Final Project Plan – Agent-Driven ERP System (Helios Dynamics)

Teammate 1 – Data & Infrastructure Specialist (Database + Deployment)

Objective: Establish the ERP system's foundation with database design, backend, and deployment setup.

Responsibilities:

- Design and initialize ERP database (**SQLite prototype** → Postgres migration path).
- Define schema for customers, orders, invoices, suppliers, inventory, ledger, glossary.
- Implement approval flows, audit logs, and compliance tables.
- Build backend with FastAPI.
- Prepare Dockerfile/docker-compose for deployment.

Deliverables:

- erp_sample.db with populated test tables.
- SQL schema documentation with migration notes.
- FastAPI backend + deployment files.
- Logs/screenshots proving database connections.

Teammate 2 – MCP & Tooling Engineer (Tool Registry + Adapters)

Objective: Configure MCP and ensure all tools are registered, available, and agent-compliant.

Responsibilities:

- Configure MCP (Modular Composable Protocols) for ERP.
- Register domain-specific tools: SQL read/write, RAG tools, anomaly detectors, glossary search, visualization adapters.
- Ensure agents comply with MCP tool interface standards.

- Implement adapters for tool integration (SQL adapters, RAG retrievers, ML models).
- Test MCP-tool-agent pipelines.

Deliverables:

- MCP tool registry file with all registered tools.
- Python adapters for SQL, RAG, and ML models.
- Logs/screenshots of successful tool usage.
- Config documentation for MCP setup.

Teammate 3 – Agent Developer (Router + Core Domain Agents)

Objective: Implement multi-agent workflows and routing logic.

Responsibilities:

- Build Router Agent to handle intent detection, routing, and approval enforcement.
- Develop at least 3 domain agents:
  - Sales/CRM Agent → SQL read/write, RAG search, lead scoring.
  - Finance Agent → invoice processing, anomaly detection, finance policy RAG.
  - Inventory Agent → stock monitoring, forecasting, supplier lookup.
- Integrate memory modules (ConversationBufferMemory + entity/global memory persisted in DB).
- Validate workflows (sales entry, invoice anomaly, stock reordering).

Deliverables:

- Router Agent + 3 domain agents Python scripts.
- Logs/screenshots of end-to-end workflows.
- Config files for tool registration + memory setup.

Teammate 4 – Analytics & ML Specialist (Visualization + Intelligence)

Objective: Enable advanced insights, reporting, and ML-driven decisions.

Responsibilities:

- Implement Analytics Agent for SQL-to-Chart workflows.
- Build RAG-powered glossary/document retriever for schema and policies.
- Train or implement ML models (e.g., lead scoring, anomaly detection, forecasting).
- Add visualization pipeline (matplotlib/Plotly/Streamlit charts).
- Optimize performance (low response latency).

Deliverables:

- Python code for Analytics Agent with chart generator.
- ML model for at least one ERP feature.
- Example visualizations (sales by region, top products, anomalies).
- Logs of ML model predictions and analytics results.

Teammate 5 – UX & Documentation Lead (Conversational Interface + Reporting)

Objective: Build the user-facing interface and package the project deliverables.

Responsibilities:

- Develop chat-based interface (Streamlit minimum, optional React).
- Integrate agents into the interface with conversational workflows.
- Write README with: setup instructions, architecture, queries, and explanation of RAG benefits.
- Compile test queries and screenshots for report.
- Coordinate and edit the 10-minute demo video (3 segments across teammates).

Deliverables:

- Functional Streamlit/React interface.

- README with instructions, sample queries, and design explanation.
- At least 10 test queries with logs/screenshots.
- Final demo video showcasing full system.

🎥 10-Minute Demo Video Division

Teammate 1 – Data & Infrastructure Specialist (2 minutes)

Demo Focus:

- Show the ERP database (erp_sample.db) with key tables (customers, orders, invoices, stock).
- Walk through MCP tool registry setup and tool registration (SQL read/write, RAG, anomaly detector).
- Briefly show Dockerfile + running backend (uvicorn or docker-compose up).

Talking Points:

- "Here's the ERP database schema with customers, orders, and stock tables."
- "We connected these with the MCP registry so agents can securely call tools."
- "Finally, the system is containerized for easy deployment."

Teammate 2 – Router Agent Developer (2 minutes)

Demo Focus:

- Explain Router Agent as the "ERP brain."
- Show it classifying and routing queries: e.g., "Add a new customer" → goes to CRM Agent.
- Show audit log entry and memory recall across two queries.

Talking Points:

- "The Router classifies the intent and sends it to the right domain agent."
- "It enforces policies like requiring approvals and logs all actions."
- "The system remembers previous context, so it can refine multi-turn queries."

Teammate 3 – Sales & CRM Agent Developer (2 minutes)

Demo Focus:

- Show CRM agent adding a new customer and storing details in DB.
- Run a lead scoring example (new lead → assigned score).
- Show order entry workflow integrated with Router.

Talking Points:

- “Here, the CRM Agent handles customer onboarding.”
- “It can also score leads based on rules or ML models.”
- “Orders are directly logged into the ERP database.”

Teammate 4 – Finance & Inventory Agent Developer (2 minutes)

Demo Focus:

- Finance Agent: run an invoice query and detect an anomaly (e.g., incorrect total).
- Inventory Agent: show stock monitoring and an auto-reorder trigger when levels drop.
- Show supplier contract retrieval using RAG.

Talking Points:

- “The Finance Agent ensures financial accuracy and flags anomalies.”
- “The Inventory Agent tracks stock levels and automatically creates supplier orders.”
- “This ensures smooth business operations with fewer manual errors.”

Teammate 5 – UX & Documentation Lead (2 minutes)

Demo Focus:

- Walk through the Streamlit/React conversational interface.
- Show an analytics query like: “What are top-selling products this quarter?” → SQL → chart → narrative explanation.
- Highlight README structure + quick setup steps.

Talking Points:

- “The conversational interface makes the ERP easy to use.”

- “Analytics are generated with text-to-SQL and visualized as charts.”
- “Documentation explains setup, architecture, and workflows for reproducibility.”