### 🧩 **Puzzle Specification Template**

**Instructions:** Use this template to define each puzzle in your escape room. Every puzzle must be clearly described so it can be built, tested, and reset by someone else. This is a living document — update it as you iterate.

#### 🧠 Puzzle Title:

*What is the name or internal label for this puzzle?*

### 1. 🎯 **Puzzle Objective**

* What should the players accomplish?
* What knowledge or skill do they need to solve it?

*Example: Redirect a laser beam using mirrors to activate a hidden sensor.*

### 2. 🔍 **Clue & Discovery Structure**

* How do players discover this puzzle?
* Are there clues hidden in the room, embedded in narrative, or given directly?
* Does it depend on solving another puzzle first?

*Example: The player finds a power diagram inside a hidden folder that hints at aligning cables by color.*

### 3. 🧪 **Puzzle Mechanics**

* Step-by-step description of how the puzzle works
* What are the inputs, what process is required, and what triggers success?

*Example: Players place mirrors on marked grid points. When the laser hits the sensor, the circuit completes and a lock opens.*

### 4. 🔧 **Components and Materials**

* List all required items (physical and/or digital)
* Include tools, props, electronics, software, etc.

*Example: 1x laser pointer module, 3x acrylic mirrors, 1x light sensor, Arduino with relay.*

### 5. ✅ **Solution Path**

* What is the correct sequence of actions or logic?
* Include diagrams if helpful.

*Example: Mirrors placed on B2, D3, F1 to direct beam to target sensor.*

### 6. 🧠 **Player Experience**

* What should players feel during this puzzle (e.g., tension, satisfaction, confusion)?
* Estimated difficulty: 🔹 Easy / 🕘 Medium / 🔺 Hard

### 7. 🔄 **Reset Instructions**

* How is the puzzle returned to its starting state?
* Who is responsible, and how long should it take?

*Example: Staff places mirrors back in the tray and resets the laser alignment — ~30 seconds.*

### 8. ⚠️ **Failure Modes / Risks**

* What can go wrong? (e.g., tech failure, player misuse, design ambiguity)
* How will you mitigate these?