Creating a **wxWidgets GUI application** for a stopwatch with timer options in Python is a fun and practical project! Below, I'll provide you with a blueprint for the app and an implementation outline.

**Key Features:**

1. **Multiple Stopwatch Tracking**: Ability to create multiple stopwatches with a description for each.
2. **Pause/Resume/Stop**: Functionality for pausing, resuming, and stopping each stopwatch.
3. **SQLite Database Integration**: Store stopwatch data persistently.
4. **Export to CSV**: Export the logged time and descriptions to a CSV file.
5. **Activity Categorization**: Track activities (e.g., "Programming," "Exercise").

**Implementation Plan:**

1. **GUI with wxWidgets**:
   * Create a main window with controls for managing stopwatches.
   * Use a grid or list to display active stopwatches.
   * Provide buttons for actions (Start, Pause, Resume, Stop, and Export).
2. **SQLite Database**:
   * Store records for stopwatch sessions (id, description, start\_time, end\_time, duration).
3. **CSV Export**:
   * Allow exporting all data or filtered data (e.g., by date).
4. **Backend Logic**:
   * Use Python's datetime or time module to manage time tracking.
   * Use a list or dictionary to track active stopwatches in real-time.

**Libraries:**

* **wxPython**: GUI library.
* **sqlite3**: Database library (built-in).
* **csv**: CSV handling (built-in).
* **pandas**: Optional, for easier CSV handling.

**Next Steps:**

1. Add Pause/Resume/Stop buttons dynamically for each stopwatch.
2. Update durations in real-time using a timer event.
3. Refine the GUI layout for better usability.

Let me know if you want to expand any specific feature! 🚀