1. Project Idea

- Briefly explain what your project is about.

  This project is a Number Guessing game where the player tries to guess a sequence of randomly generated digits (0–9) across multiple columns. The player has a limited number of rows (attempts) to guess it right.

- What will it do? What's the main purpose or goal?

  The goal is to create an interactive, logic-driven challenge that encourages players to deduce the correct digits and their positions using visual feedback.

- Imagine you're pitching this to someone who’s never heard of it before — make it clear and simple.

  Example: "I'm building a simple quiz app that lets users answer multiple-choice questions and gives them a score at the end."

  I'm building a number guessing game where the computer picks a sequence of digits, and the player has to guess them row by row. After each guess, the game shows color-coded feedback. Green for correct digits in the right place, orange for correct digits in the wrong place, and red for incorrect digits. The player wins by guessing all digits correctly before running out of attempts.

2. Win Conditions / Success Criteria

- What are you trying to achieve with your project?

  Create a fully functional guessing game that is intuitive, responsive, and gives real-time feedback.

- What does a finished or working version of your project look like?

  1. "Start" button to initialize the game
  2. Grid of input fields and "Check" buttons for each row
  3. Feedback after each guess using color-coded tiles
  4. Timer that tracks how long the player takes
  5. High score display based on shortest time
  6. Win message when all digits are guessed correctly
  7. Loss message when all rows are used
  8. "Restart" option to play again

- What features must be present for it to be considered complete?
  Example: "A user can start a quiz, answer all the questions, see their final score, and restart the quiz."

  A player can start the game, enter guesses row by row, receive visual feedback, win or lose based on accuracy and attempts, and replay to improve their time.

3. How You Will Build the Logic

- Walk me through your plan for how the app will work behind the scenes.
- A rough breakdown of steps or logic flow.

- **Initialize Game**
    - Generate a unique set of random digits (no duplicates)
    - Display ‘?’ placeholders for the target digits
    - Create input fields and buttons dynamically for each row
    - Reset timer, messages, and previous inputs
    - Show previous best time if available

- **Capture User Input**
    - Enable inputs for the first row
    - Disable all other rows until the current guess is submitted
    - Validate each input (must be a digit between 0–9)

- **Compare & Feedback**
    - Green: correct digit and position
    - Orange: correct digit, wrong position
    - Red: incorrect digit
    - Show feedback immediately after each guess
        - If correct, stop timer and show win message
        - If incorrect, move to next row

- **End Game**
    - If all rows are used without a correct guess, show "Game Over"
    - If guessed correctly, show "Victory!"
        - Compare elapsed time to best time and update if faster
    - Allow player to restart the game