

Kat's Concert Events - WIX Setup

Wix Concert Events Page Using CMS, Repeaters, and Velo

Purpose

This setup replaces a manually maintained concert-events page with a Wix CMS-backed page.

The goal is to maintain all concert records in **one CMS collection**, while Wix automatically displays them in two separate sections:

- **Upcoming Events**
- **Past Events**

Events are automatically moved from Upcoming to Past based on their date.

1. Create the Concerts CMS Collection

In Wix:

1. Open the site editor.
2. Open **CMS**.
3. Create a new collection.
4. Choose **Content Collection**.
5. Name the collection:

Concerts

Use a multiple-item collection, where each row represents one concert or event.

Recommended fields:

Display Name	Example Field ID	Type
Event Name	eventName	Text
Date	date or eventDate	Date
Venue	venue	Text
City	city	Text
State	state	Text
Description	description	Text or Rich Text
Ticket URL	ticketUrl	URL
Link Text	linkText	Text
Published	published	Boolean

Additional temporary or reference fields can also exist, but they are not required for the automatic Upcoming/Past logic.

Important: the Date field must be a real **Date** field, not a text field containing a date-looking value.

2. Populate the CMS Collection

The original concert list was imported from an Excel spreadsheet by saving it as CSV and importing the CSV into the `Concerts` collection.

After the initial import, the CMS collection can simply be maintained directly in Wix.

For normal maintenance:

1. Open **CMS**.
2. Open the `Concerts` collection.
3. Add a new row for each new concert.
4. Edit existing rows as needed.
5. Enter the proper event date.

There is no need to manually mark an event as Upcoming or Past.

The page determines that automatically.

3. Create the Upcoming Events Repeater

On the Shows page:

1. Add a **Repeater**.
2. Place it under the **Upcoming Events** heading.
3. Design one repeater item to match the desired page layout.

Typical elements inside the repeater might include:

- Date
- Event Name
- Venue
- City
- State
- Description
- Ticket button or link

Connect the repeater to the `Concerts` CMS collection through a dataset.

4. Create the Upcoming Events Dataset

Create a page dataset connected to the `Concerts` collection.

Recommended dataset ID:

`#upcomingDataset`

Configure the dataset sort as:

`Date → Ascending`

This causes the nearest upcoming concert to appear first.

Do not create a separate CMS collection for upcoming events.

The Upcoming dataset is only a filtered view of the same `Concerts` collection.

5. Connect the Upcoming Repeater to the Upcoming Dataset

This step is important.

The repeater itself and every connected element inside the repeater must reference the **Upcoming dataset**, not the raw CMS collection.

For example:

Upcoming Repeater

Event Name → Upcoming Dataset / Event Name

Date → Upcoming Dataset / Date

Venue → Upcoming Dataset / Venue

City → Upcoming Dataset / City

State → Upcoming Dataset / State

Ticket Link → Upcoming Dataset / Ticket URL

If the repeater itself is linked to one dataset but the elements inside it are linked to another source, Wix may show the correct number of rows but repeat the same event content in every row.

This happened during the original setup and was corrected by relinking the repeater fields to the proper dataset.

6. Create the Past Events Repeater

Duplicate the Upcoming repeater or create another repeater.

Place it beneath the **Past Events** heading.

This second repeater should have the same field layout.

7. Create the Past Events Dataset

Create another dataset connected to the same `Concerts` CMS collection.

Recommended dataset ID:

`#pastDataset`

Configure its sort as:

`Date → Descending`

This causes the most recent past concert to appear first.

The architecture is therefore:

Concerts CMS Collection

|

+-- Upcoming Dataset

|

|

| +-- Upcoming Repeater

```
|  
+-- Past Dataset  
|  
+-- Past Repeater
```

Both datasets read from the same CMS collection.

No concert records are duplicated.

8. Connect the Past Repeater to the Past Dataset

As with the Upcoming repeater, verify that the repeater and all of its inner elements are connected to the **Past dataset**.

Example:

```
Past Repeater  
  Event Name → Past Dataset / Event Name  
  Date       → Past Dataset / Date  
  Venue      → Past Dataset / Venue  
  City       → Past Dataset / City  
  State      → Past Dataset / State  
  Ticket Link → Past Dataset / Ticket URL
```

During the original setup, this repeater also reverted to the CMS collection directly and had to be relinked to the Past dataset.

9. Enable Wix Dev Mode / Velo

The date-based filtering is performed using Wix Velo JavaScript.

In the Wix editor:

1. Open the Shows page.
2. Enable **Dev Mode**.
3. Open the page code editor.
4. Make sure the code being edited belongs to the Shows page.

The code editor normally contains something similar to:

```
$w.onReady(function () {  
  
});
```

10. Find the Dataset IDs

The JavaScript refers to datasets by their page element IDs.

Recommended IDs:

```
#upcomingDataset  
#pastDataset
```

These IDs can be viewed or renamed in the dataset settings or page connection properties.

The IDs used in the JavaScript must match the actual Wix dataset IDs exactly.

11. Find the Date Field ID

Velo code uses the CMS field ID, not necessarily the visible field name.

For example, the displayed field might be:

```
Date
```

while its field ID could be:

```
date
```

or:

```
eventDate
```

The field ID can be found from the CMS collection field properties or through the Dev Mode database/schema view.

Use the actual field ID in the Velo code.

12. Add the Automatic Upcoming/Past Filter Code

Add the following code to the Shows page.

Replace the dataset IDs or Date field ID if the actual Wix IDs differ.

```
import wixData from 'wix-data';

$w.onReady(function () {

  const today = new Date();

  // Set the comparison point to the beginning of today.
  // This keeps an event in the Upcoming list for its entire event date.
  today.setHours(0, 0, 0, 0);

  $w('#upcomingDataset').onReady(async () => {
    try {
      await $w('#upcomingDataset').setFilter(
        wixData.filter()
          .ge('date', today)
      );

      console.log("Upcoming filter applied");
    }
    catch (err) {
```



```

        console.error("Upcoming filter error:", err);
    }
});

$w('#pastDataset').onReady(async () => {
    try {
        await $w('#pastDataset').setFilter(
            wixData.filter()
                .lt('date', today)
        );

        console.log("Past filter applied");
    }
    catch (err) {
        console.error("Past filter error:", err);
    }
});
});

```

If the Date field ID is `eventDate`, change:

```
.ge('date', today)
```

to:

```
.ge('eventDate', today)
```

and likewise change:

```
.lt('date', today)
```

to:

```
.lt('eventDate', today)
```

13. How the Filtering Works

The Upcoming dataset receives:

```
Date >= Today
```

The Past dataset receives:

```
Date < Today
```

This means that an event remains in the Upcoming list for the entire day on which it occurs.

Beginning the following day, it automatically appears in the Past list.

The final behavior is equivalent to:

```
-- Upcoming events

SELECT *
FROM Concerts
WHERE Date >= Today
ORDER BY Date ASC;

-- Past events

SELECT *
FROM Concerts
WHERE Date < Today
ORDER BY Date DESC;
```

14. Test the Page

Before publishing:

1. Click **Preview**.
2. Open the Shows page.
3. Verify that Upcoming contains only current/future events.
4. Verify that Past contains only older events.
5. Verify that Upcoming is sorted oldest-to-newest.
6. Verify that Past is sorted newest-to-oldest.

If troubleshooting is necessary, open the browser developer console:

F12 → Console

Successful execution should show:

Upcoming filter applied

Past filter applied

15. Common Problem: Repeater Shows the Same Event Multiple Times

A repeater may display the correct number of entries but show the same event repeatedly.

For example:

5 repeater rows
but all 5 show the same concert

This usually means that the repeater is receiving five records correctly, but the text or link elements inside the repeater are bound to the wrong dataset or to a fixed literal value.

Check every element inside the repeater.

All Upcoming repeater fields should use:

Upcoming Dataset

All Past repeater fields should use:

Past Dataset

Do not rely only on the repeater's top-level connection.

Check the individual text and link elements as well.

16. Common Problem: Filter Does Not Appear to Work

If one repeater displays all concerts rather than only Past or Upcoming concerts, check:

1. The repeater is connected to the correct dataset.
 2. The elements inside the repeater are connected to the same dataset.
 3. The JavaScript dataset ID matches the actual Wix ID.
 4. The CMS Date field ID matches the ID used in JavaScript.
 5. The Date field is actually a Wix Date field.
 6. The page is being tested in Preview or on the published site.
-

17. Normal Maintenance Procedure

Once this setup is working, maintaining the page is simple.

To add a concert:

1. Open the **Concerts** CMS collection.
2. Add a new row.
3. Enter the event details.
4. Enter the event date.
5. Save/publish the CMS change as required by Wix.

The event will automatically appear under:

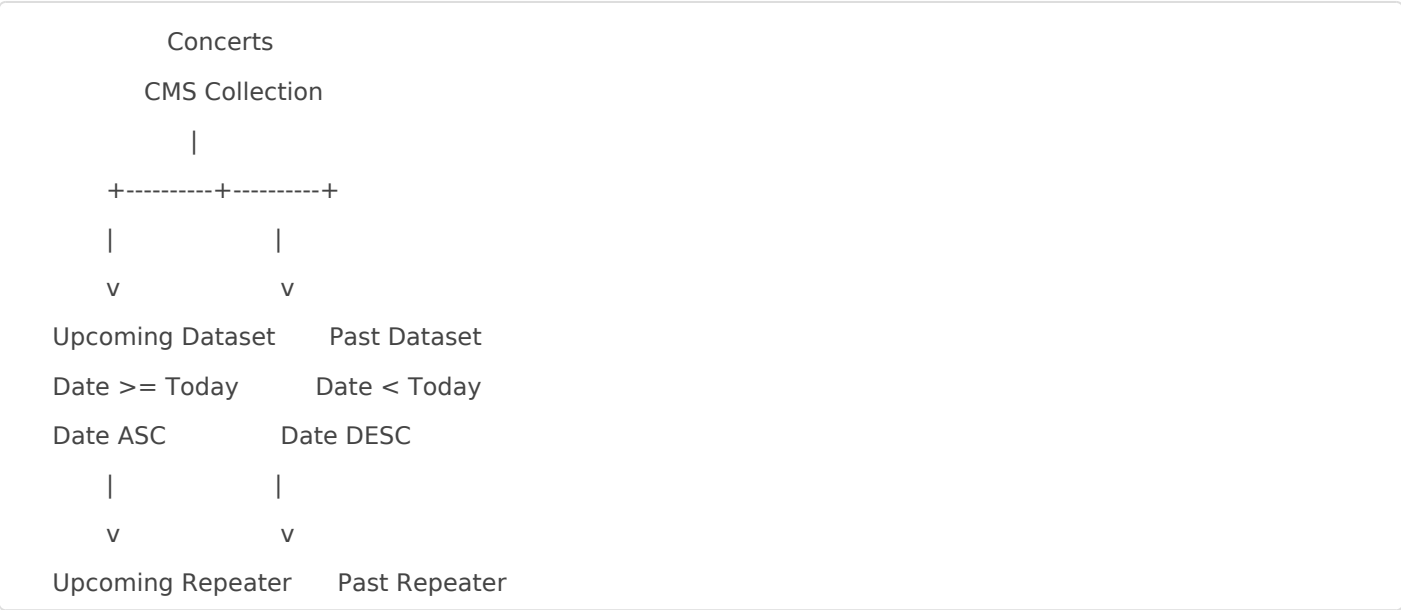
Upcoming Events

When the date passes, the event will automatically appear under:

Past Events

No page editing or manual event movement is required.

Final Architecture



This keeps the Wix CMS collection as the single source of truth while allowing the public Shows page to organize concerts automatically.

Revision #1
Created 10 August 2026 03:53:06 by glwhite
Updated 10 August 2026 03:53:37 by glwhite