CMS & Frontend Updates Documentation

1️⃣ Media Center Module (News)

CMS Changes: - The `media` field in **News** now accepts **both images and videos**.
- Field configuration:
- Type: `Media`
- Allowed types: `Images + Videos`
- Multiple: `true`

Frontend Implementation: - Differentiate between images and videos when rendering.

Example (React/Vue):

```
news.media.map(item => {
  if (item.mime.startsWith("image")) {
    return <img src={item.url} />;
  }
  if (item.mime.startsWith("video")) {
    return (
      <video controls>
        <source src={item.url} />
      </video>
    );
  }
});
```

2️⃣ Projects Module – Multi-Select Project Types

CMS Changes: - `project_types` relation updated to **many-to-many**.
- Projects can now have **multiple types**.

Frontend Implementation: - Display all types:

```
project.project_types.map(type => (
  <span>{type.name}</span>
));
```

- Update filtering logic:

```
projects.filter(p =>
  p.project_types.some(t => t.slug === selectedType)
);
```

3️⃣ Download Brochure Field

CMS Changes: - Added `brochure` field in Projects.
- Field type: `Media`
- Allowed types: `PDF + Images`
- Required: `true`

Frontend Implementation: - Display a download button:

```
<a href={project.brochure.url} target="_blank">
  Download Brochure
</a>
```

- Optionally detect file type:

```
if (project.brochure.mime === "application/pdf") { ... }
```

4️⃣ Homepage – Partner Logos Section

CMS Changes: - Converted `logos` field into a **repeatable component**:

```
logos (component)
  ├ logo → Media
  └ project → Relation → Projects
```

Frontend Implementation: - Make logos clickable and link to the related project:

```
homepage.logos.map(item => (
  <a href={`/projects/${item.project.slug}`}>
    <img src={item.logo.url} />
  </a>
));
```

5️⃣ Privacy Policy Page

CMS Changes: - Created **Single Type: Privacy Policy**
- Fields:
- `header_image` → Media
- `title` → Text
- `content` → Rich Text

Frontend Implementation: - Fetch API: `GET /api/privacy-policy`
- Render:

```
<img src={data.header_image.url} />
<h1>{data.title}</h1>
<div dangerouslySetInnerHTML={{ __html: data.content }} />
```

6️⃣ Terms & Conditions Page

CMS Changes: - Created **Single Type: Terms & Conditions**
- Fields:
- `header_image` → Media
- `title` → Text
- `content` → Rich Text

Frontend Implementation: - Fetch API: `GET /api/terms-condition`
- Render same as Privacy Policy.

7️⃣ Site Settings – Google Analytics & Tag Manager

CMS Changes: - Created **Single Type: Site Settings**
- Fields:
- `google_analytics_id` → Text
- `google_tag_manager_id` → Text
- Optional: `custom_head_scripts` → Rich Text

Frontend Implementation: - Fetch IDs from API: `GET /api/site-setting`
- Dynamically inject scripts:

```
const ga = data.google_analytics_id;
const gtm = data.google_tag_manager_id;

// GA script
const gaScript = document.createElement("script");
gaScript.src = `https://www.googletagmanager.com/gtag/js?id=${ga}`;
gaScript.async = true;
document.head.appendChild(gaScript);

// GTM script
const gtmScript = document.createElement("script");
gtmScript.innerHTML = `
(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':
new Date().getTime(),event:'gtm.js'});
var f=d.getElementsByTagName(s)[0],
j=d.createElement(s),dl=l!='dataLayer'?'&l='+l:'';
j.async=true;
j.src='https://www.googletagmanager.com/gtm.js?id='+i+dl;
f.parentNode.insertBefore(j,f);
})(window,document,'script','dataLayer','${gtm}');
`;
document.head.appendChild(gtmScript);
```