APOLLO

onecms

Inception of Information & Data Centralisation

OneCMS was built to solve the problem of creating/ updating content pages in upGrad's ecosystem

Time degree to publish one content page

Different countries entities have to be created separately, which is repetitive

Moving to consolidate all course platforms on the same page

System has more than only content stress because instructors's course

There is no right base to upload all the newly created

What was the real problem?

As upGrad was increasing its presence in various regions, it was difficult managing different demographic relations and creating program pages at an exponential scale

80+ Landing program pages

60% Page content has created replicas

12 days To launch a new program page

Why do we need it?

Of all the problems we had after discussion and observation, here are the three main ones which we wanted to solve for

1. **Making outsourcing easy**
   We wanted to put the content creation/updation data load on the university offering the course

2. **Managing load of big data chunks**
   We wanted our system to support the data of 200+ content pages by end of 2022

3. **Creation of content approval**
   We wanted to cut loose all the informal channels of content approval and version management

Who were we building it for?

There were essentially three roles in the system that powered the entire solution

Moderators
The set of users that are responsible for creating templates and approving requests from creators

Inhouse creators
upGrad's content creators that use templates created by Moderators to make pages

Outsourced creators
The college/course specific content creators which act as proxy for inhouse creators

How did we solve it?

The hotspot of solving an expensive solution **was solving this problem through inheritance**, the most basic principle of programming.

Inheritance

Templates
We clearly envisaged the information pool and created blueprints on the basis of domains

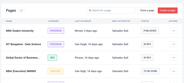
Pages
A page is merely a child entity of the template which reflects course specific information

The child entities are also exclusive and can be altered individually

Parent entity can have manipulations done that affected all the children

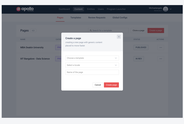
A few visuals of the solution

The visuals were not a major part of the solutioning as we were using Ant design system for this project already.



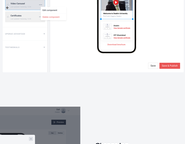
Authority driven
The entire solutioning was done in a way to clearly mark the next step in the course of action

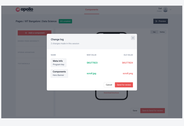
Creation of a page
The content page can be pulled from the predefined template and can be fed with course specific content



Creation of a template
A boiler plate for different domain pages were created to easily create child pages from the same

Live Preview
The current solution also included missing live rendering of the page being formed





Change log
The content pages being so information heavy, we above a change log to tell the user as to what was change and can be sent to review

What metrics did we move?

The solution was rolled out to the internal teams and eventually to the college for creating those page. The new numbers were very promising

12 Landing Templates pages
↗ 50%

10% Landing Pages that created replicas
↗ 80%

6 hrs Landing page from scratch
↗ 37%

What we couldn't solve?

- Despite creating a solution which was leaps and bounds ahead, we still couldn't trim down the number of entry items required to create a content page
- Content creators were using CSV and Excel for adding bulk data as it was easier for them, we were still struggling with the friction of bringing everyone to this platform
- We couldnt simulate iOS and Android environment on web as there were plenty tech constraints

We went a step further!

OneCMS is packed full of features that help you save time and work more effectively. Of all the new features introduced, here are few listed

Shared library
A pool of shared resources which is common for all the users onboarded. Reduced multiple

Multiple updation
OneCMS gives a new feature of adding or updating multiple components at the same time

Permission Driven
The entire canvas is permission driven, that means only the respective users have