

eCommerce using the JAM stack on static websites

Atishay Jain

The web development landscape



Image by saint on [pixabay](#)

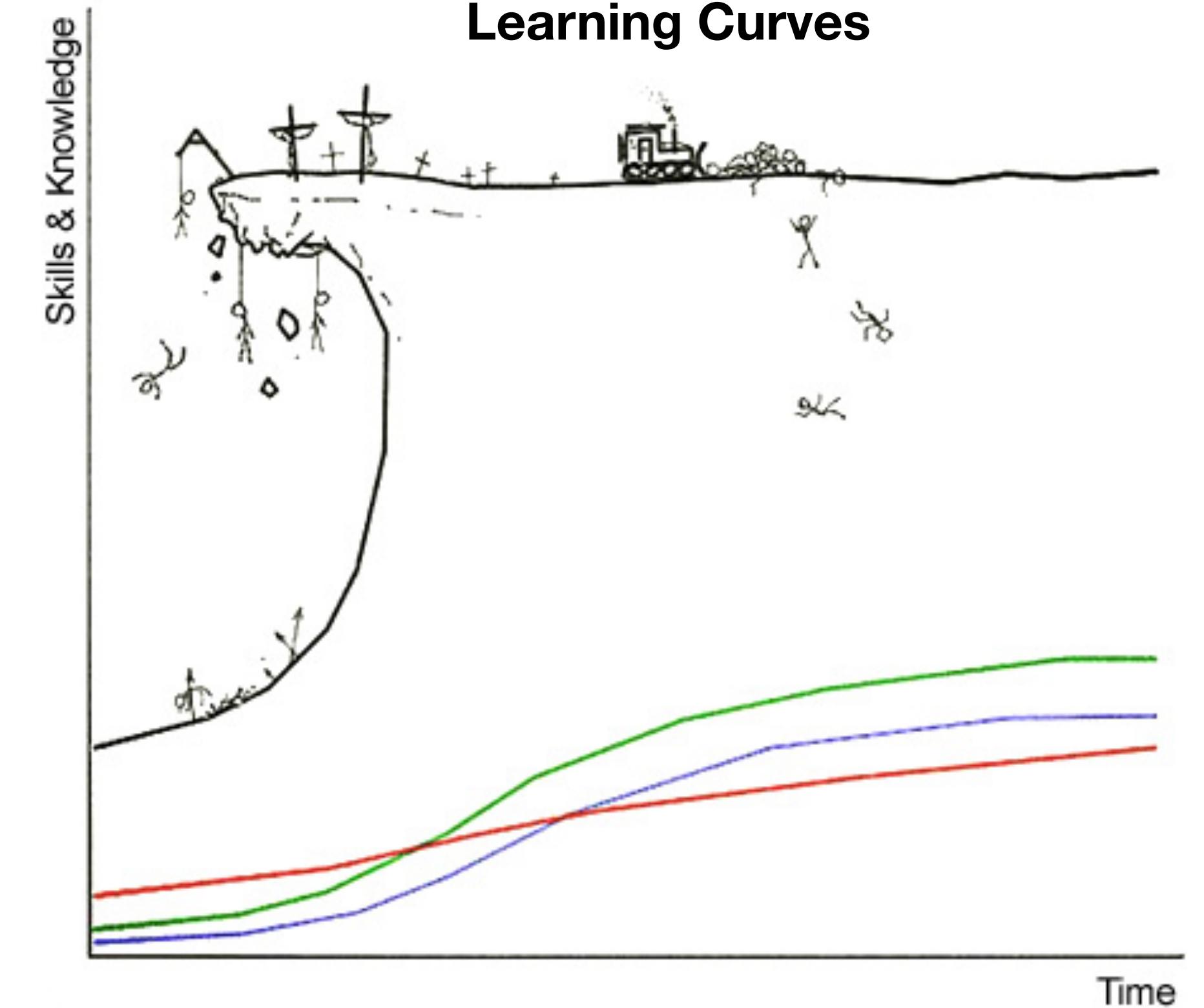
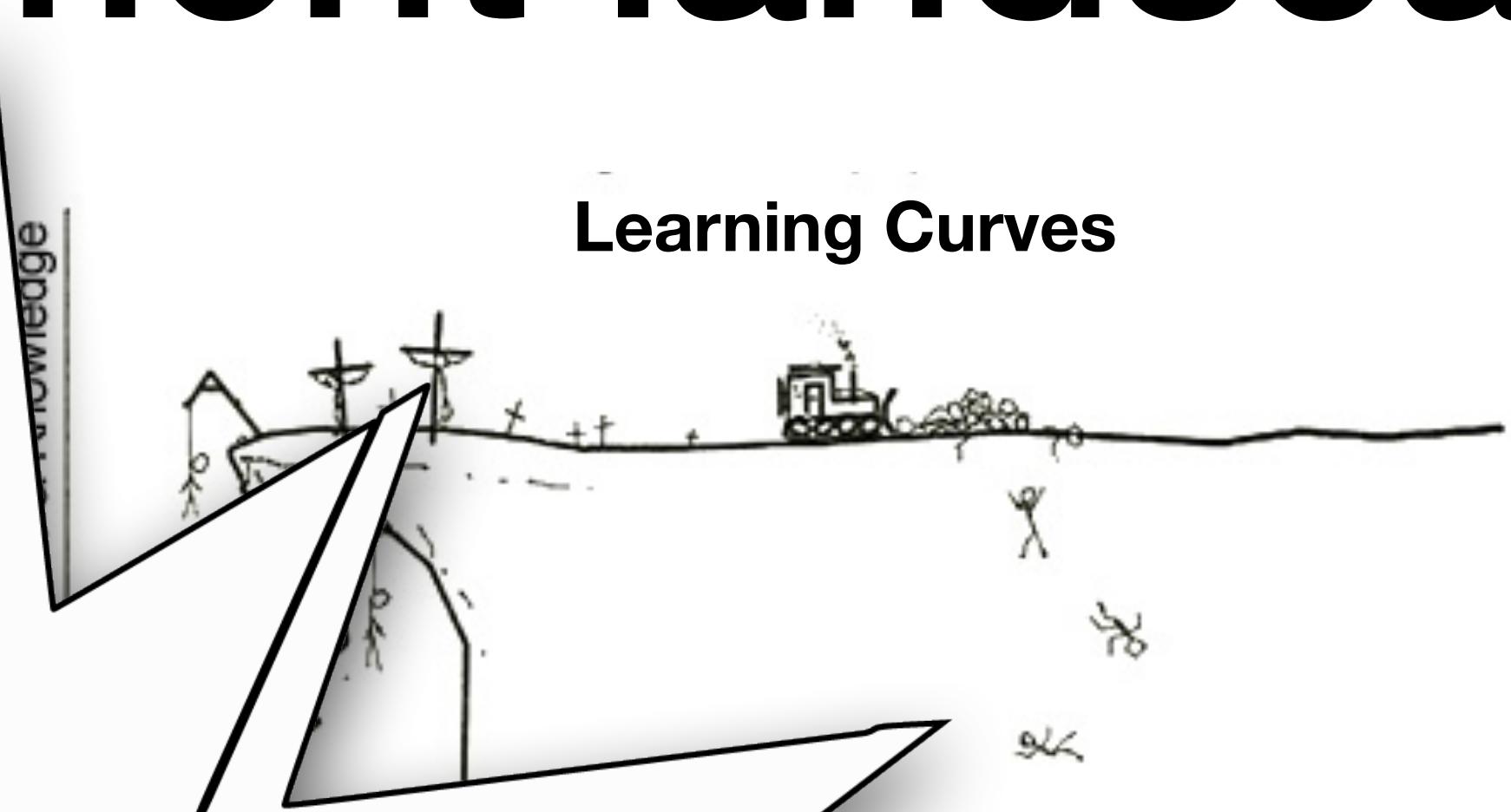


Image cropped from the [learning curve for CMS](#)

The web development landscape



JAM^{STACK}



Learning Curves

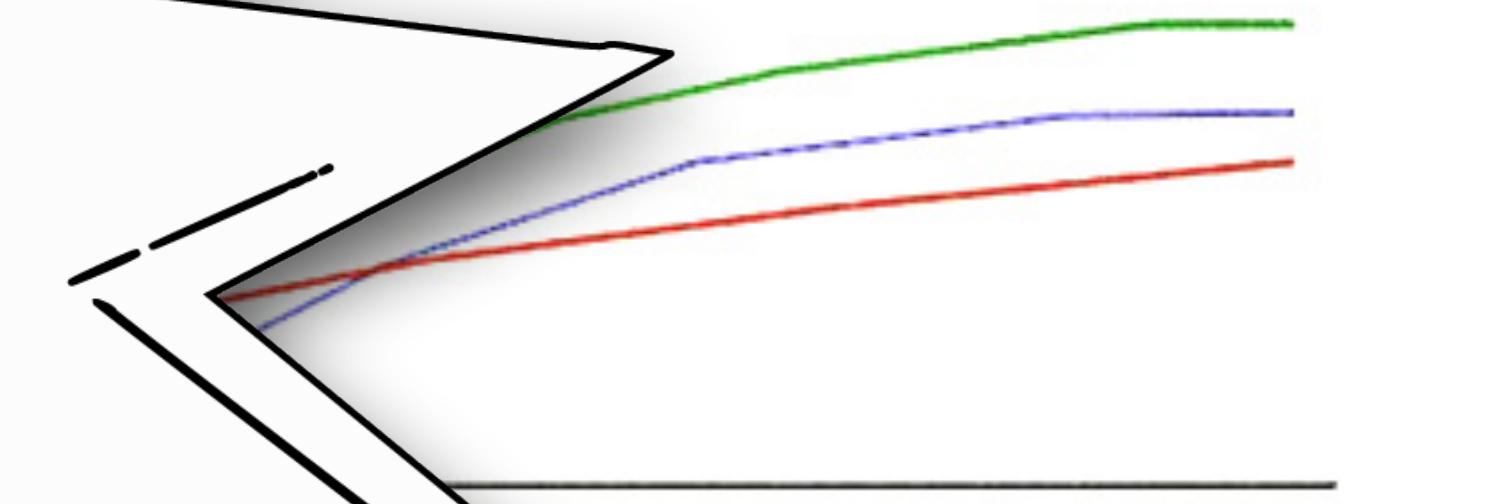


Image by

iXabay

Image

from the [learning curve for CMS](#)

Contents

- My experiments with static websites.
- Should you use the JAM stack for eCommerce?
- How can you use the JAM stack for eCommerce?
- It won't JAM your stack

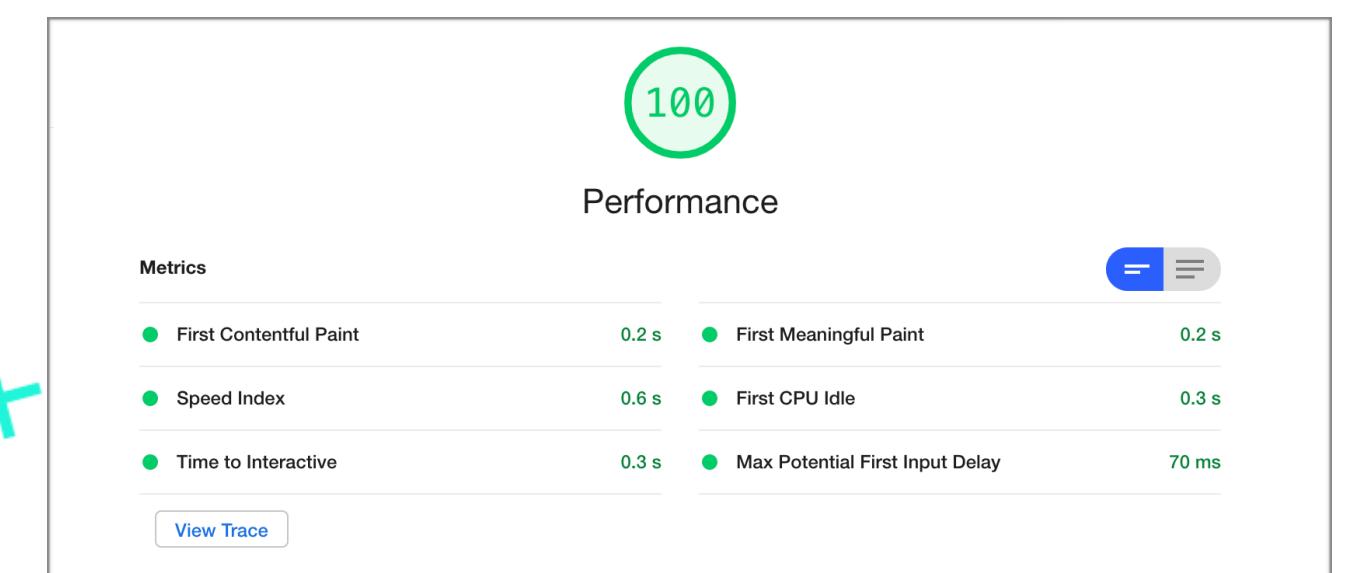
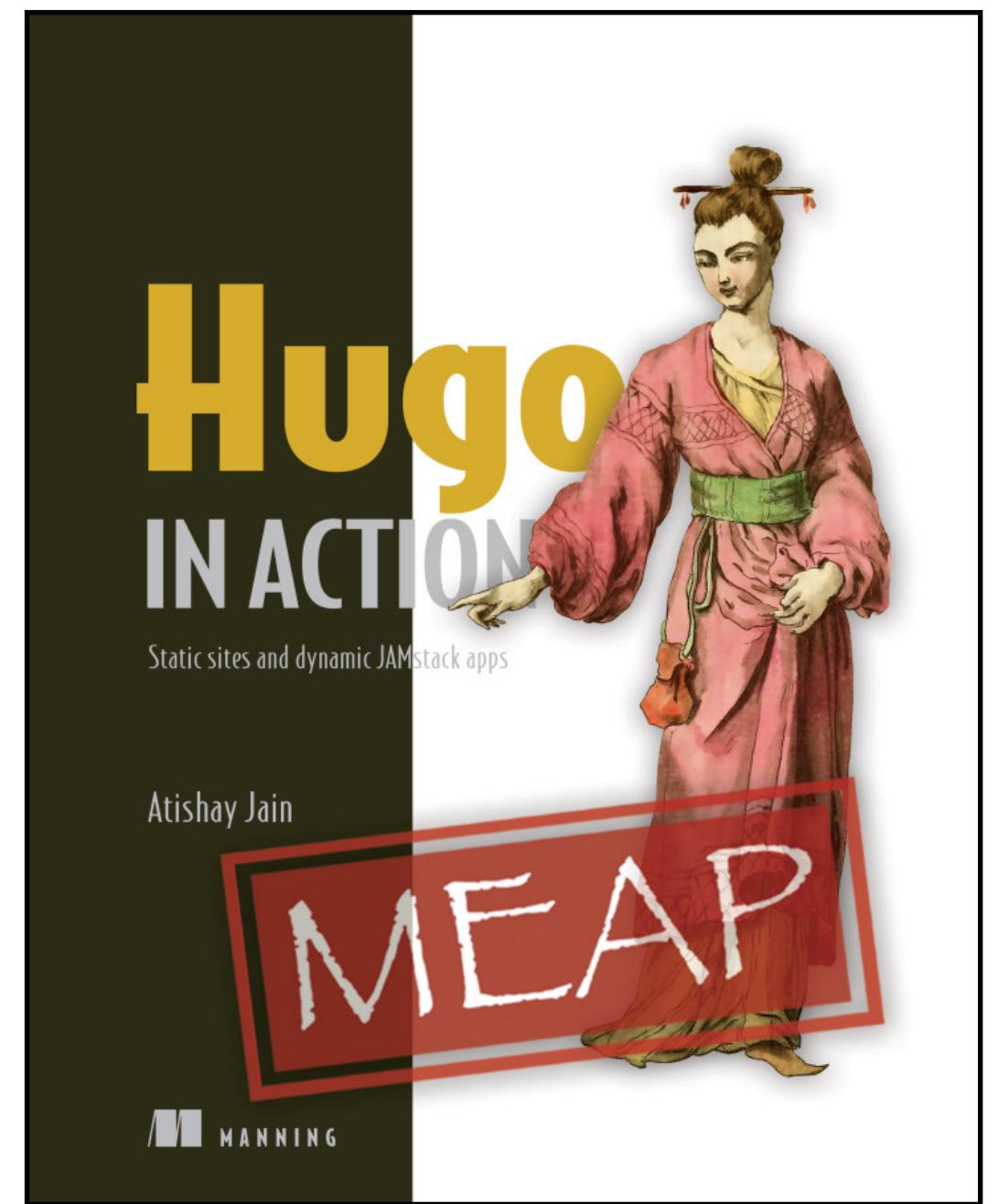
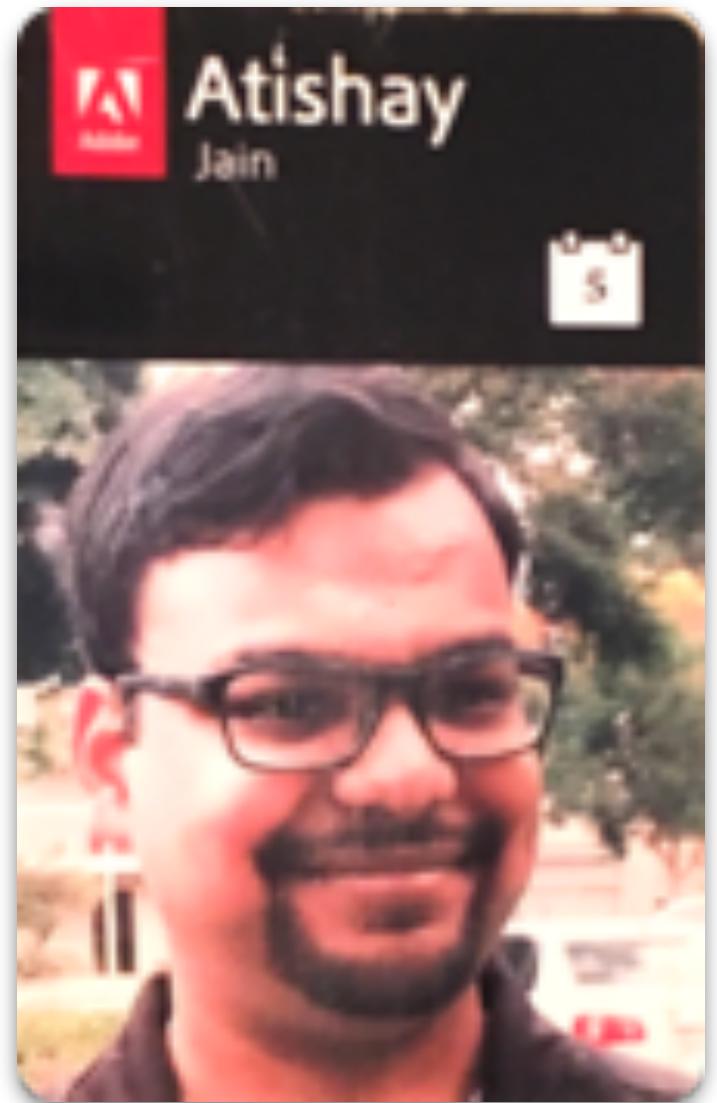
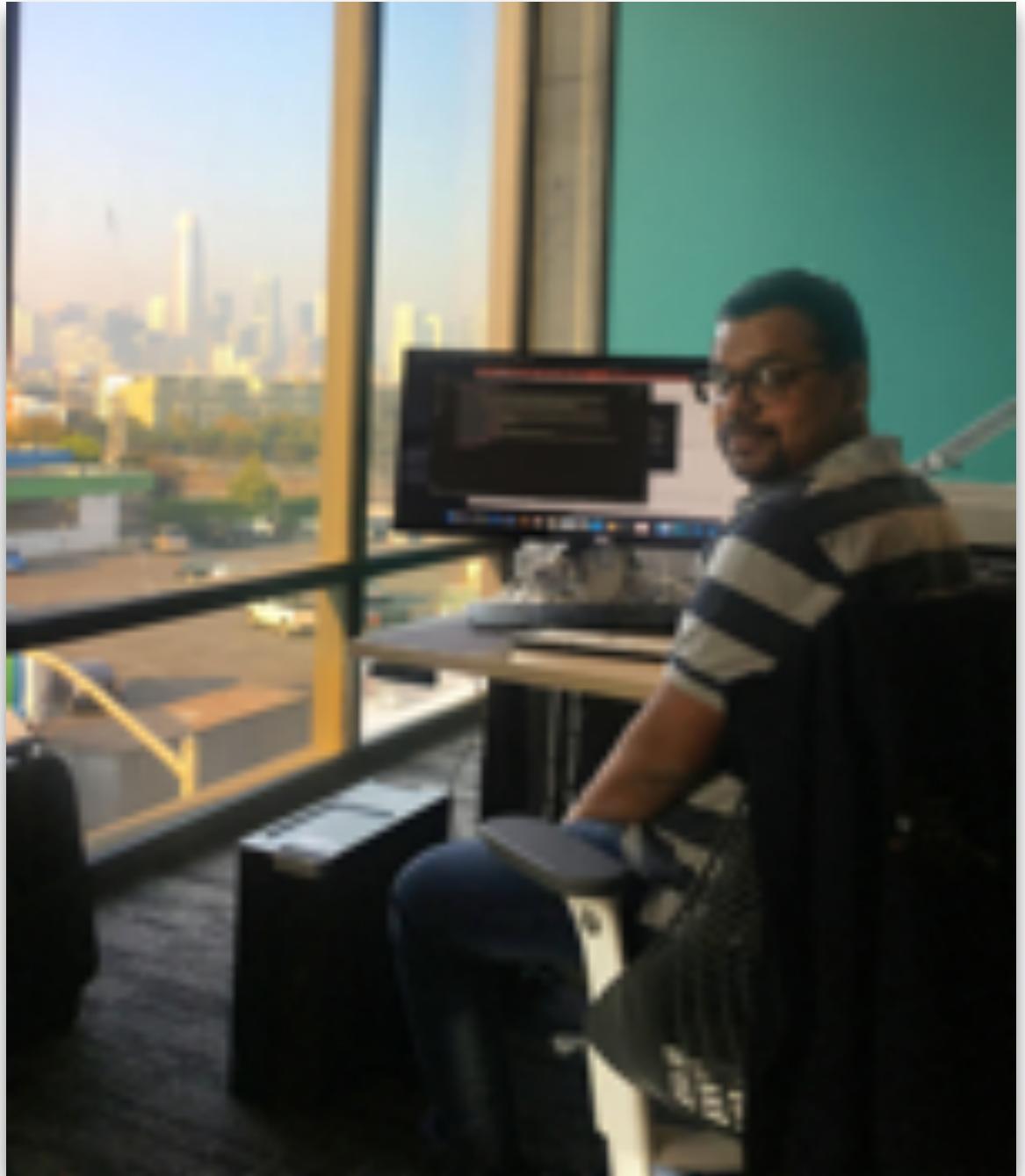


Image by [Alexas_photos](#) on [Pixabay](#)

My experiments with the JAM stack

Me

- Senior Computer Scientist @ Adobe
 - From Photoshop to Adobe Capture.
 - Right here in San Francisco
- Author of VSCode's All Autocomplete
- Author of Hugo In Action
- More at <https://atishay.me>



Sanity of the JAM stack

1. Started with  .
2.  - simple template until defacement.
3.  Blogger™ based low effort solution.
Lacked flexibility.
4.  based static site until it become unbearably slow.
5.  based solution still in action.
6. Writing a book on .



Image by Kidaha on [pixabay](#)



Should you use the JAM stack for eCommerce?

Can JAM stack do it?

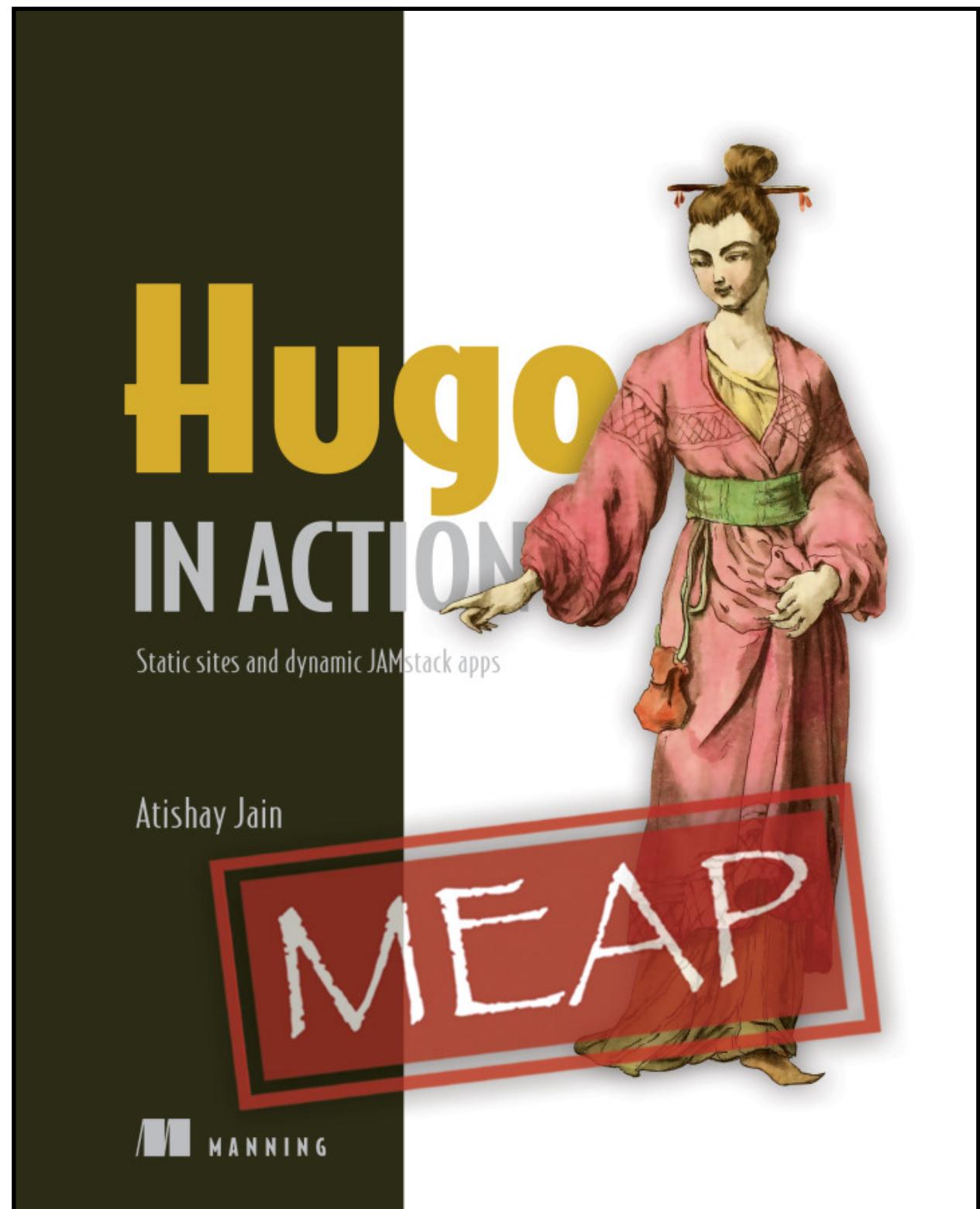


Image by [der_renner](#) on [pixabay](#)

The eCommerce problem

- Many product pages that don't change and have lots of hits.
- Accessibility to bots and search engines.
- APIs for the mobile app.
- Service Oriented Architecture
 - Inventory management system
 - Logistics system.
 - Billing system (third party credit card payment system).



Image by mohamed_hassan on pixabay

How can you use the JAM stack for eCommerce?

The JAM stack solution



JavaScript



APIs



Markup

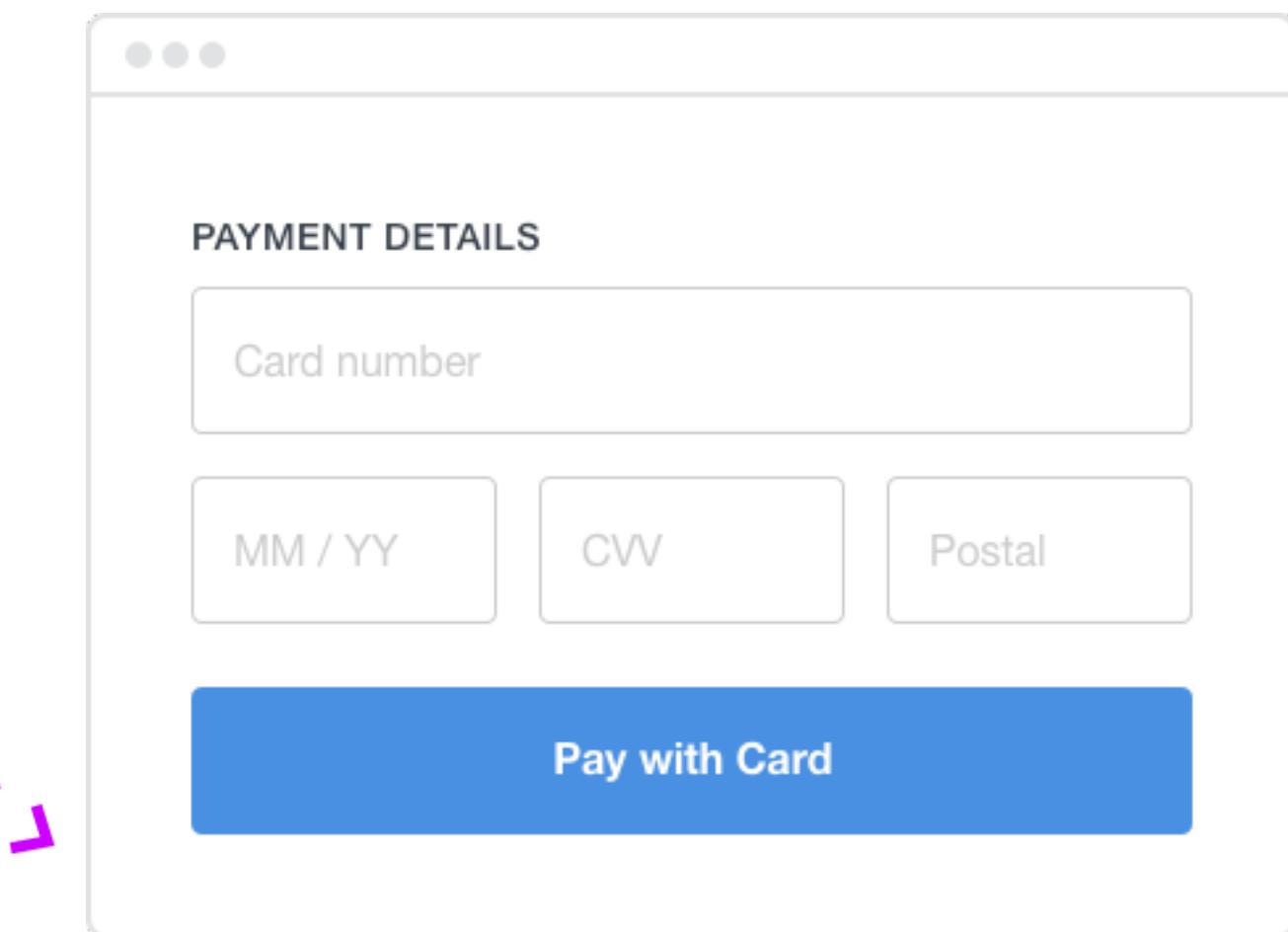
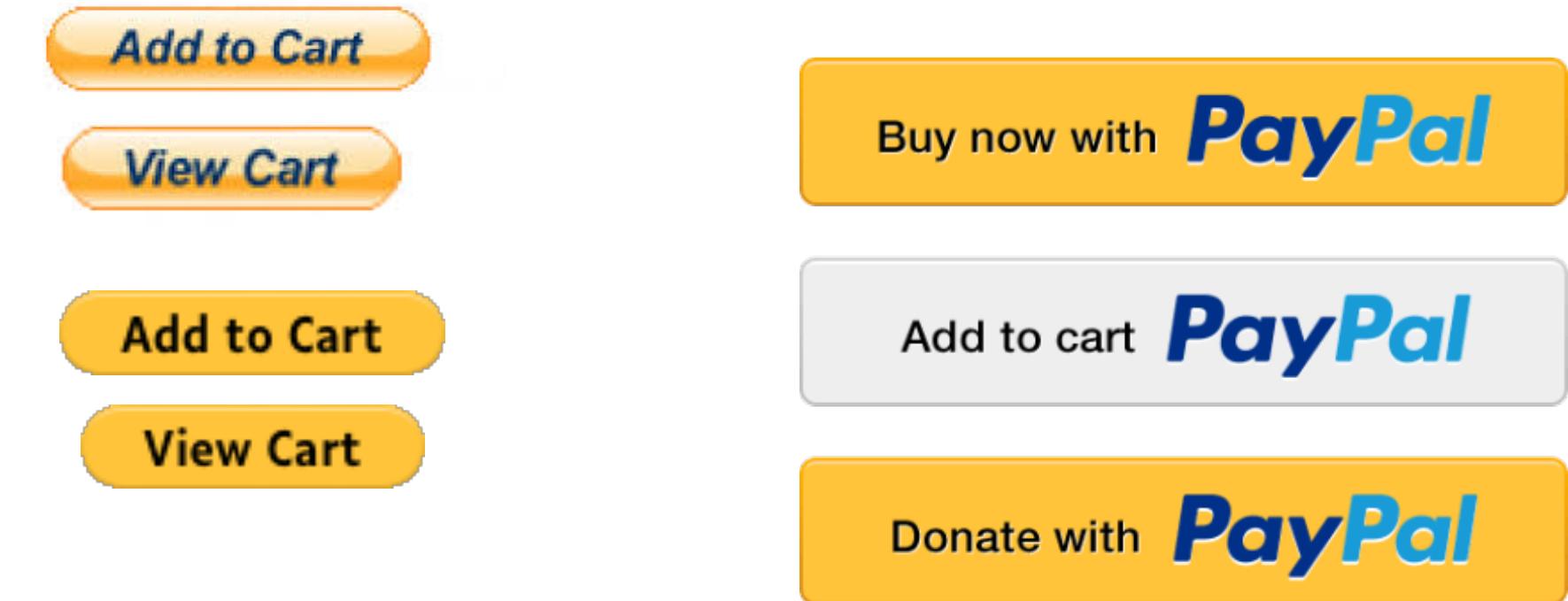
- Communication to the independent systems.
- The shopping cart experience

- The Inventory management system
- The billing system.
- The logistics system.

- The product pages

PayPill with PayPal or Squares with SquareUp

- Third party cart management system.
- You own the product pages, keep them in sync with the payment system.
- The third party system manages billing and inventory.
- No hosted servers, no downtime.



Adding a cloud function for a digital followup

- Use the PayPal Order API or Square's GetPayment API on the server (cloud function) to follow up on a digital request.
- Do it the JAM stack way on a cloud function to not be forced into managing Operations.



Image by noshed on [pixabay](#)

It won't JAM your stack

Conclusion

- Use JAM Stack for ecommerce where
 - Your product selection does not change every minute.
 - Your team has followed the service oriented architecture with a separate inventory and billing system.
 - You prefer to have no DevOps, no on-calls and lots of fun building.
- You can also build a hybrid JAMstack website with the pages like about us or privacy policy built static and the rest can be server based.



Image from [Wikipedia](#)

Thank you

