Burndown Chart Commentary

- **Mar 6–10:** Focused on Supabase auth setup and initial routing.
- **Mid Sprint:** Some rework on UI components caused a temporary plateau.
- **Final Days:** Completed the last 5 story points after design finalization.
- **Outcome:** All goals completed, system foundation ready for Sprint 2.



✅ Sprint 1 Highlights

- Fully functional login system and routing with Supabase.
- Sidebar and page layout navigation established.
- Initial system architecture with Next.js and TailwindCSS ready.
- Created reusable UI components and began mock data seeding.

📈 Lessons Learned

- Early seeding and layout prep saved time.
- Task assignments were clear, but testing was minimal.
- Identified need for better error-handling logic and mock-data structure.

🟦 Sprint 2 – Burndown Summary

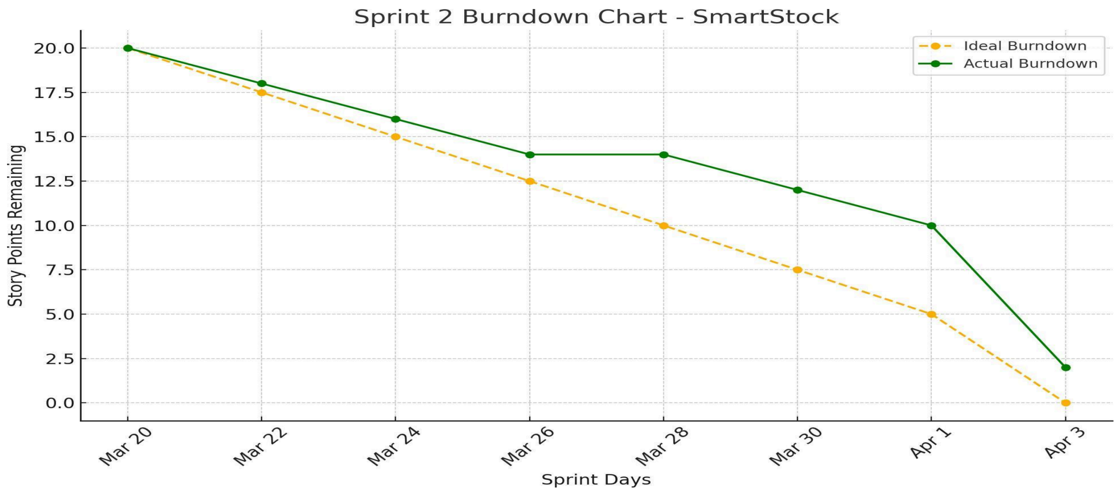
🎯 Sprint Goal

To complete the remaining SmartStock features, including Orders CRUD, Dashboard Analytics, Recent Activity logging, profile editing, and UI polish. Full Supabase integration across modules was also prioritized.

Sprint	Planned Points	Actual Completed	Velocity
Sprint 2	20	18	⚠ Slight Delay

📌 Burndown Chart Commentary

- **Mar 20–24:** Completed Orders UI and began logging integration.
- **Mid Sprint:** Development slowed due to unexpected complexities in Supabase syncing and implementing the export functionality, which temporarily impacted momentum.
- **End Sprint**: The team recovered well, finalizing the Dashboard and Profile UI, and successfully wrapping up the Weather Widget and Analytics components to complete the sprint goals on time.



✅ Sprint 2 Highlights

- After fixing the resources and database back-end and triggers made by Max, we surprisingly found many similar patterns that could be rinsed and repeated across all features Therefore, we separated these similar patterns as composable modules that are shared among all features. These modules were such as:
    - UI components (most important features had tabular functionalities this TanStack table functionality from the Back-end really helped to speed up the creation of features for the rest of the team, **hence the efficient workflow at the end were the team were extremely productive and could ship the features extremely quickly, and the work became very "plug-and-play".**
    - Data fetching functionalities (queries)
    - Hooks
    - Mutations (POST requests, servers actions)

- Live Supabase data integrated across all pages.

- Orders/Customers/Products page supports CRUD functionalities, and search and filters and sort and pagination.

- Profile avatar (with fallback) and routing finalized.

- Home and analytics dashboard completed.

📈 Lessons Learned

- **Task Separation Should Prioritize Layers, Not Features:**
  A key lesson was realizing that dividing tasks by app sections (e.g., customers, products, orders) created unnecessary dependencies between teammates. Frontend developers had to wait for backend logic to be complete, leading to idle time and blocked progress. In future sprints, tasks should be assigned by **general responsibilities**—such as frontend, backend, and integration—so both layers can progress independently and in parallel.

- **Supabase Permissions Require Early Planning:**
  Supabase's RLS system is powerful but unforgiving without proper foresight. We learned that access control policies and role definitions must be planned in detail before development begins, as changing them mid-sprint led to confusion, broken access paths, and more debugging. Establishing permissions early on prevents major delays and ensures a secure foundation from the start.

- **Burndown Tracking Was Steady, Backend Caused Minor Bottlenecks:**
  Our burndown chart showed steady progress overall. Some delays occurred due to backend components taking longer than expected, which temporarily blocked frontend tasks. However, these delays were mitigated with focused catch-up efforts and did not affect the final product demo or delivery date.

- **Edge Case Testing Needs Improvement:**
  While core functionality was well-tested, we identified gaps in handling uncommon but important edge cases—such as failed logins, expired tokens, and empty data states. In the future, we will implement more robust test coverage for these scenarios to ensure the app remains stable and user-friendly in real-world conditions.

🔚 Final Thoughts

- Sprint 1 built a strong base; Sprint 2 layered in complete functionality.

- 31 story points planned across 2 sprints, with 29 completed (94% velocity).

- The team is on track to finalize remaining exports and multi-tenant switching in Sprint 3.