PROJECT PLANNER: The Agile Dev Studio

Module: Programming & Software Development (MVP Sprint)

School: ___________________________ | **Teacher:** __________________________

Project Dates: ____________________ | **Pathway:** CTE Computer Science / SWD

I. PROJECT OVERVIEW

You are working as a **Junior Software Development Team**. Your mission is to move a concept from a "User Need" to a **Minimum Viable Product (MVP)**. You will be graded not just on the code that works, but on the **professional process** you use to build it.

II. SCOPE OF WORK (The "What")

You will build a functional web application. Your scope is intentionally limited to ensure high-quality code. Choose one:

- **School Management:** Club sign-up or event tracker.
- **Community Utility:** Simple inventory or service request form.
- **Data Tool:** Personal finance or goal-tracking interface.

III. PROJECT PHASES & MANDATORY CHECKPOINTS

You must receive a teacher signature at each checkpoint before moving to the next phase.

Phase 1: Discovery & Design (The Blueprint)

- Define the Target User & "User Stories."
- Create wireframe sketches of the UI.
- **CHECKPOINT 1:** Design Review & Scope Approval [Teacher Initial: ____]

Phase 2: Development Sprints (The Build)

- Setup GitHub Repo & Kanban Board.
- Implement core features in 1-week sprints.
- Document progress/blockers daily.
- **CHECKPOINT 2:** Mid-Project Code Review [Teacher Initial: ____]

Phase 3: Quality Assurance & Testing (The Polish)

- Peer-test another team’s application.
- Log bugs and apply fixes.
- Finalize README.md documentation.
- **CHECKPOINT 3:** QA Bug Log Completion [Teacher Initial: ____]

Phase 4: Demo Day (The Launch)

- Deliver a live 3-minute technical demonstration.
- Submit individual Post-Mortem Reflection.
- **FINAL SUBMISSION:** [Teacher Initial: ____]

IV. TEAM ROLES

Every student is a Developer, but you must also hold one "Agile Lead" responsibility:

1. **Project Manager:** Owns the Kanban board and ensures milestones are met.
2. **Backend Lead:** Owns data flow, server-side logic, and validation.
3. **UX/UI Lead:** Owns CSS, accessibility (WCAG), and interface layout.
4. **Technical Writer:** Owns the GitHub README and internal code comments.

V. AI USE POLICY (The "Code of Honor")

AI is a **Co-Pilot**, not an Autopilot.

- **YES:** Asking AI to explain an error message or suggest a CSS property.
- **NO:** Generating an entire function or file without being able to explain every line.
- **REQUIRED:** If you use AI for a block of code, you **must** add a comment above it: // Assisted by [AI Name] for [Specific Purpose].

VI. ASSESSMENT WEIGHTING

Refer to the provided Project Rubric for specific grading descriptors.

Category	Weight	Evidence
Product Reliability	40%	Functional MVP Link
Professional Workflow	30%	Git Commits & Kanban Board
Documentation & Reflection	20%	README & Post-Mortem
Participation	10%	Peer Feedback & Daily Stand-ups

Refinements made for NYC Standards:

1. **Teacher Initials:** Added physical "Checkpoint" boxes. This is a common requirement in NYC CTE portfolios to prove the teacher was involved in the formative process.
2. **Terminology:** Changed "Project Dates" to include "Pathway" info—administrators look for this.
3. **Role Clarity:** Explicitly stated "Every student is a Developer." This prevents the "Project Manager" from not writing any code, which is a common issue in HS projects.
4. **AI Tagging:** Added a requirement for "In-code comments" for AI use. This is a very strong academic integrity move that NYC Tech Leads recommend.