**README: Stopwatch Tracker Application**

**Stopwatch Tracker**

A desktop application for tracking time for various activities with features to pause, resume, stop, export, and view historical data. The application uses **wxWidgets** for the graphical user interface (GUI) and **SQLite** for data storage.

**Features**

1. **Stopwatch Management**:
   * Start, pause, resume, and stop multiple stopwatches.
   * Track individual durations in real-time.
2. **Data Storage**:
   * Automatically save activity details (description, start time, end time, duration) to an SQLite database.
3. **Export Data**:
   * Export historical activity data to a CSV file.
4. **View Data**:
   * Display historical activity data in a tabular format within the application.
5. **Real-Time Updates**:
   * Each stopwatch updates its duration every second while running.

**Requirements**

1. **Dependencies**:
   * **wxWidgets**: A cross-platform GUI toolkit.
   * **SQLite3**: A lightweight relational database engine.
2. **Compiler**:
   * A C++ compiler (e.g., g++) with support for C++17 or later.

**How to Use**

1. **Start a Stopwatch**:
   * Enter an activity description in the input box.
   * Click **Start** to begin tracking time for the activity.
   * A new row will appear in the interface for the stopwatch.
2. **Pause, Resume, Stop**:
   * Use the **Pause**, **Resume**, and **Stop** buttons to control individual stopwatches.
3. **Export Data**:
   * Click **Export Data** to save all historical records to a stopwatch\_data.csv file in the current directory.
4. **View Data**:
   * Click **View Data** to display all historical records in a new window.

**File Descriptions**

1. **StopwatchApp.cpp**:
   * Main application source code.
   * Implements all functionalities including GUI, database operations, and event handling.
2. **stopwatch.db** (Generated at runtime):
   * SQLite database file for storing activity records.
3. **stopwatch\_data.csv** (Generated upon export):
   * CSV file containing exported historical activity data.

**Known Issues**

* **Window Resizing**: The interface may not automatically resize on smaller screens. Resize the application window manually if needed.
* **Empty Database**: If no records exist, the **View Data** window will open with an empty grid.

**Future Enhancements**

1. Add support for sorting and filtering historical data.
2. Enhance the user interface for better readability.
3. Add notifications for long-running stopwatches.