RACI Matrix

Who does what on the Mario's Pizzeria Kiosk project

Project	Mario's Pizzeria Automated Ordering Kiosk
Client	Mario's Pizzeria (classroom brief)
Delivery team	Shelly Software Co. (student developer, sponsor, AI pair-programmer)
Version	1.2 — adds v1.2 visual enhancement tasks and updated QA documents
Date	9 September 2026
Status	Approved

1. What a RACI is

A RACI matrix lists every task in a project and says who is Responsible for doing it, who is Accountable for making sure it gets done, who should be Consulted before or during, and who just needs to be Informed afterwards. Real engineering teams use it so nobody assumes someone else is doing the work.

R	Responsible — does the work. At least one R per task.
A	Accountable — owns the outcome and signs it off. Exactly one A per task.
C	Consulted — asked for input before the work is finished (two-way).
I	Informed — told when it is done (one-way).

2. Roles

- Student Developer — writes and owns the Python code; this is the person being assessed.
- Project Sponsor (Dad) — provides the tools, hosting and the "what does a real project look like" layer.
- Customer / Mario (Teacher) — set the brief and decides whether the result meets it.
- AI Pair-Programmer (Claude) — drafts documents and website, explains code, never replaces the student's own understanding.
- QA Tester (Family) — orders pizzas and tries to break the kiosk.

3. The matrix

Task / Deliverable	Student Developer	Project Sponsor (Dad)	Customer / Mario (Teacher)	AI Pair-Programmer (Claude)	QA Tester (Family)
Read and understand the brief	**R**	C	**A**	C	I
Write Business Requirements Specification	C	**R**	**A**	**R**	I
Define RACI and roles	I	**A/R**	I	C	I
Design the ordering flow (questions, prices)	**A/R**	C	C	C	I
Code pizza_bot.py (core requirements)	**A/R**	C	I	C	I
Code extensions (garlic bread, delivery)	**A/R**	I	I	C	I
Code pizza_turtle.py (Shelly draws the pizza)	**R**	C	I	**R**	I
Test against the example output ($15 Sarah case)	**R**	C	I	C	**A/R**
Build the showcase website and web kiosk	C	**A**	I	**R**	I
Host on pizza.locbee.ai (DNS, Pages deploy)	I	**A/R**	I	C	I

Task / Deliverable	Student Developer	Project Sponsor (Dad)	Customer / Mario (Teacher)	AI Pair-Programmer (Claude)	QA Tester (Family)
Present / demo the project	**A/R**	C	I	I	I
Submit pizza_bot.py to the teacher	**A/R**	I	I	I	I
v1.1 — Build enhanced.html (animations, receipt send, log, new menu)	C	**A**	I	**R**	C
v1.1 — Code pizza_bot_enhanced.py (orders.log, new options)	C	**A**	I	**R**	I
v1.1 — Update BRS with optional requirements FR-14 to FR-21	I	**A**	I	**R**	I
v1.1 — Write Requirements Traceability Matrix	I	**A**	I	**R**	C
v1.1 — Execute test cases and write Test Summary Report	C	**A**	I	**R**	**R**
v1.1 — Run pizza_turtle.py on a real screen (TC-6)	**R**	**A**	I	I	C
v1.2 — Build deluxe.html (pizzeria-at-night visual experience, canvas pizza animation, Shelly reactions, oven render, receipt printer)	C	**A**	I	**R**	I
v1.2 — Execute deluxe.html test cases (TC-12, TC-13, TC-14)	C	**A**	I	**R**	**R**
Security — define the security requirement (NFR-06) and what "secure enough" means for a static site	C	**A**	I	**R**	C
Security — review the code and site for injection, secrets, headers and third-party risk	C	**A**	I	**R**	C
Security — execute TC-15 and record findings before go-live	C	**A**	I	**R**	**R**
Security — accept or reject residual security risk at sign-off	I	**A**	C	I	I

4. Rules we follow

- Every task has exactly one A. If two people are accountable, nobody is.
- The Student Developer is A/R for all assessed code — the AI can be Consulted, not Responsible, for pizza_bot.py.
- If a task gets stuck, the A for that task raises it at the next family stand-up.

5. Notes

The turtle drawing and the website are extras that show how the same requirements can be delivered in more than one way. They are not part of the school submission and so the Student is only R (not A) for them.

v1.1 rows (marked in the matrix) cover the optional enhancements. The AI Pair-Programmer is Responsible for building them and the Sponsor is Accountable; the Student is Consulted so they can explain what changed, but never assessed on it. TC-6 is the one task still open: the turtle drawing has to be run on a real Windows screen, and that is the Student's job.