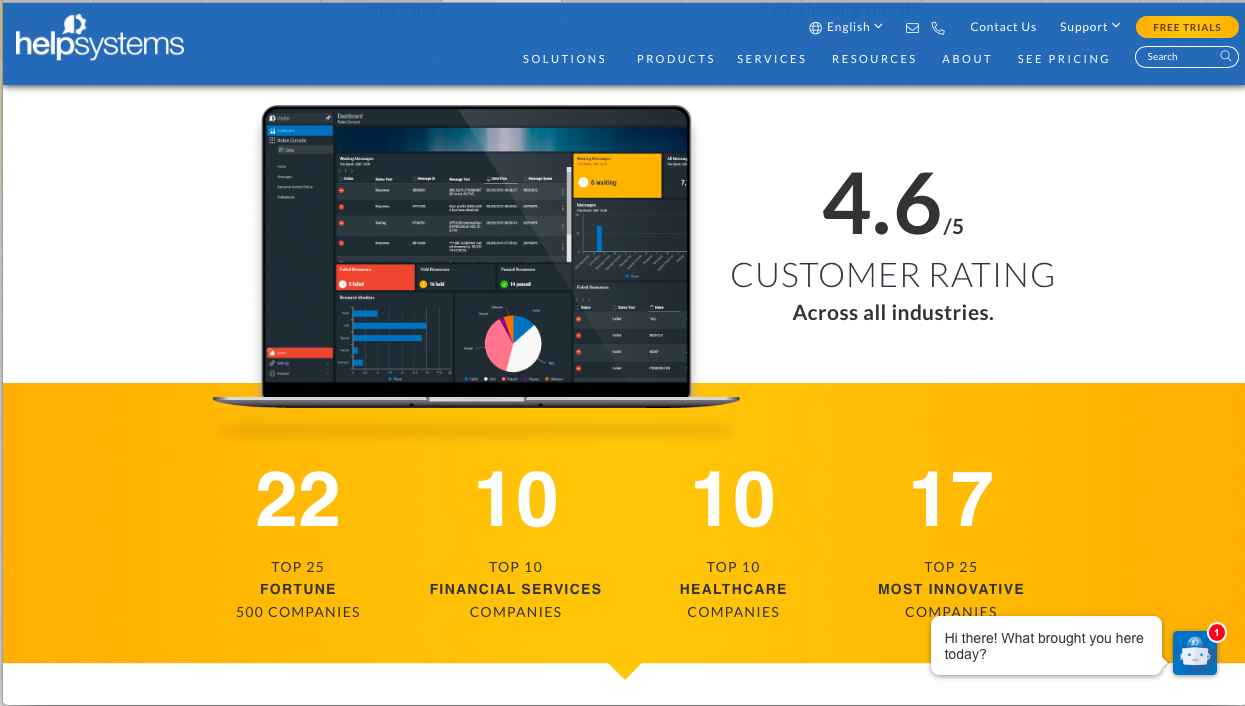
**Pamela Lies: Drupal Development Examples**

**https://www.helpsystems.com**

[](https://www.helpsystems.com)

The HelpSystems website is currently a Drupal 7 site, with work underway to migrate the site to Drupal 8. I was the tech lead in the design and development of this site (which combined several disparate product sites on different platforms into one over-arching corporate site), and I have had a hand in the development of virtually every aspect of the site as well.

HelpSystems is primarily a business-to-business, back-office software company. The software products are for the most part quite complex and expensive, so there is very little in the way of eCommerce on the site. The site’s main purpose then, is lead generation and sales nurturing; helpsystems.com drives more than 85% of the company’s sales.

The HelpSystems site is far more than a brochure-ware site, however, and many of the company’s most crucial business functions are fulfilled by the website.

A few examples of the business challenges solved with the development projects I’ve completed on this website are:

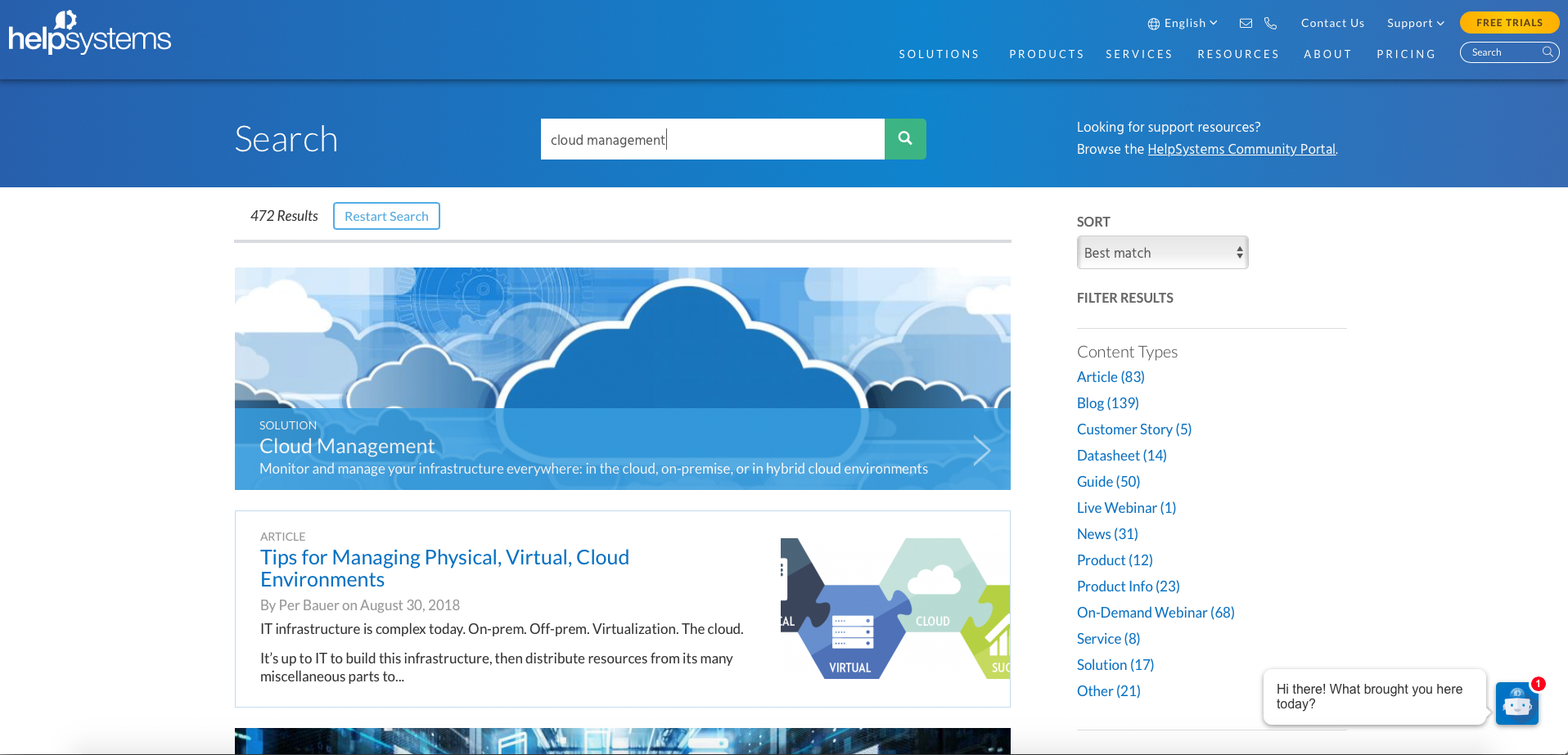
* **Content flexibility, but with enforced consistency**: our old websites had very little in the way of a consistent look or style across the board. With this site, we wanted to give our content creators enough flexibility to be creative with their content, but yet maintain a consistent look across all areas of the site. We accomplished this by using custom content types, and several different types of Drupal Paragraphs available within each content type. We locked down permissions more tightly so that editors aren’t able to create content that diverges from our corporate style guide. One simple example of this, for instance, is when creating a button – editors are able to choose the button’s color from a dropdown containing a prescribed palette of corporate colors, and the site-wide CSS takes care of styling it with the right look and size for each particular area of the site. This effort keeps content consistent yet somewhat flexible across all areas of the site, despite having many different content contributors.
* **Engage and educate our audience, establish industry authority and trust.** We had a lot of great content on our old sites that needed to be cleaned up and consolidated so we could showcase our industry authority and increase our audience involvement. We used Disqus to create an easily-searchable blog that also drives audience engagement. All this content is also setup to be easily shared via email or social media.

**e.g.:** <https://www.helpsystems.com/blog/how-goanywhere-mft-can-support-you-during-covid-19>



* **Improved search functionality.** We could tell from our site analytics that our old sitesdid a very poor job of delivering good search results to our users. On our new site we were able to integrate Solr to deliver much more targeted and faceted results, and this success is now reflected in our site analytics.

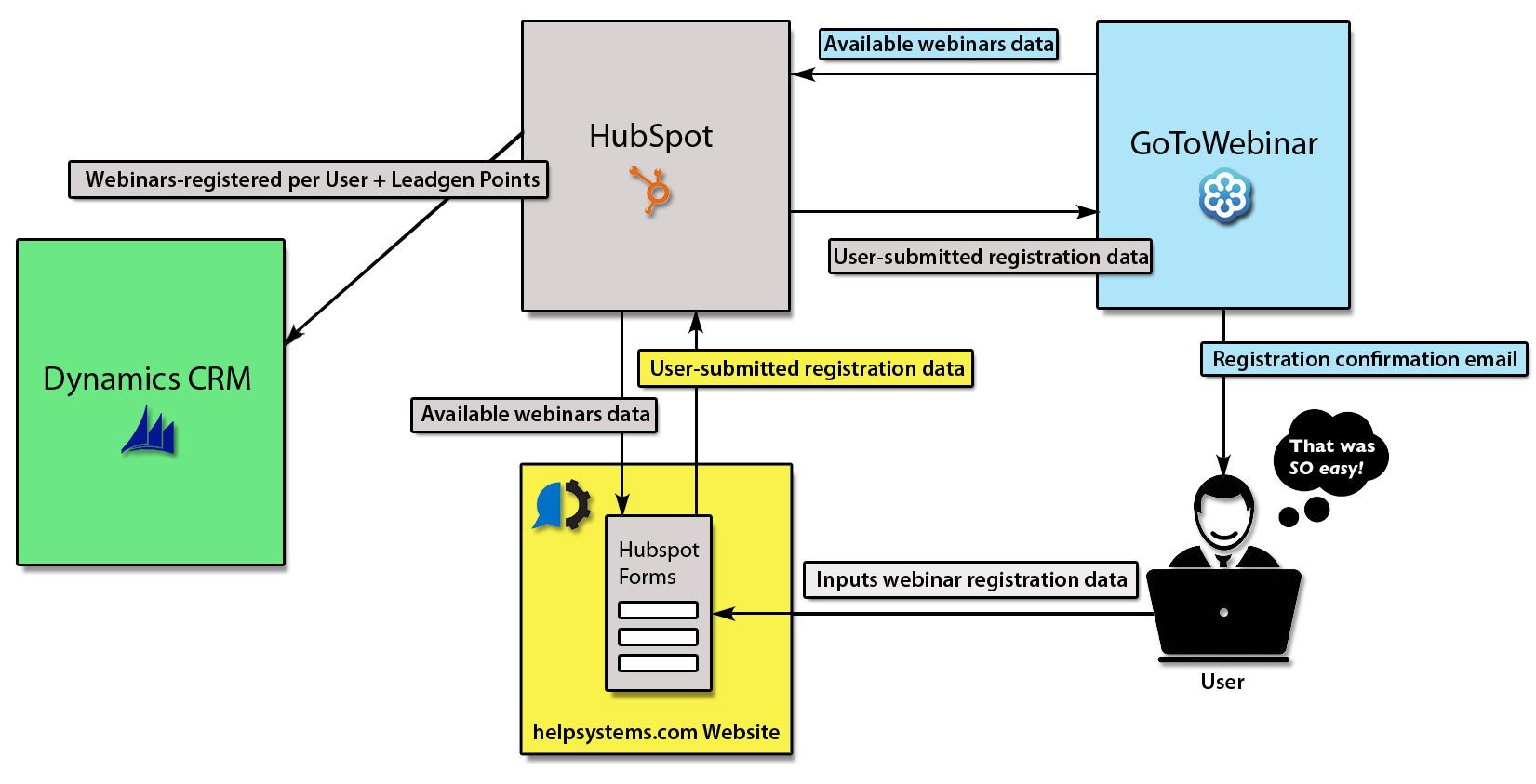
**e.**g.: <https://www.helpsystems.com/search?keywords=cloud>



* **Streamlined Webinar registration process.** HelpSystems conducts *a lot* of online webinars, and those webinars help drive a lot of business for the company as well. Our existing webinar registration process involves including a registration button on each live webinar page that when clicked, takes the registrant off our site and over to gotowebinar.com to register (you can see a live example of this existing process here: <https://www.helpsystems.com/resources/webinars/two-faces-sql-ibm-i>).

Our biggest pain points with this process were that we were losing a lot of traffic from our website when users clicked away to register for a webinar, and that there was a lot of manual work required both to set up ongoing webinar pages, and to pull the webinar registrant data back into our CRM.

In response to these problems, I created a custom Drupal module that takes much of the manual work out of the whole process, and has numerous other benefits as well. The diagram below outlines this new process at a very high level. The new process, which is currently in test on our staging server and will be live soon, instead uses a HubSpot form on each webinar registration page (that is easily chosen from a drop-down when setting up the page). Once submitted, the user’s registration data flows seamlessly to HubSpot, which passes it on to GoToWebinar (GTW) via their API. As a result of this single form submission on our site, the user is registered for their selected webinar(s), the success page they’re directed to keeps them on the helpsystems.com website, and they receive an email confirmation of their webinar registration straight to their inbox from GTW. Additionally, the user’s registration data as well as their accompanying lead-generation points flow automatically from their HubSpot form submission into their customer record in our Dynamics CRM. The last piece of this new solution includes generation of a dynamic XML file of all currently available webinars on GTW. This file is pulled back to HubSpot daily, and then parsed by code in my new module, which allows us to create standing registration pages that will automatically include all webinars with a selected topic, or a series of all webinars with a specified title.



Some of the major benefits resulting from this new registration process are:

•    Retained traffic on our site when users register for webinars – registrants now remain on our website.

•  The new process provides a much more seamless user experience for registrants.

•    Marketers don’t need to spend large amounts of time constantly adding new webinars to the site.

•    Increased webinar attendance by presenting users with related webinars in a series or program – with simultaneous registration.

•    Users now need only fill out one form to register for multiple webinars, not one form for each webinar.

•    “Standing pages” automatically pull in the latest of ongoing webinars parsed from an XML file from GTW, saving a lot of setup time.

•    We gain additional analytics by using HubSpot forms for registration.

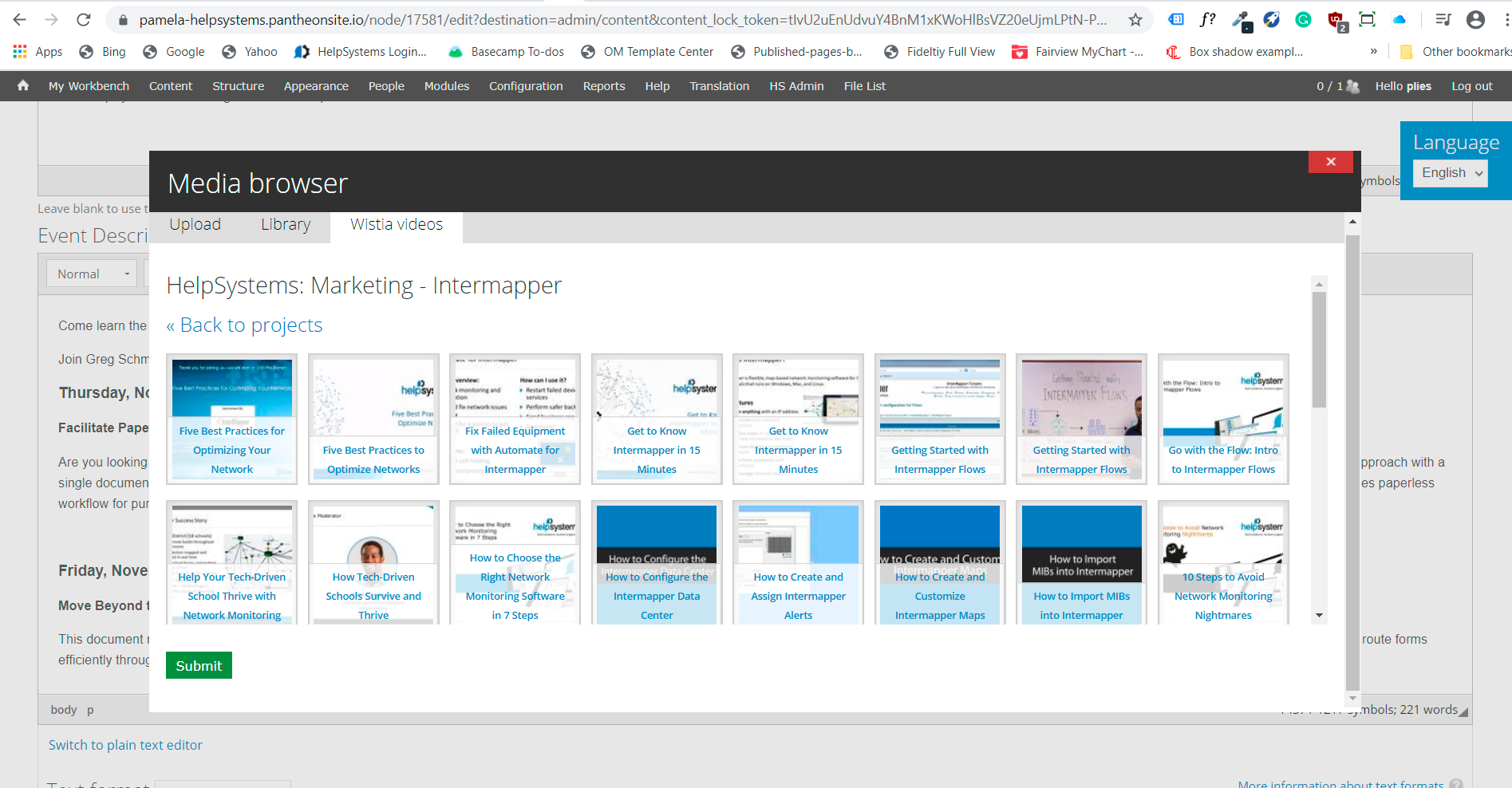
•    Webinar registration data is written directly into each customer record in Dynamics CRM and automatically contributes to their lead-generation points without the need to manually import data from GTW.

•  Our content people currently testing the new process are super happy with it! One literally did handsprings down the hall. ☺

* **Easily incorporate multilingual content.** As our business has expanded, it is our goal to provide as much of our content as possible in several different languages. We currently offer five different languages on the site, and Portuguese and Japanese are in the works. With our new site, we needed a way to streamline the process of offering and adding new multilingual content. We were able to use the Drupal Internationalization modules to make this process fairly easy, and the “TMGMT” (or Translation Management Tool) module to make submission of new content to our translation provider super streamlined. Content editors can just check a “Submit for Translation” box on any page (and specify desired languages), and that page’s content is automatically sent over to our translation provider behind-the-scenes. When the new translation is ready – it’s automatically available for review and approval within the site and the editor is notified.

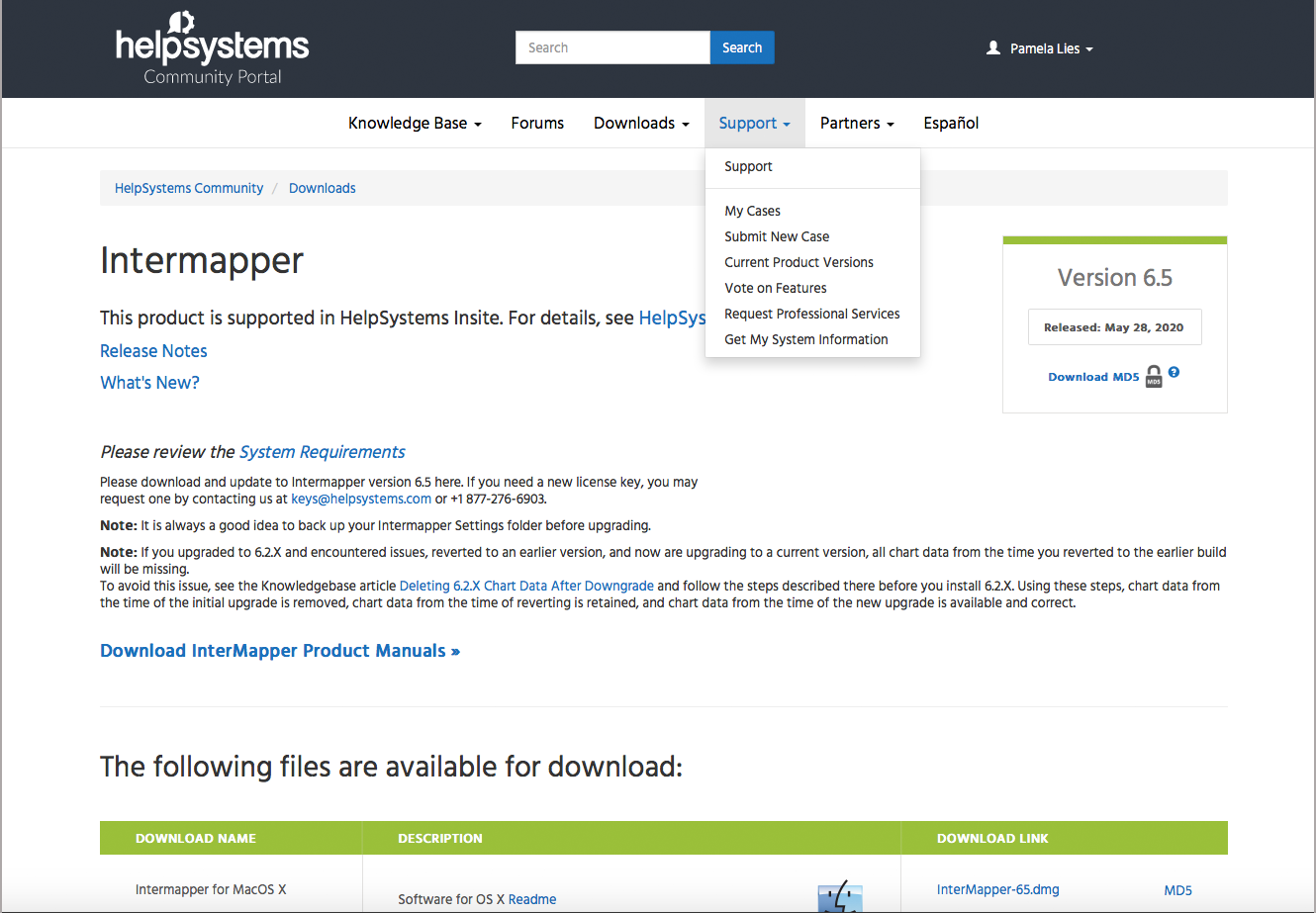
**e.g**.: <https://www.helpsystems.com/fr/solutions/automatisation>

* **Automated recorded webinar selection and insertion.** In addition to the many live webinars HelpSystems conducts, we have hundreds of recorded webinars available on the site for on-demand viewing as well (<https://www.helpsystems.com/resources?f%5B%5D=type:recorded_webinar>). To place a webinar on the site previously, editors had to login to Wistia, find the video they wanted, retrieve the webinar embed code, and manually insert it into the HTML of a page. This process was time-consuming, error-prone, and often required the help of a developer to get it right. The new code I wrote interfaces with the Wistia API behind-the-scenes, and then presents all of the recorded webinars in our Wistia account (organized alphabetically within each Wistia project) to choose on a new “Wistia videos” tab of the familiar Media browser we were already using to place images on the page (see screenshot below). Now all an editor has to do is choose the video they want to display from their media browser and hit “Submit” – and the webinar code is placed on the page for them – no developer intervention required! This has saved a lot of time and frustration on the part of both editors and developers alike.

****

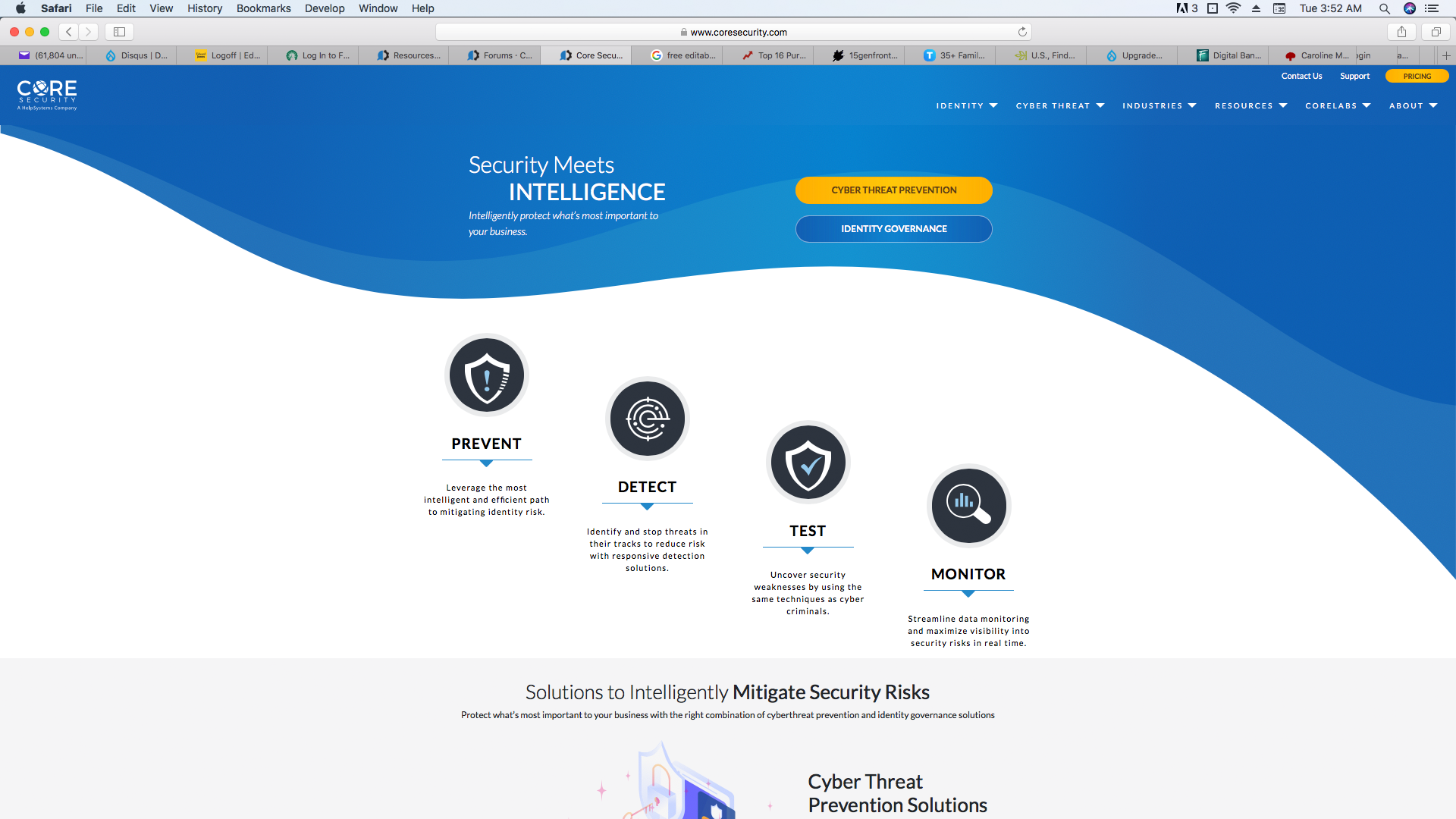
Incidentally, there already did exist a “Media – Wistia” module for Drupal 7 that was supposed to provide this functionality, but it didn’t work with the most recent (at the time) version of the Media module – which we needed to run on our site for other purposes. So the code I wrote consisted of a patch to enable the “Media – Wistia” module to work correctly with the 7.x-2.x branch of the Media module. When I finished my development, I rolled a patch for it and provided it back to the Drupal.org community for other people to use as well.

* **Convenience and service for our existing customers.** While the majority of the HelpSystems site is aimed at driving new sales, we needed to have an authenticated area of the site to make it easy for existing customers to fulfill their needs as well. The Community Portal section of our site allows existing customers to download the latest versions of their software and user’s manuals, create and monitor support cases, and easily find answers to their questions via the Knowledge Base and User Forums. This section of the site is tied in with our back-end systems as well so that only products and versions a custom has purchased are available for them to download when they log in. When a customer purchases a new product, it’s available for them to download as soon as their invoice clears. The look of this part of the site coordinates with the main site, but has a bit more of a utilitarian look, so existing customers can get right to what they need quickly.



Of note on this project, the development was initially set to be done by a developer hired by our IT group just for the purpose of creating our customer (and later also partner) portal. When that developer suddenly left to join a startup, they handed me the project and said “here Pamela, you can do this now”. They had already chosen to create the portal using Zend Framework – as it was argued to interface most easily with our back-end systems. I had no experience building with Zend at the time, but it was deemed too late to start the project with a different framework. Fortunately I was able to dive in and pick up Zend development pretty quickly – and completed the project successfully and on time! Just wanted to note that I’m good at jumping in and learning what I need to know on my own, in order to complete whatever work is needed!

[**https://www.coresecurity.com**](https://www.coresecurity.com)



This site doesn’t have as many interesting applications as our main corporate site – it was primarily created to educate users and demonstrate our industry authority and products in the Cybersecurity space. I mainly included it here as an example of a site for which we’ve already completed the work of migrating the site from Drupal 7 to Drupal 8.

The site does make extensive use of Drupal Views to present important Cybersecurity-related information that is widely used by many people. Some examples of these types of information can be found on pages describing recent exploits (<https://www.coresecurity.com/core-labs/exploits>), current threat advisories (<https://www.coresecurity.com/core-labs/advisories>), and threat mitigation publications (<https://www.coresecurity.com/core-labs/publications>).

**In Conclusion:**

This is just a small subset of the many interesting development projects I’ve completed recently, chosen to hopefully demonstrate a wide variety of my application work. There are many others I could provide descriptions for if needed!

I was asked to also provide answers to the following two questions for each project:

* What did you enjoy about the project?
* What would you do differently if you could do it over?

I didn’t provide separate answers to these questions for all projects because by and large the answers are the same. The thing I love the most about my job is diving in and figuring out how to solve new development problems every day! And for the most part there isn’t much I would choose to do differently, because I attempt to come up with the best plan of attack before jumping in and starting actual development.