**✅ Prompt for ChatGPT:**

You are an expert Python and Pygame developer. Write a complete, error-free Python program for a **Snake game** using **Pygame**, including all features described below. The code should be modular, visually clean, and runnable without any undefined variables or missing elements.

**✅ Game Requirements:**

1. **Dynamic Window Size**: The game window should be resizable, and the playfield must adjust accordingly.
2. **Main Menu**:
   * Visually pleasing design with well-spaced options.
   * Options: “Play”, “Select Theme”, “View High Scores”, and “Quit”.
3. **Theme Selection Menu**:
   * A separate in-game screen for choosing from 3 themes.
   * Each theme should include a background and snake color.
   * Include a **live visual preview box** for each theme (with contrasting colors for visibility).
4. **Game Loop**:
   * Starts by asking the player to choose the difficulty (Easy, Medium, Hard) **in-game**, not via console.
   * Snake starts with an initial length of **3 segments**, facing left.
   * Snake should **move initially** without waiting for user input.
   * Arrow keys to control, ‘P’ to pause/resume.
   * Snake grows by 1 segment when eating food.
   * Food appears randomly in bounds.
   * Game over if snake hits itself or the wall.
5. **Pause/Resume Feature**:
   * Press ‘P’ to pause/resume.
   * While paused, screen should display "Paused" message.
6. **High Score System**:
   * Scores are tracked **per difficulty level**.
   * Scores are stored in a local highscores.json file.
   * When a player **beats any of the top 5 scores** of that mode, prompt for the player’s name in-game (not in console).
   * Only **top 5 scores** should be stored per mode.
   * Add option in the main menu to **view all high scores**, grouped by difficulty.
7. **Audio**:
   * Background music plays during gameplay.
   * Sound effects on eating food and on game over.
8. **Code Requirements**:
   * Use Pygame's built-in methods (no external UI libraries).
   * Use modular functions (main\_menu, game\_loop, select\_theme\_menu, pause\_game, etc.).
   * Define all necessary fonts, colors, sounds, and file paths.
   * No undefined variables or missing functions.
   * Use pygame.RESIZABLE to support dynamic window size.

**Additional Notes:**

* Include appropriate imports and file reading/writing for highscores.json.
* Predefine all fonts (e.g., title\_font, menu\_font, font\_style) properly using pygame.font.Font.
* Assume audio files (e.g., eat.wav, gameover.wav, background music) are in the same directory.
* If files are missing (e.g., sounds or JSON), handle them gracefully (e.g., skip audio or create empty JSON file).