Cat 3 (Trimester 9–12+)

Awards: Best Product Potential, Best Code Quality

Criteria	Exemplary (4)	Proficient (3)	Developing (2)	Beginning (1)
Theme Relevance & Product Value	Product strongly aligns with the theme and presents a clear, high-impact real-world use case; demonstrates strong value and purpose	Product aligns well with the theme; use case is meaningful and practical	Partial alignment with theme; product idea is somewhat generic or lacks depth	Weak or unclear link to theme; product lacks relevance or value
System Design & Architecture	Well-defined architecture with proper modularity, API/database integration, and structured design flow; scalable and maintainable	Logical and consistent design; integrates multiple components with reasonable structure	Some structural clarity but lacks cohesion; design partially functional	Disorganized or missing architecture; lacks modularity or design logic
Implementation & Functionality	All key features work smoothly; database and API integrations are reliable; minimal bugs; demo is polished and complete	Most features functional; minor glitches or incomplete modules	Some core functions incomplete or unstable; partial demo	Limited functionality; product fails to run as expected
Code Quality & Engineering Practice	Clean, efficient, well-commented code following software engineering standards; consistent naming, modular structure	Mostly clean and maintainable code; some inconsistencies in structure or documentation	Functional code but lacks clarity, structure, or proper engineering practice	Poorly written or unstructured code; lacks maintainability or clarity

Presentation & Explanation	Clear, professional presentation demonstrating technical flow, architecture, and product potential; feasible for future development	Good presentation with understandable explanation of design and functionality	Somewhat unclear presentation; lacks discussion of feasibility or design	Weak presentation; unable to justify decisions or explain functionality

Judging Rubric - Weight Distribution

Criterion	Best Product Potential	Best Code Quality	Rubric Notes
Theme Relevance & Product Value	30%	10%	Alignment with theme, clarity of purpose, and real-world applicability
System Design & Architecture	25%	25%	Technical structure, scalability, modularity, and integration
Implementation & Functionality	25%	30%	Core features, reliability, execution, and overall usability
Code Quality & Engineering Practice	10%	30%	Code readability, efficiency, maintainability
Presentation & Explanation	10%	5%	Ability to communicate design, demonstrate feasibility, and showcase product value

Example: If a team is evaluated below,

Criterion	Score (?/4)
Theme Relevance & Product Value	**3**
System Design & Architecture	**3**
Implementation & Functionality	**3**
Code Quality & Engineering Practice	**4**
Presentation & Explanation	**4**

Score type	formula	score
Original	3 + 3 + 3 + 4 + 4	17**/20**
Best Product potential	(3×0.30) + (3×0.25) + (3×0.25) + (4×0.10) + (4×0.10)	3.05**/4**
Best Code Quality	(3×0.10) + (3×0.25) + (3×0.30) + (4×0.30) + (4×0.05)	3.25 **/4**

Focus:

Projects in Cat 3 should demonstrate end-to-end design and implementation capability, covering system architecture, development, and a functional prototype all within the 6-hour development window.

Participants may develop any form of software solution, including but not limited to:

- APIs or backend systems
- Console tools or automation utilities
- Websites or full-stack web applications
- Mobile or cross-platform applications

The solution does not need to be fully complete, but it should present a minimally functional prototype that clearly reflects:

- A well-thought-out architecture or system plan
- Proper logical flow and coding discipline
- Usable and testable core features relevant to the theme

While the use of advanced frameworks, libraries, or AI integrations is encouraged, the emphasis remains on:

- Product completeness and technical soundness
- Maintainable, modular, and readable code
- Real-world feasibility and clarity of vision, rather than excessive feature-building or over-engineering.