**JavaScript Terminal Project – Riddle Game: Step-by-Step Tasks**

**📁 Folder & File Setup Tasks**

**Task 1: Project Folder Structure**

* Create a main folder called riddle-game.
* Inside riddle-game, create:  
  + app.js → Runs the entire game.
  + riddles/ folder → Store your riddle files.
  + classes/ folder → Store your class files.

**Example Structure:**

riddle-game/

├── app.js

├── riddles/

├── classes/

**📄 Riddle File Creation Tasks**

**Task 2: Create Example Riddle Files**

* Inside riddles/, create files like r1.js, r2.js.
* Each file should export a riddle object like this:

// riddles/r1.js

export default {

id: 1,

name: "Easy Math",

taskDescription: "What is 5 + 3?",

correctAnswer: "8"

};

* Create at least 2 riddle files to start.

**🏋️‍♂️ Class Building Tasks**

**Task 3: Create the Riddle Class**

* File: classes/Riddle.js
* Properties:  
  + id
  + name
  + taskDescription
  + correctAnswer
* Method:  
  + ask(prompt)  
    - Show the riddle.
    - Ask for user input.
    - Keep asking until the correct answer is given.

**Task 4: Create the Player Class**

* File: classes/Player.js
* Properties:  
  + name
  + times[] (Array of time durations per riddle)
* Methods:  
  + recordTime(start, end)
  + showStats() → Show total and average time.

**🎮 Game Logic Tasks (app.js)**

**Task 5: Setup Terminal Input**

* Install prompt-sync:

npm install prompt-sync

* Setup input:

import PromptSync from 'prompt-sync';

const prompt = PromptSync();

**Task 6: Ask for Player Name**

* Show welcome message.
* Use prompt to ask: "What is your name?"
* Create a Player object with the name.

**Task 7: Load All Riddles at Once**

* Create an index file in riddles/:

import r1 from './r1.js';

import r2 from './r2.js';

export default [r1, r2];

* In app.js:

import AllRiddles from './riddles/index.js';

**Task 8: Main Game Loop**

* Loop through all riddles:  
  + Record startTime with Date.now().
  + Create a Riddle object and call ask().
  + Record endTime.
  + Save time using player.recordTime().

**Task 9: Show Final Stats**

* After all riddles:  
  + Show total time.
  + Show average time per riddle.

**⏱️ Optional Decorator for Timing**

* Wrap the riddle with timing logic:

const start = Date.now();

riddle.ask(prompt);

const end = Date.now();

player.recordTime(start, end);

**🌟 Bonus Challenge: Multiple Choice Riddles (Optional)**

**Task 10: Create MultipleChoiceRiddle Class**

* Inherit from Riddle.
* Add choices[] property.
* Show numbered options.
* Accept numeric input and compare to correct answer.

Example structure:

export default class MultipleChoiceRiddle extends Riddle {

constructor(details) {

super(details);

this.choices = details.choices;

}

ask(prompt) {

// Show choices and handle input

}

}

**📌 Final Deliverables**

* Project folders and files ready.
* At least 2 riddles defined.
* Working Riddle and Player classes.
* app.js runs the full game.
* Terminal input with prompt-sync.
* Time tracking for each riddle.
* Final stats display.
* Optional: Multiple-choice riddles.