TestBuddy – Phase 2 Summary (MCP Core + Agent Framework)

# 1. Overview

Phase 2 established the MCP (Modular Control Plane) Core and Agent Framework within the FastAPI backend. This included integrating multiple autonomous agents — Planner, Designer, Author, Executor, and Curator — each designed to handle specific testing phases intelligently. The framework successfully enabled modular control, dynamic routing, and independent agent communication. The backend was deployed on Render and verified end-to-end.

# 2. MCP Core Architecture

• Implemented a registry-based Modular Control Plane to orchestrate agent-level tasks.  
• Added five autonomous agents:  
 – Planner: Generates a high-level test plan from intake sources.  
 – Designer: Converts outlines into detailed test scenarios.  
 – Author: Generates Cypress + Cucumber-based feature files and step definitions.  
 – Executor: Simulates running of generated test suites and returns summarized results.  
 – Curator: Analyzes run outputs and provides improvement insights.  
• Ensured each agent communicates independently through FastAPI routers.  
• Confirmed modular design supports easy agent replacement and extension.

# 3. Backend Enhancements

• Integrated all agent routes into FastAPI’s main application.  
• Standardized the toolchain by consolidating Cypress + Cucumber as the only supported framework.  
• Defined dedicated Pydantic models for structured request/response handling.  
• Verified functionality via Swagger UI and command-line curl commands.  
• Enabled full modular operation of Planner, Designer, Author, Executor, and Curator.

# 4. Deployment Verification

• Successfully deployed backend to Render at:  
 https://test-buddy-b110.onrender.com  
• Validated API routes via Swagger UI (`/docs`) and health endpoint (`/health/`).  
• Verified the Author Agent returning Cypress+Cucumber artifacts with feature and step definitions.  
• Confirmed complete end-to-end connectivity between backend modules.

# 5. Frontend Preparation

• Configured environment variable in Vercel frontend:  
 NEXT\_PUBLIC\_API\_BASE=https://test-buddy-b110.onrender.com  
• Frontend successfully integrated backend route verification indicator (Backend OK status).  
• Prepared groundwork for frontend agent interactions and UI feedback flow for the next phase.

# 6. Success Indicators

|  |  |  |
| --- | --- | --- |
| Component | Validation | Status |
| MCP Core Registry | Operational on Render | ✅ |
| Agent Routes | /plan, /design, /author, /execute, /curate | ✅ |
| Cypress+Cucumber Framework | Generated feature + step definitions | ✅ |
| Swagger Documentation | Accessible via /docs | ✅ |
| Frontend Linkage | Verified via environment variable | ✅ |

# 7. Next Phase – Data Persistence and Orchestration

The next phase (Phase 3) will focus on implementing persistent data storage and agent orchestration. Each agent’s output will be stored for traceability, and frontend interactions will synchronize with backend analytics. This phase will establish stateful insights and feedback loops for the TestBuddy platform.