Event Cards & Page

Content Setup Summary (what i have done on my own)

Events (Current + Upcoming)

- Use the **Activities** post type
- Add: **Title**, **Date**, **Location**, **Image (16:9)**
- Shows on: **Homepage cards** + **Events page**

Latest News

- Use regular **Posts**
- Add: **Title**, **Summary**, **Image (16:9)**
- Shows on: **Homepage news grid** + **News page**

Featured News (Hero Banner)

- Use a checkbox in News post to mark as "Featured"
- Add: **Large banner image (1920×740)**
- Shows at: **Top of homepage slider**

FGS Video

- Static embed — just paste YouTube or Vimeo link
- Shows in: **Homepage video section**

Reports / Guest Artists / Exhibitions

- Can be custom post types or News with a category
- Add: **Title**, **Image (4:3)**, **Description**
- Shows on: **Homepage cards** + **Gallery page**

Daily Quote from Venerable Master

- Store all quotes in a custom post type

- Add: **Quote text**, **Author (optional)**

- Shows in: **Homepage block that rotates daily**

Tea House Promo

- Static image + link

- Add: **Promo image (16:9)**

- Shows only on: **Homepage**

Rule of Thumb

- If it **changes often** → put it in the **database** using ACF

- If it **stays the same** → hardcode it as **static content**
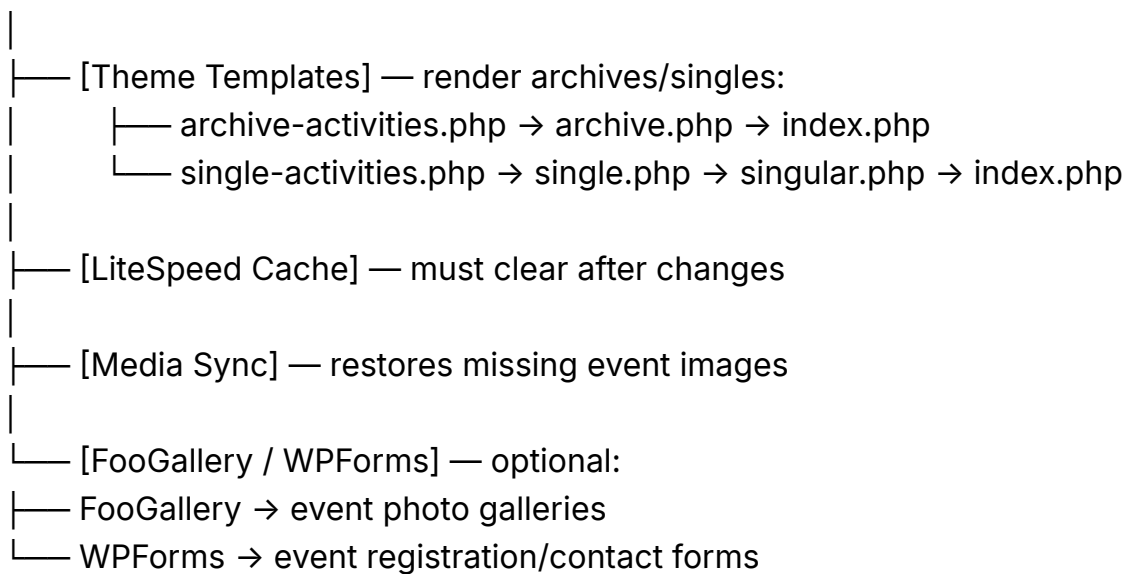
FGS Events - Visual Map

This helps me to show visually how all these parts connect. T

```
[CPT UI] — registers → [activities CPT ("Events")]
 |
├── Taxonomies: Categories + Tags (WP Core)
 |      |
 |      └── "featured-events" category → used for homepage/current events
grid
 |
├── [ACF] — adds custom fields:
 |      ├── event_priority (Number) → controls sort order
 |      ├── location (Text/Select)
 |      └── date/time fields (if present)
 |
├── [PostX] — displays Events in grids:
 |      ├── Query Settings → Post Type = activities
 |      ├── Filter by category = featured-events
 |      ├── Order by = event_priority (ASC)
 |      └── Styling → matches Figma card specs
 |
```

```
|
├── [Theme Templates] — render archives/singles:
│       ├── archive-activities.php → archive.php → index.php
│       └── single-activities.php → single.php → singular.php → index.php
│
├── [LiteSpeed Cache] — must clear after changes
│
├── [Media Sync] — restores missing event images
│
└── [FooGallery / WPForms] — optional:
├── FooGallery → event photo galleries
└── WPForms → event registration/contact forms
```

How to Use This Map

- **Need to add a new event?** → Add in activities CPT, assign featured-events category, set event_priority.

- **Need to change order?** → Edit event_priority in ACF.

- **Need to change layout?** → Edit PostX block settings or theme template.

- **Need to fix missing images?** → Use Media Sync.

- **Not seeing changes?** → Clear LiteSpeed cache.

(What Jai has reccomended, i went away and learned and did for myself)

Understanding the Goal here:

I want a reusable block in the WordPress editor that:

- Shows your latest events from the "Activities" CPT

- Automatically updates when new events are added

- Is styled like a card layout

- Can be dropped anywhere on a page

Prepping the Environment

Need:

- Access to WordPress site's file system (via Hostinger or FTP)

- A terminal or command line (or Hostinger's terminal if available)

- Node.js installed (for npx command)

If you don't have Node.js, you can still manually create the block folder—but npx makes it faster.

Scaffold the Block Plugin

This sets up all the files you need.

Run this command in your terminal:

bash

npx @wordpress/create-block event-card-block

This creates a folder called event-card-block inside your wp-content/plugins directory. Inside, you'll find:

File	Purpose
block.json	Metadata for your block
index.js	Editor-side code
render.php	Frontend rendering
style.css	Frontend styles
editor.css	Editor styles

🧠 STEP 4: Register the Block in PHP

Open render.php and add this function:

php

```php
function render_event_cards_block($attributes) {
    $events = get_posts(array(
        'post_type' ⇒ 'activities',
        'posts_per_page' ⇒ 5,
        'orderby' ⇒ 'meta_value',
        'meta_key' ⇒ 'event_date',
```

```php
        'order' ⇒ 'ASC',
    ));

    if (empty($events)) return '<p>No events found.</p>';

    $html = '<div class="event-cards">';
    foreach ($events as $event) {
        $html .= '<div class="event-card">';
        $html .= '<h3>' . esc_html($event→post_title) . '</h3>';
        $html .= '<p>' . esc_html(get_field('event_date', $event→ID)) . '</p>';
        $html .= '<a href="' . get_permalink($event→ID) . '">View Details</a>';
        $html .= '</div>';
    }
    $html .= '</div>';

    return $html;
}
```

Then register the block in your plugin's main PHP file:

php

```php
register_block_type(__DIR__, array(
    'render_callback' ⇒ 'render_event_cards_block',
));
```

🎨 STEP 5: Style It

In style.css, add:

css

```css
.event-cards {
  display: grid;
  gap: 20px;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
}

.event-card {
  border: 1px solid #ccc;
  padding: 16px;
```

```
  border-radius: 8px;
}
```

This gives you a clean, responsive card layout.

🧪 STEP 6: Activate and Test

1. Go to **Plugins > Installed Plugins**

2. Activate "Event Card Block"

3. In the editor, add your new block

4. It should show your latest events automatically

🧠 What You've Learned

- How to scaffold a block plugin

- How to use render_callback for dynamic content

- How to query CPT posts and display them

- How to style blocks with CSS