

Kanban Board Project Week 34 Deliverables

Timeline: April 14 – April 18, 2025

Objective:

Following the April 12 demo, this week's focus shifts toward incorporating demo feedback, stabilizing the product for production readiness, conducting performance benchmarking, and documenting the final system configuration. Emphasis will be on prioritizing post-demo refinements, addressing technical debt, finalizing handoff documentation, and aligning all teams for the production release window.

Backend Team Deliverables

April 14–15: Post-Demo Analysis & Feedback Triage

- Review backend issues, gaps, or questions raised during the demo from stakeholders and QA.
- Categorize demo feedback into immediate fixes, enhancements, and long-term backlog items.
- Begin implementing high-priority changes or optimizations identified in feedback.
- Revisit logs captured during the demo to detect latency, failures, or anomaly patterns.

April 16: Technical Debt Remediation & Refactoring

- Clean up unused functions, commented code blocks, and redundant API logic.
- Refactor critical modules for improved readability, scalability, and reusability.
- Conduct peer code reviews focused on maintainability and performance hygiene.

April 17–18: Production Readiness Verification

- Audit API authentication, session expiry, and token renewal workflows.
- Re-validate data integrity across services, including task management, board creation, and user actions.

- Finalize and version API documentation in Postman or Swagger.
 - Lock production config variables, confirm backup and rollback strategies.
-

Frontend Team Deliverables

April 14–15: Post-Demo Fixes & UI Enhancements

- Review demo feedback related to UI/UX glitches, state issues, or visual clarity.
- Patch any bugs identified during the walkthrough or dry-run.
- Improve transitions, button responses, and error messages flagged during demo.

April 16: Performance Profiling & Code Cleanup

- Run Lighthouse and dev tool audits to identify slow-loading components or large bundle sizes.
- Remove unused assets, optimize image and icon delivery.
- Refactor long component trees into reusable sub-components.

April 17–18: Production Freeze Prep & Regression QA

- Prepare and lock the production frontend build, tagging the final commit and branch.
 - Conduct end-to-end regression testing across all browsers and screen resolutions.
 - Ensure build environment variables match production targets.
 - Confirm updated documentation for setup, deployment, and debugging instructions.
-

UI/UX Team Deliverables

April 14–15: Demo Feedback Synthesis & Prioritization

- Compile all visual-related feedback from the demo, both internal and external.
- Highlight critical pain points and propose quick design fixes where applicable.
- Prepare updated Figma files for finalized screens reflecting any changes.

April 16: UI Consistency & Visual Polish

- Reassess consistency across components — button sizes, font hierarchy, icon sets.
- Provide a UI style audit sheet to assist developers in aligning visual standards.
- Update design tokens if applicable (spacing, colors, typography).

April 17: Production-Ready Design Package Finalization

- Deliver full set of final visual assets including logos, illustrations, and optimized SVGs.
- Archive deprecated flows and export final Figma documentation.
- Conduct final sync with frontend team to ensure visual parity.

General & Cross-Team Deliverables

April 14–15: Feedback Tracker & Task Prioritization

- Create and share a consolidated demo feedback tracker categorized by theme and priority.
- Assign JIRA tasks and confirm alignment across dev, design, QA, and PM functions.
- Monitor Clockify for work hours and update daily stand-up logs.

April 16: Risk Mitigation & Release Planning

- Identify any outstanding blockers for production deployment.
- Align timelines, freeze windows, and contingency plans for the go-live event.

- Finalize communication plan for internal and external rollout.

April 17–18: Documentation & Knowledge Transfer

- Ensure final project documentation is up to date in Confluence or internal wiki.
- Update the README with complete setup, environment, and deployment instructions.
- Host a knowledge-sharing session or onboarding walkthrough for long-term maintainers.
- Ensure GitHub branch protection, CI pipelines, and code coverage thresholds are in place.