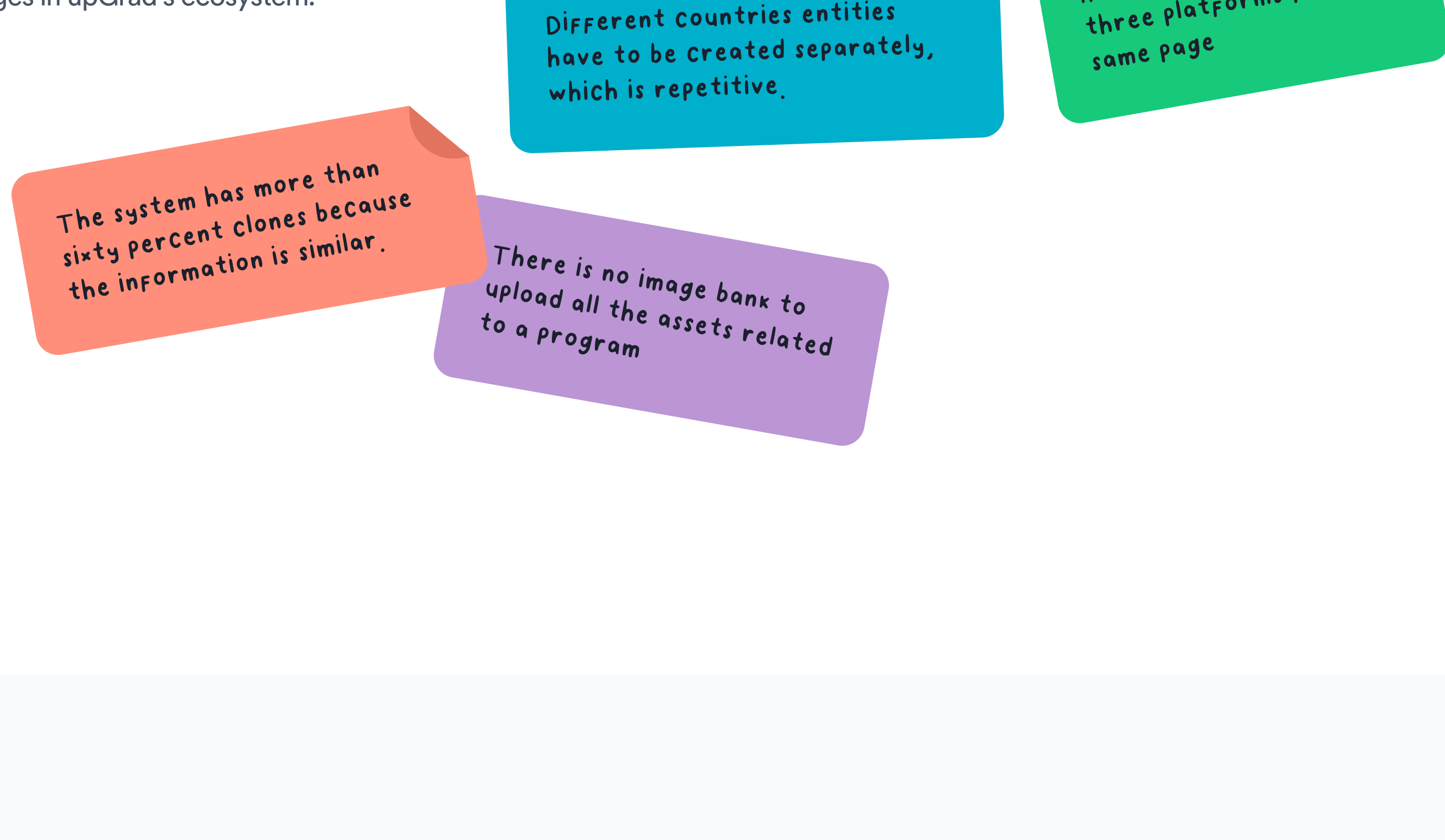


Inception of Information & Data Centralization

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



What was the real problem?

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

80+

unique program pages

60%

page clones that create 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/update data load on the university offering the course.

2 Managing the load of big data chunks

We wanted our system to support the data of 200+ content pages by the 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 creator that uses 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?

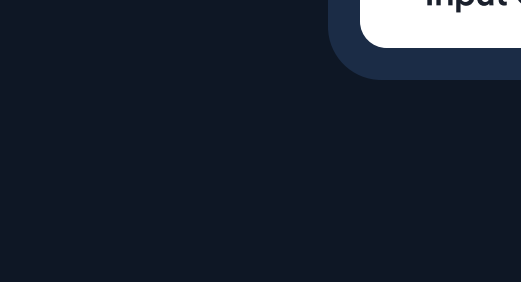
The holy grail of solving an expensive solution was solving this problem through inheritance, the most basic principle of programming.

Inheritance



Templates

We closely observed the information pool and created blueprints on the basis of domains.

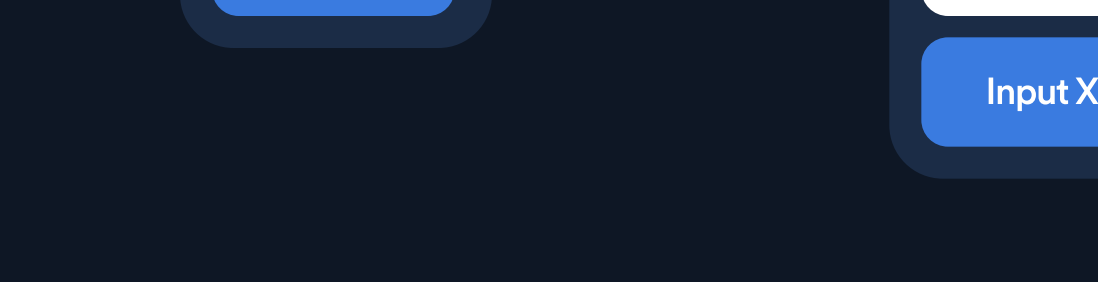


The child entities are also exclusive and cannot be altered individually.

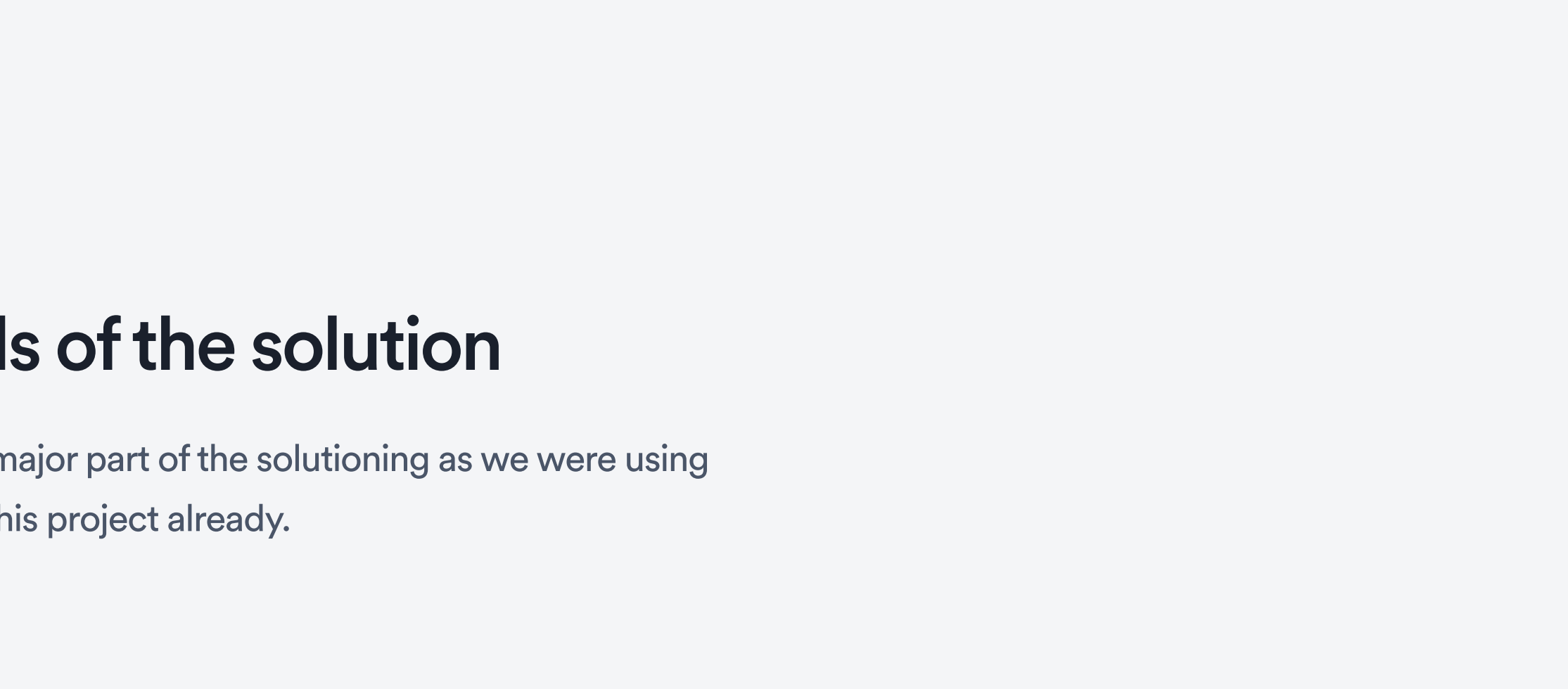


Pages

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



Parent entities can have manipulations done that affect 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.

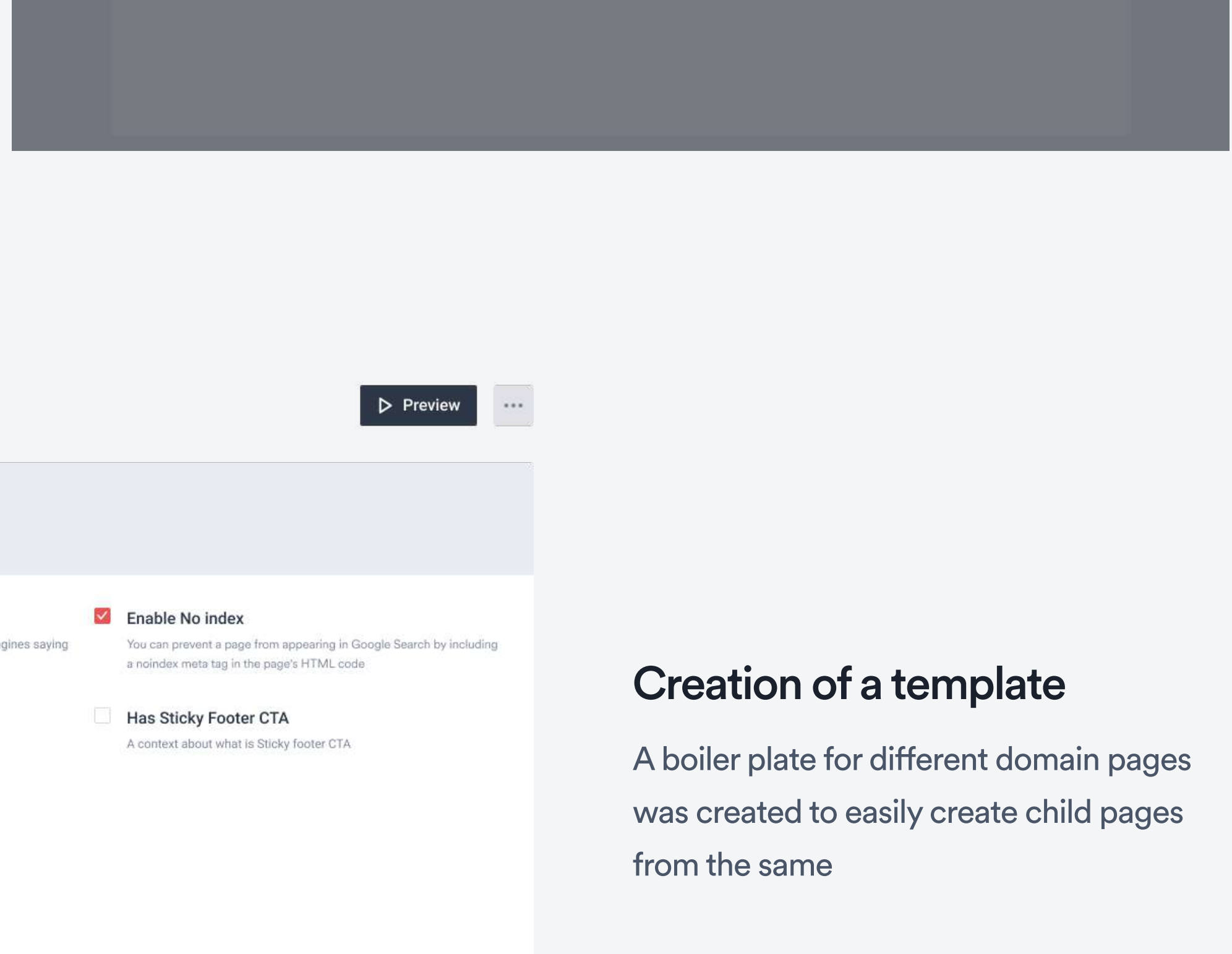
NAME	CATEGORY	LAST ACTION BY	NEXT ACTION FOR	STATUS	ACTIONS
MBA Deskin University	PROGRAM	Monet, 3 days ago	Salvador Dali	PUBLISHED	...
IIT Bangalore - Data Science	PROGRAM	Van Gogh, 16 days ago	Salvador Dali	IN REV	...
Global Doctor of Business...	SEO	Picasso, 16 days ago	Salvador Dali	IN REV	...
MBA (Executive) MMIMS	CONTENT	Van Gogh, 16 days ago	Salvador Dali	TO BE DONE	...
Finance and ML-AI	PROGRAM	Van Gogh, 16 days ago	Salvador Dali	IN REV	...
BITM MBA Masera	SEO	Van Gogh, 16 days ago	Salvador Dali	PUBLISHED	...

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.

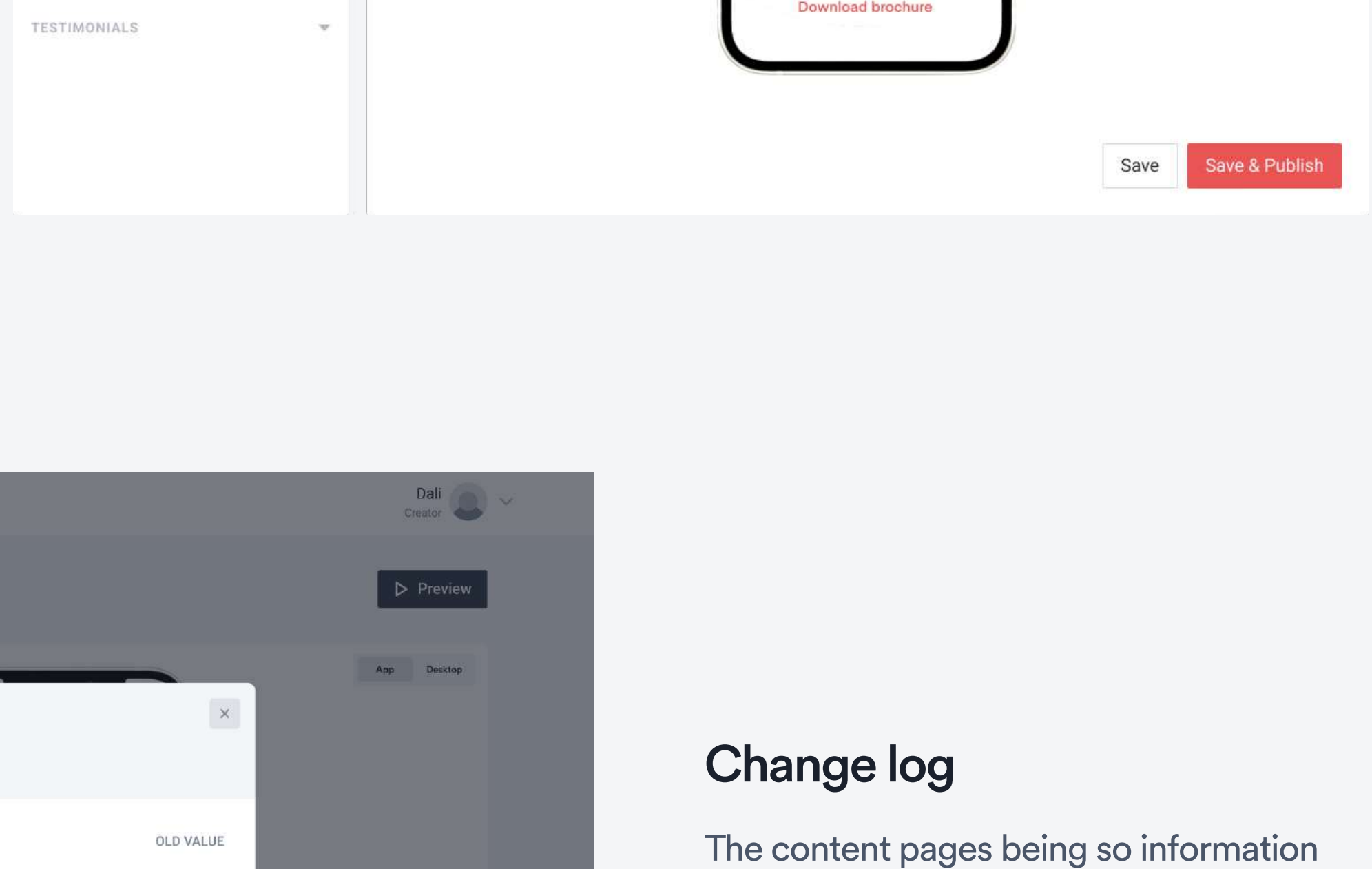


Creation of a template

A boiler plate for different domain pages was created to easily create child pages from the same

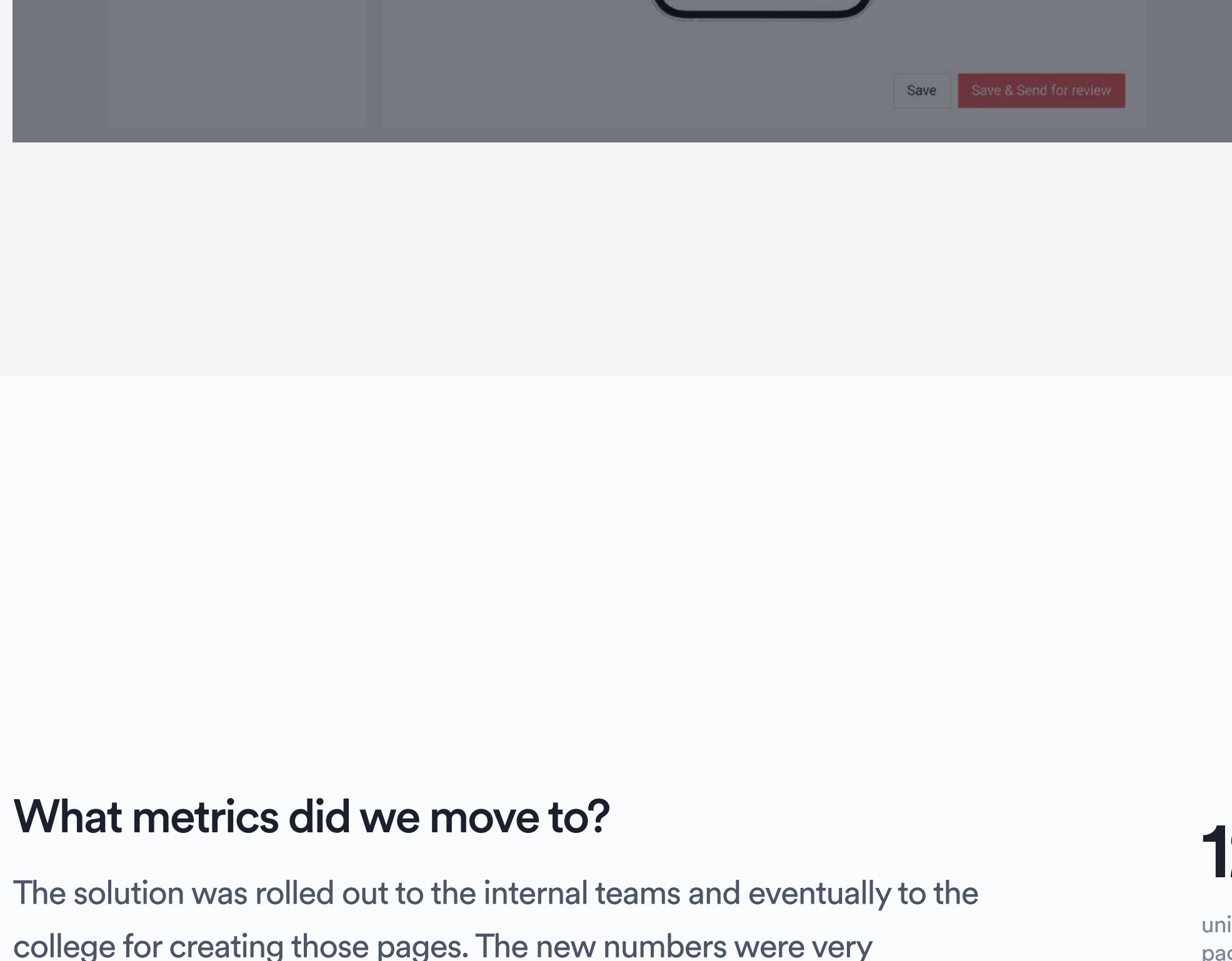
Live Preview

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



Change log

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



What metrics did we move to?

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

12

unique template pages

✓ 85%

10%

page clones that create replicas

✓ 82%

6 hrs

creating a page from scratch

✓ 97%

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