**Code Integration Lead – Project Report**

**Role and Mandate**

As Code Integration Lead, my primary responsibility was managing the project's GitHub repository. This included setting branch protection rules to prevent accidental merges, reviewing and integrating contributions from team members, and ensuring a clean and structured codebase. Alongside this, I contributed to product framing, documentation, and early technical proof of concepts.

---

**1. Product Framing and Scope**

- Reframed the stream, based on PO guidance, into the **AI Co-Founder** concept: a web-based platform that enables solo retailers to run online stores with the support of AI agents. The system is designed to be simple and intuitive, particularly for low-tech users.
- Defined a lean, testable **MVP: Cart-Recovery**, supported by measurable success metrics such as open rate, click-through rate, and recovery rate, with a clear completion path.

---

**2. Planning and Artefacts**

- Consolidated research outputs into a unified narrative covering the problem, KPIs, ethical considerations, and human-in-the-loop guardrails.
- Produced two anchor visuals for non-technical stakeholders:
    - **Ecosystem Map** showing connections between Agents, MCP, Shopify, Klaviyo, and n8n.
    - **Dashboard Wireframe** covering Store Health, Alerts, Campaigns, Inventory, and an optional chat panel.
- Authored and shared the **Software Requirements Specification (SRS)** and a **Prototype Implementation Guide** for a local setup without the use of paid infrastructure.

---

**3. Technical Progress (Local Proof of Concept)**

- Deployed a local AI stack with **Ollama** (qwen2.5-7B and llama3.1-8B) and a **FastAPI chat endpoint**; round-trip API tests were stable.
- Built a minimal **MCP server** with a tool registry and HTTP methods and validated interoperability using open-source tool specifications.
- Added the first **cart-analysis tool** using synthetic dump data and created a Jupyter notebook to verify metrics and logic.
- Initiated **n8n orchestration** using Docker and documented the end-to-end flow from webhook to agent to email.

---

**4. Collaboration and Cadence**

- Paired with Kris on UI and UX: completed **Dashboard Draft v1** and initiated **Draft v2** to deliver a more professional SaaS-style design.
- Proposed short huddles, smaller task slices, and clearer ownership to address low activity within the stream.
- Provided documentation and setup notes to help other members onboard and contribute more effectively.

---

**Challenges and Mitigation**

- **n8n and MCP node issues**: encountered container networking errors (ECONNREFUSED) and schema validation problems such as missing required property name. *Mitigation*: temporarily shifted to direct HTTP calls while isolating nodes and refining schema mappings.
- **Low team contribution**: flagged the risk early, simplified tasks, and offered pairing and onboarding support to maintain progress.

---

**Evidence and Artefacts (Shared)**

- **SRS** covering scope, architecture, roles, functional and non-functional requirements, data model, example MCP tools, and MVP acceptance criteria.
- **Prototype Guide** covering n8n Docker setup, MCP server with FastAPI, Ollama configuration, Klaviyo integration, and webhook test.
- Ecosystem diagram, dashboard wireframe, synthetic dataset, and Jupyter notebook for cart analysis.

---

**Next Steps (Toward a Demo)**

1. Resolve MCP schema and networking issues and complete the n8n to MCP to Klaviyo integration.
2. Embed chat with MCP tool list within Kris's dashboard (Draft v2) and record a short demo video.
3. Push the repository scaffold and artefacts to GitHub, adding contribution rules and onboarding documentation.
4. If time allows, implement one additional automation such as segment preview and email draft.