

Survey results

This survey was constructed in order to investigate how Jamstack is perceived from practitioners in the web development domain focusing on its perceived benefits and limitations, its main application domains, and the supporting tools.

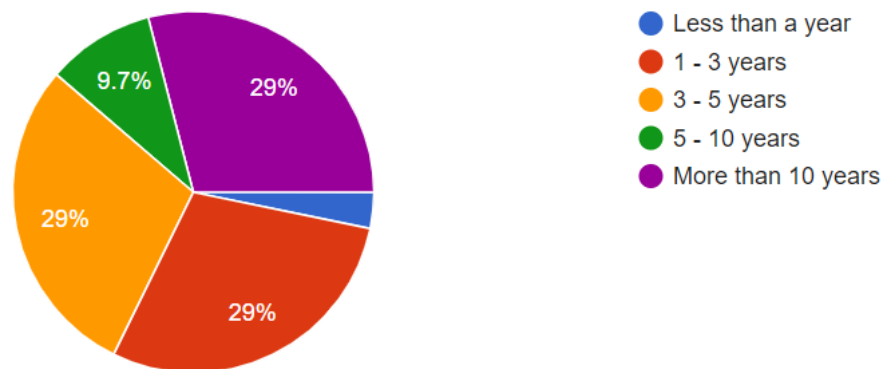
These are the results collected from online survey which involved 31 respondents.

We would like to thank all respondents that took the time to be part of our survey which enabled this research to be possible.

Survey constructed: 17.03.2021.

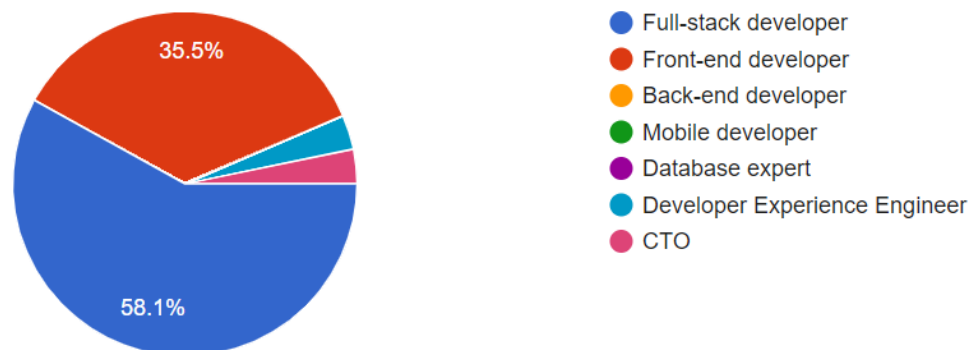
1. How much experience do you have in web development?

31 responses



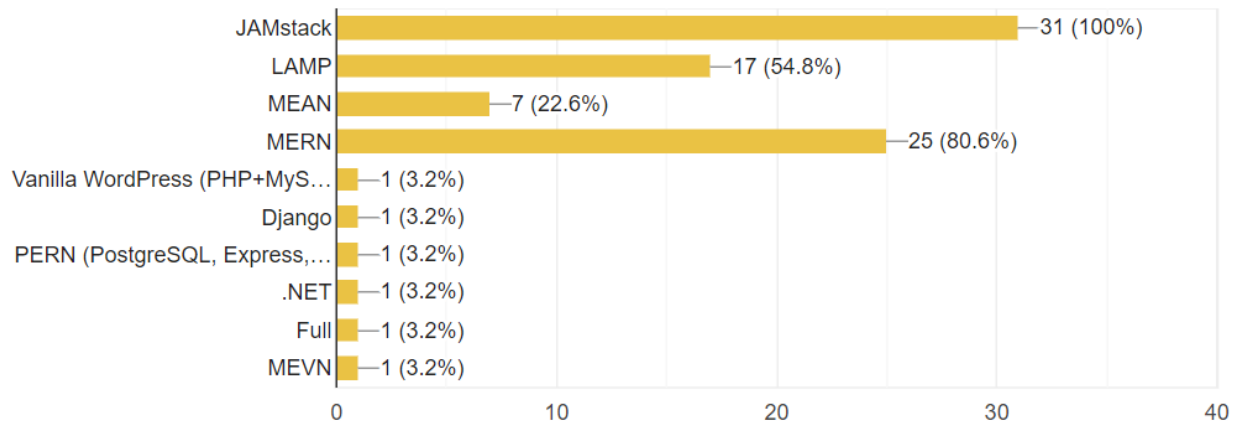
2. Select the role that fits you best?

31 responses



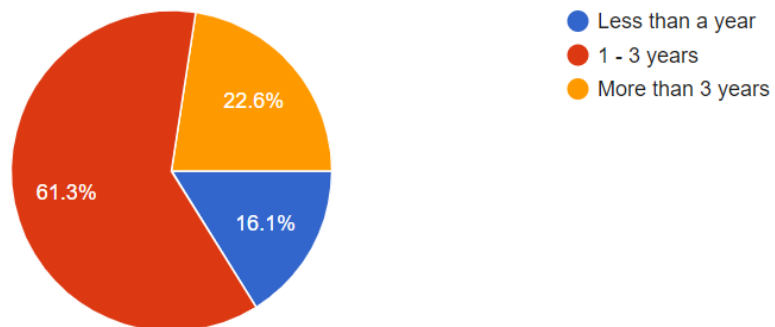
3. Which of the following stacks have you had experience with?

31 responses



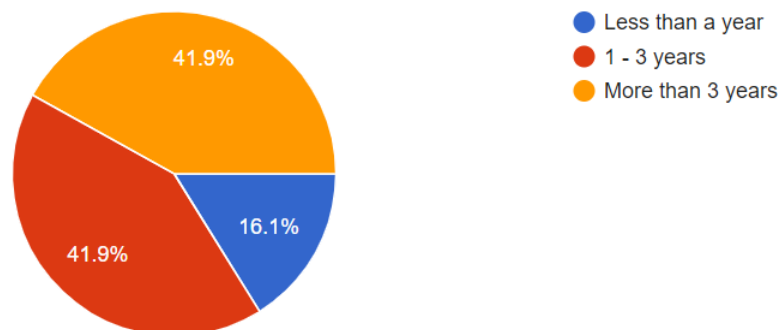
4. How much experience do you have in Jamstack?

31 responses



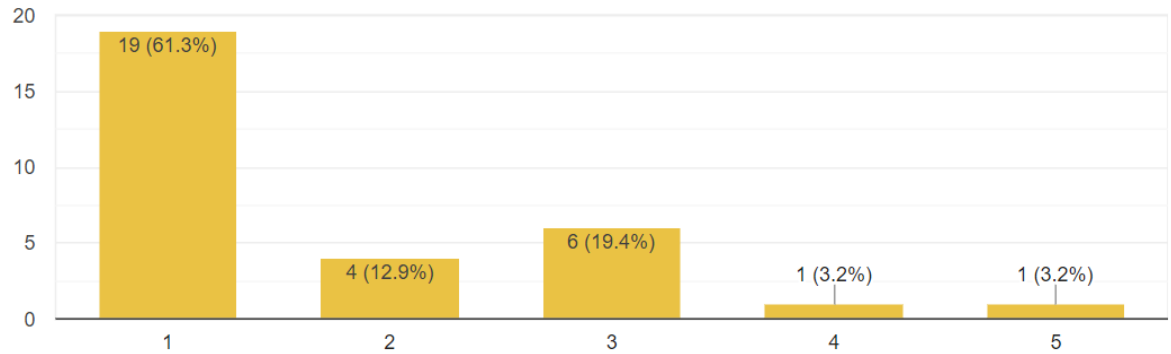
5. How much experience do you have in traditional stacks (LAMP/MEAN/MERN)?

31 responses



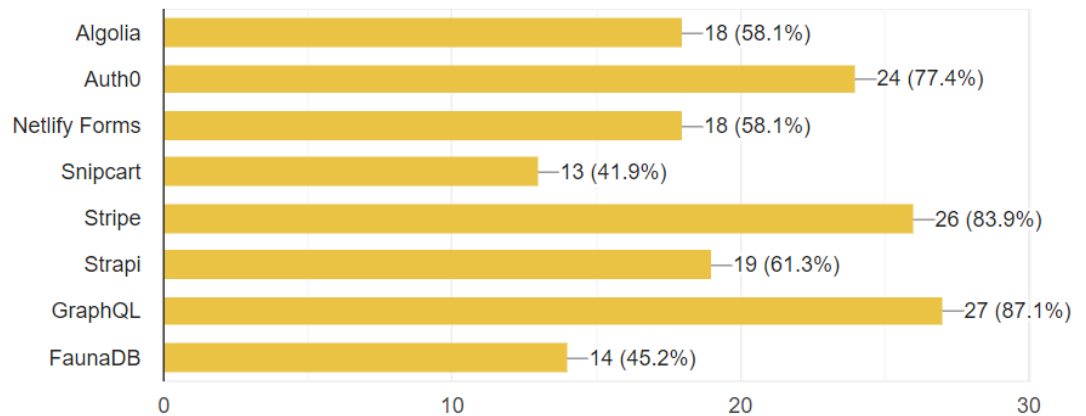
6. To which of the following stacks are you leaning towards? (1 – Jamstack, 3 – neutral, 5 – traditional stacks)

31 responses



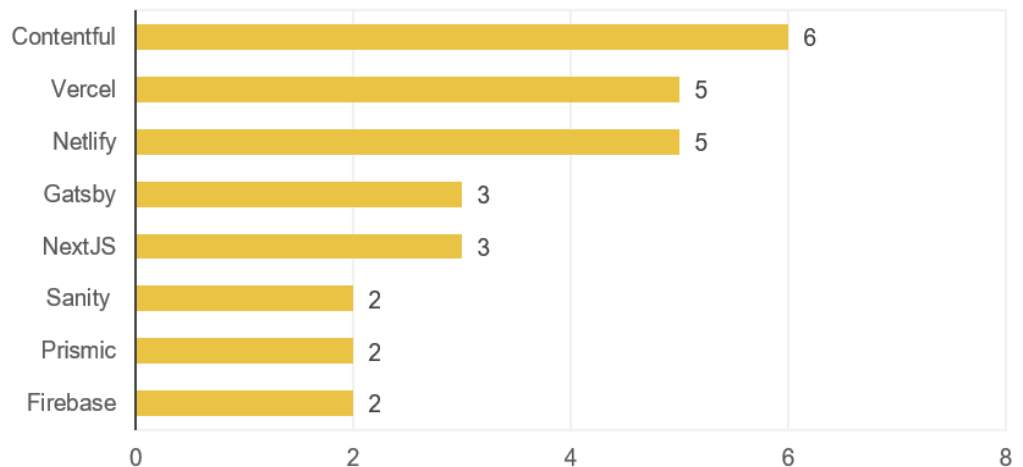
7. Which of the following APIs and tools have you heard of and find useful for Jamstack?

31 responses



8. Are there any APIs or tools for Jamstack that you would particularly point out?

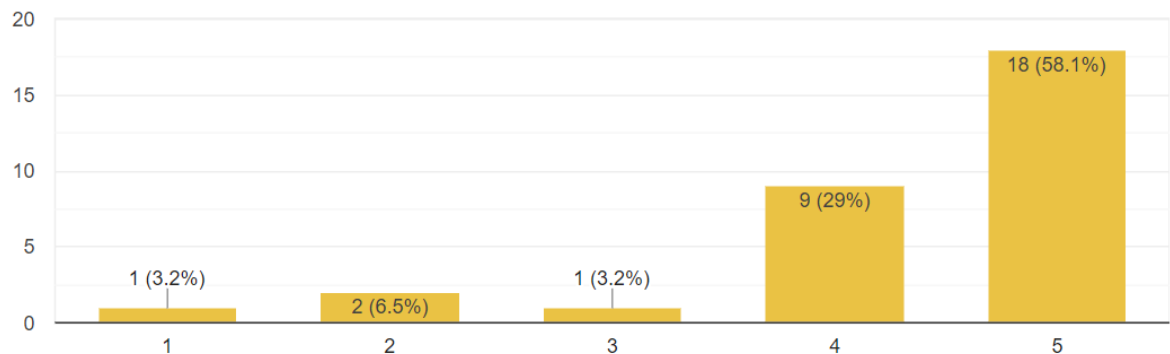
26 responses



9. Jamstack benefits (1 - strongly disagree, 3 – neutral, 5 – strongly agree)

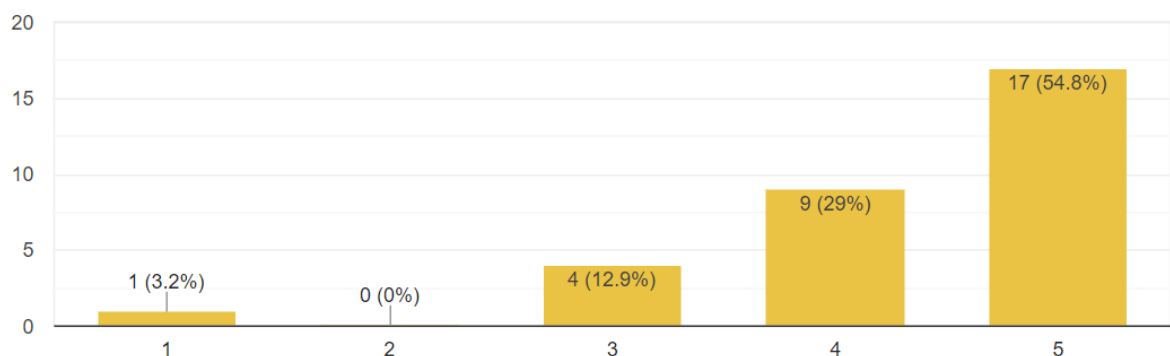
- a) Performance - With static pages delivered through CDN the loading speed of the JAMstack website usually achieves the highest possible score on speed tests. There are three aspects that are usually used as metrics of website speed and they are: LCP (largest contentful paint), FID (first input delay), CLS (cumulative layout shift).

31 responses



- b) Security – Thanks to the static storage or CDN that serves your application, APIs that are read-only and accessible exclusively for building environment, attacks can be reduced to minimum if not even removed completely. Since the layer has no database, sites are not vulnerable to server-side include injections. Besides, the use of external platforms such as Netlify can provide protection against DDOS and DNS attacks.

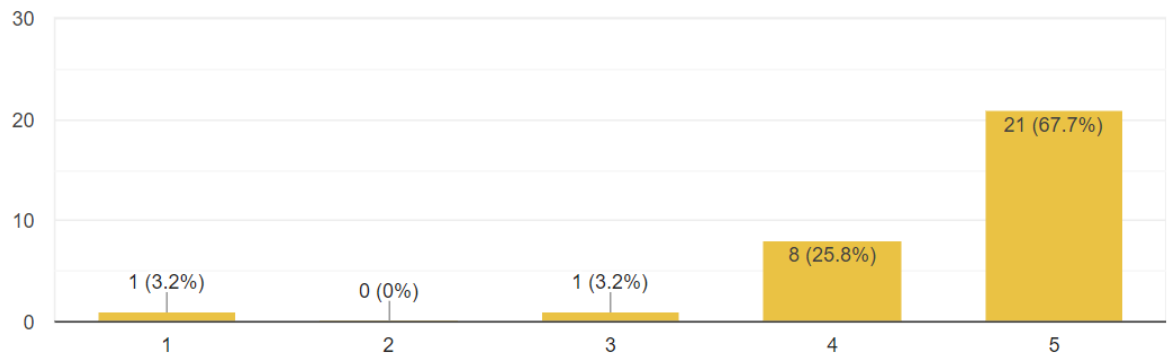
31 responses



- c) Flexibility - Jamstack uses third-party services more known as headless CMS for managing content on the website. Since headless CMS are separated from the front-

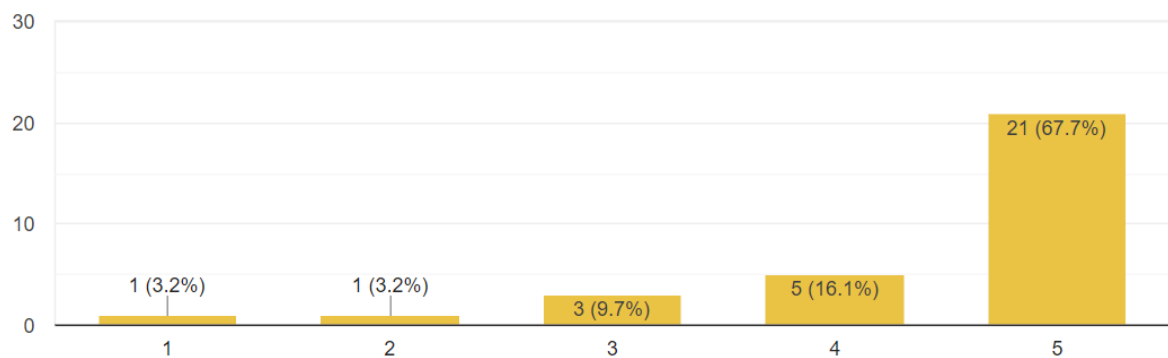
end of the website it means that the API that is created with headless CMS can be used and easily displayed across multiple platforms.

31 responses



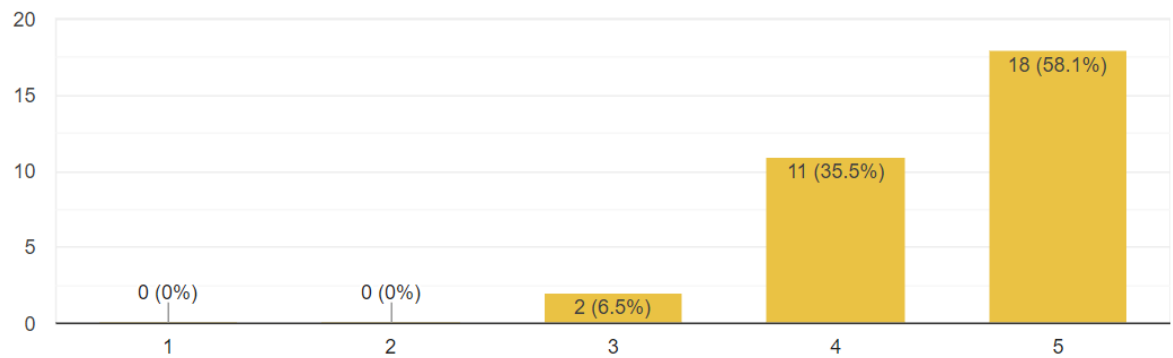
- d) Pricing - The whole Jamstack website requires a small amount of resources and can be hosted on CDN using some free static hosting like Netlify or GitHub Pages which will reduce the cost.

31 responses



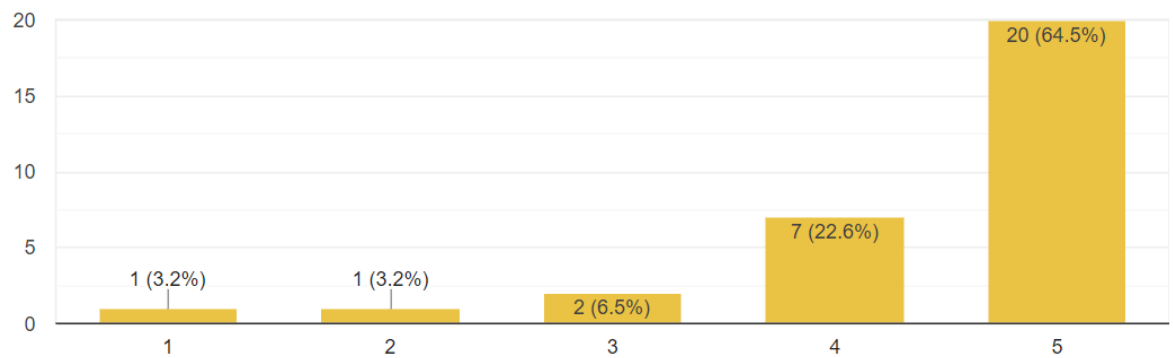
- e) SEO - Jamstack page content is static which means that the speed of the website will be fast and indexing the pages of the website will be a lot easier in comparison with other web stacks which are the two most important factors for a good ranking on the internet.

31 responses



- f) Developer experience - Jamstack developers have more freedom in choosing the technologies they will use in accordance with their knowledge, wishes and needs unlike other web stacks. The fact that front-end is separated from the backend makes developer experience far more enjoyable and easier because they can focus on one thing. Since JAMstack is static, all files can be stored on GitHub, Netlify or Vercel which makes the deployment easy and what matters to many, all for free.

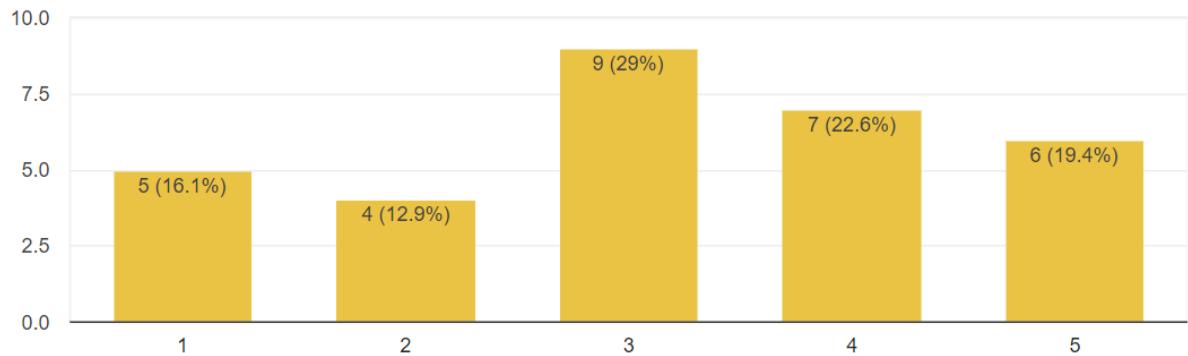
31 responses



10. Jamstack limitations (1 – strongly disagree, 3 – neutral, 5 – strongly agree)

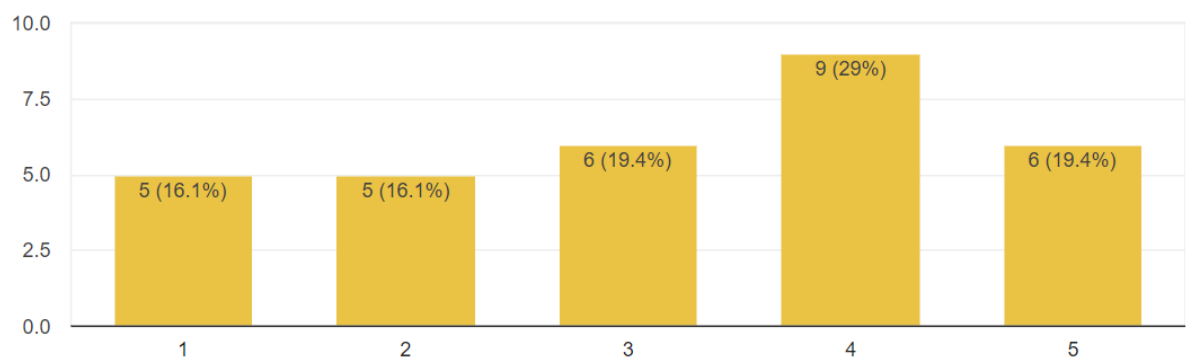
- a) Dynamic functionalities - While Jamstack can be dynamic it is usually harder to implement some dynamic functionality and it mostly requires relying on third-party services for adding more complex features (For example if you want to have search on your website you can't do it on your own. You will need to use Algolia or other third-party service).

31 responses



- b) Delay in live preview - Before seeing the actual changes on the website, everything needs to be pushed on Github. This can cause frequent delays especially when it comes to very dynamic sites that need constant changing of the content.

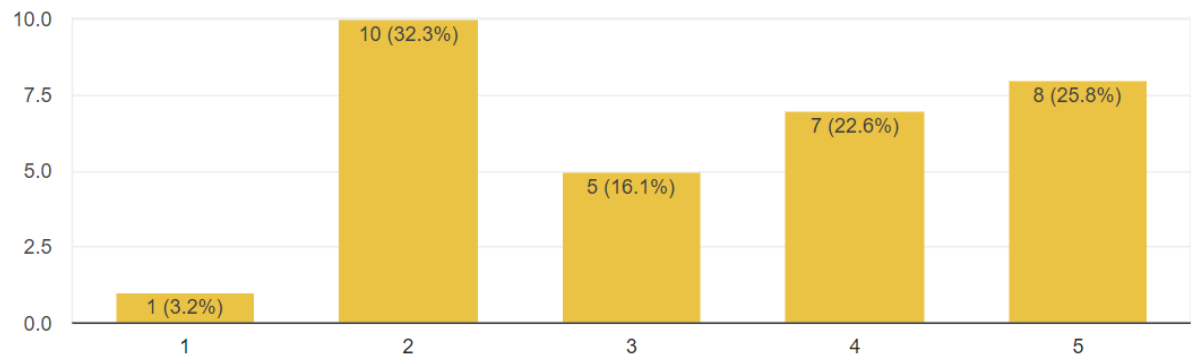
31 responses



- c) Reliance on third-party systems

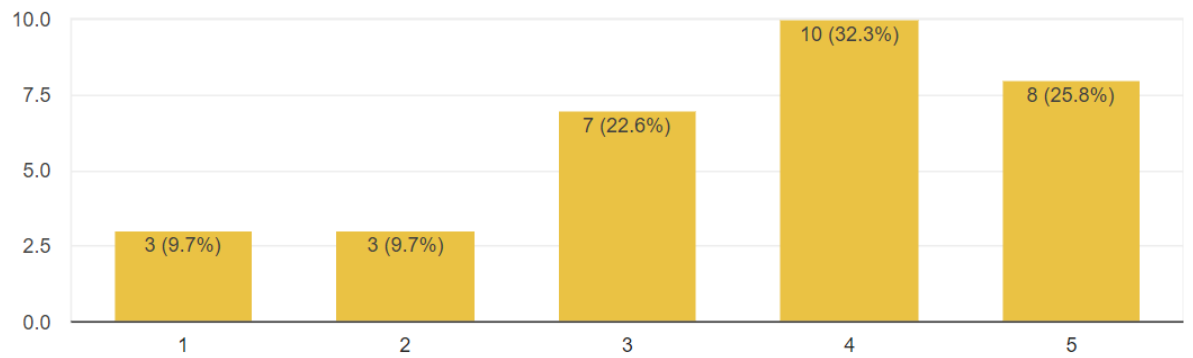
Availability - In case the system goes down, your website will feel the consequences because there is no communication with the API. Whether a solution will be reached and if the system will function again, is out of your hands and depends entirely on the provider.

31 responses



Confidentiality - Entrusting all content to a third-party is the risk that confidential content could be disclosed.

31 responses



11. Did you encounter any other limitations or benefits that are not already listed while using JAMstack? Please describe them.

29 responses

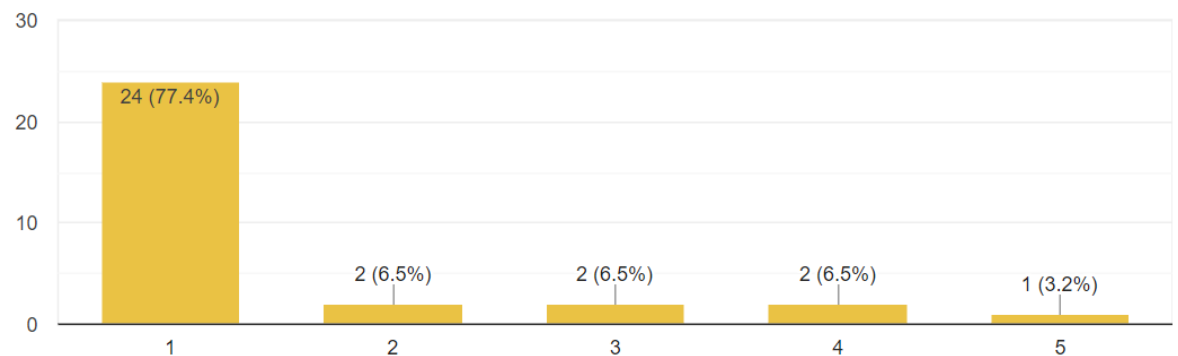
Respondent	Answer
1.	Headless CMS still lags behind traditional Wordpress etc in usability for non-tech users
2.	n/a
3.	It was (up until very recently - last 60 days) that if you needed to make changes, modify or add additional things that the build time in Gatsby was a bit long. It's been fixed and that you must use a third-party tool to properly compress, lazy load and render (like a Cloudflare). There is no one perfect system or tool (they all have Pro's and Cons) so it's about choosing the right tool for the job.
4.	JAMstack falls short when encountering any sort of role-based authentication.
5.	Content is reusable and can be saved as flat files in useful formats like markdown
6.	Costs - Usually we have to add multiple third party service, It's not just one thing and that may cost more even if the cost of hosting itself is low.
7.	Performance and security are the top 2, definitely. I think there should be more questions about accessibility here; as that's part of performance.
8.	Sometimes it's slightly tough to explain to clients why their site is not in WordPress.
9.	I did not
10.	It can be easy to get started but there is a pretty significant learning curve.
11.	I've found that platforms like WordPress are easier for small clients and business to use, understand, and build with at least at this point. Connecting all the services together (and maintaining the connections) in JamStack can sometimes be a hassle and can cost more as well.
12.	You can make hybrid apps for adding features that are not available in jamstack
13.	Headless CMS allows employees with lesser to no knowledge in softwares to put their input for a live website. Example: A retail salesgirl can put an urgent discount using the headless CMS inorder to cope up with the market changes whereas, companies would spend a lot of time and effort to hire a developer to do that for them.
14.	Historically you didn't always see the latest content, new improvements from Nextjs/Gatsby have gone a long way to help rectify this.
15.	Initial interaction with the page can bit slow if you have CSS, image heavy page.
16.	I think you are confused on the idea that APIs *have to be* third party services. The APIs on which a JAMstack site or application is built on can be built in-house and

	hosted on-premise. So these limitations are native to third-party cloud services, not the JAMstack.
17.	Real time features, web-sockets and lack of database connection pooling technologies
18.	not everything can be pre rendered and smallest change requires an entire rebuild.
19.	Because the cms are not self hosted(except for the top tier ones), statically generating new pages may prove a lot harder than it should be and require more technique. It's easier serve content towards pre-existing static pages rather than create those pages with the cms like word-press.

12. In your experience with JAMstack and traditional stacks, for which kind of site one would be a more suitable option than the other? (1 - Jamstack, 3 – neutral, 5 – traditional stacks)

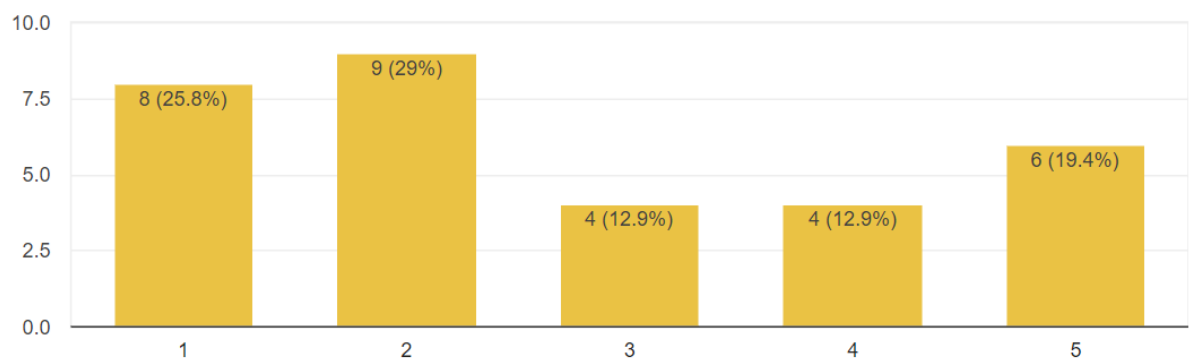
a) Blog and Portfolio

31 responses



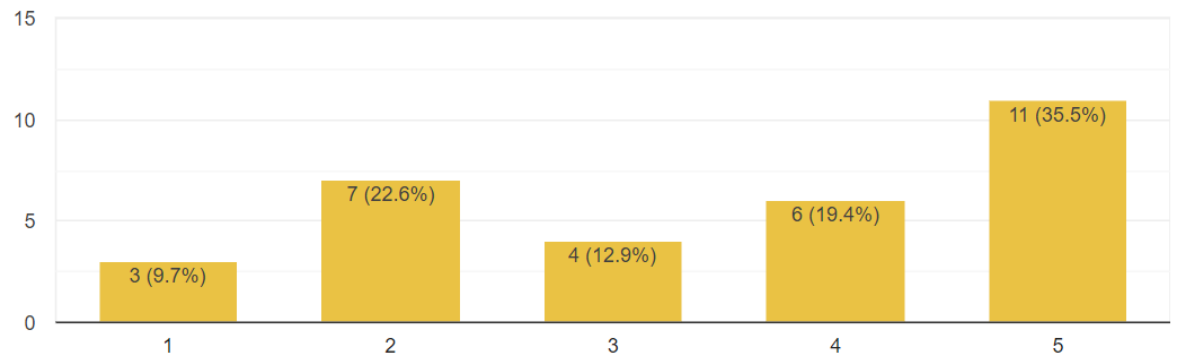
b) E-commerce

31 responses



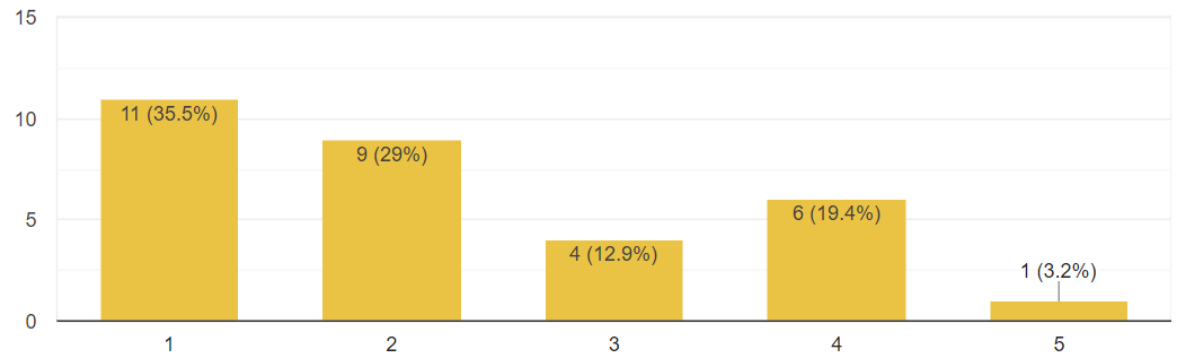
c) Social media

31 responses



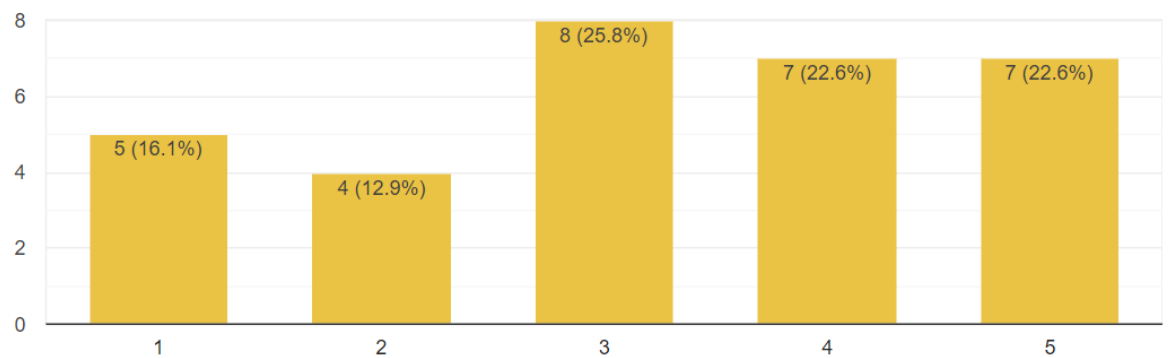
d) News and Magazine

31 responses



e) Media streaming

31 responses



13. What would you list as the advantage of traditional stacks (LAMP, MEAN, MERN) over JAMstack? Please explain.

23 responses

<i>Easier learning curve</i>	<i>More capabilities</i>	<i>No reliance on third party services</i>	<i>More established</i>	<i>More control</i>
<i>Jamstack is paradigm that can take a while to get beyond basics</i>	<i>Lamp, Mean, Mern have more capabilities.</i>	<i>No third party providers</i>	<i>LAMP most established and more widely used</i>	<i>Lamp, Mean, Mern allows you to host everything and build infrastructure</i>
<i>MERN have easier learning curve</i>	<i>Dynamism</i>	<i>Challenge of reliability is huge to discard as we are entrusting our data to third party services</i>	<i>High support, mature and scalable.</i>	<i>Control of the process from development to production</i>
<i>Easier to get started</i>	<i>Faster database queries</i>	<i>No 3rd party providers eyes on your data</i>	<i>Better for clients, biggest ecosystem</i>	<i>Traditional stacks gives more control</i>
	<i>Using traditional stack will give you better flexibility (functionality)</i>	<i>Can be cheaper than paying for multiple services</i>		<i>Total control over your code.</i>
	<i>Functionality tailored for user sessions</i>			

14. In your experience so far, do you think JAMstack will be able to overcome other web stacks in the future and become de facto standard? Please elaborate your answer.
27 responses

Respondent	Answers
1.	JAMstack is a paradigm that can take a while to get your head round beyond the basics.
2.	LAMP, MEAN, MERN are more flexible, with more capabilities. Many functionalities that are data-heavy are easier to implement in these traditional stacks.
3.	A LAMP/MEAN/MERN stack would allow you to host everything and build the infrastructure, which may be what you need for a bespoke solution.
4.	LAMP = Very battle tested, most established and most widely used by development community. MEAN = Very Good (up to a certain point for certain scenarios) in enterprise level, Object-Oriented app way. MERN = The fastest growing stack in adoption that will or has already surpassed LAMP; easier learning curve.
5.	It can be hard to prove compliance with PCI and HIPAA.
6.	Dynamism
7.	Content updates instantly
8.	control of the process from development to integration and automation to production
9.	Controlling the back-end and tightly coupling it with the front-end means exactly what you'd imagine; faster db queries and no 3rd party providers' eyes on your data.
10.	If you're building a web application with significant business logic, using a traditional stack will give you better flexibility and control (e.g. Using Django/RoR).
11.	A lot quicker!
12.	If your application requires integration to legacy back-end systems like accounting, WMS, etc... the older stacks probably have more 3rd party plugins or integrations.
13.	Easier to get started with, better for clients, bigger ecosystems (so far), can be cheaper than paying for multiple services
14.	Performance score, Security, Reusability, etc..
15.	The challenge of reliability is pretty huge to discard as we are entrusting our entire business to third party services. Therefore it is of utmost importance that a backup of the data should be stored in your system regularly.
16.	Dynamic content and functionality tailored for the user session.
17.	When access to dynamic data is needed to build content.

18.	Traditional stack has advantage when you're building a pure client based application.
19.	Developer experience, developer experience, developer experience and finally... developer experience!! The fact that a developer can build a whole system with much effort without having to rely on infrastructure (or devops processes depending the services you use) makes the whole experience really enjoyable.
20.	Single source of truth for code and content
21.	Developer Experience
22.	high support, mature, and scalable.
23.	Total control over your code to bootstrap more novel functionalities for demanding sites, Minimal dependence on third party offerings hence reducing downtime chances.