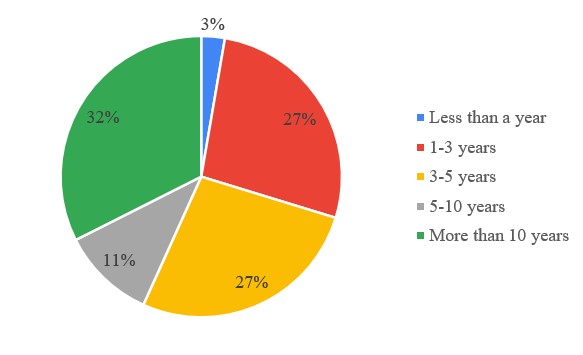
**Survey results**

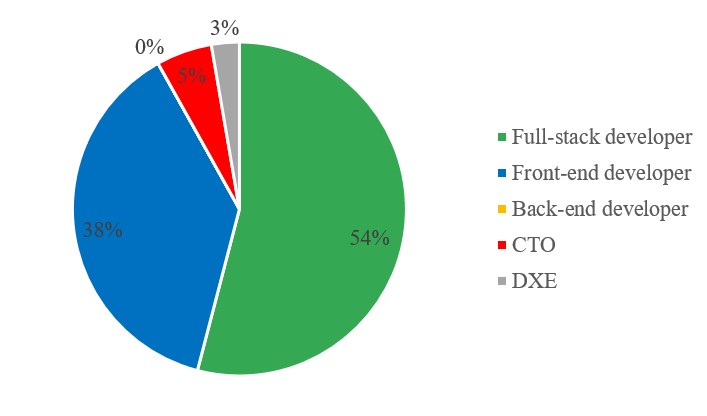
1. How much experience do you have in web development?

*37 responses*



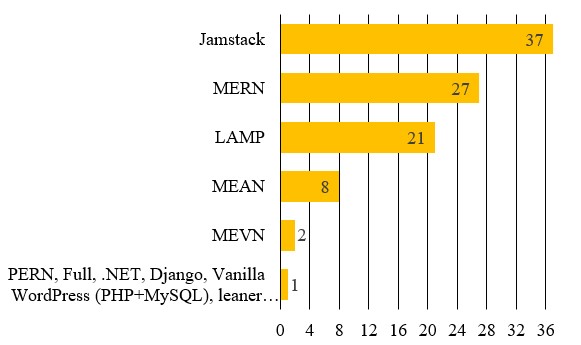
1. Select the role that fits you best:

*37 responses*



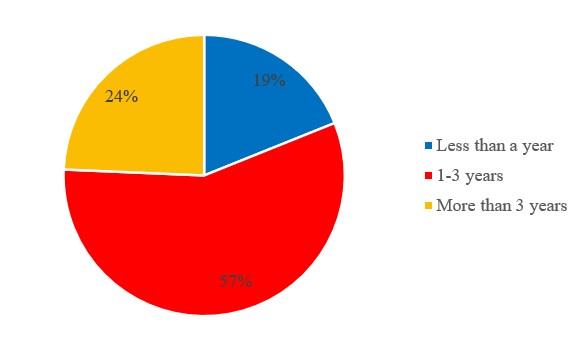
1. Which of the following stacks have you had experience with?

*37 responses*



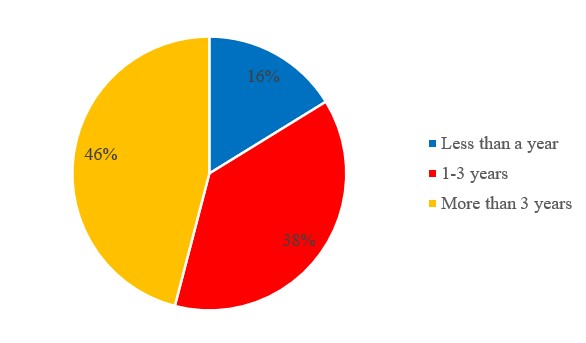
1. How much experience do you have in JAMstack?

*37 responses*



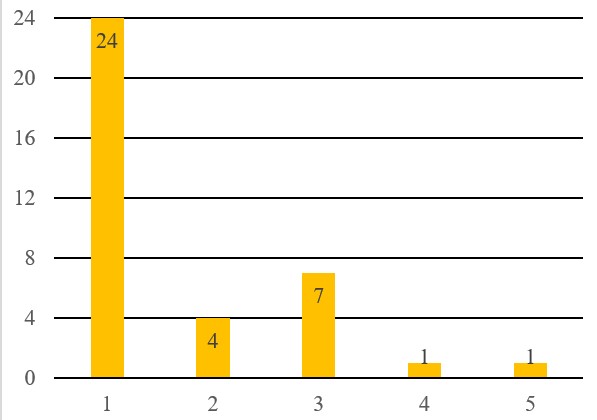
1. How much experience do you have in traditional stacks (LAMP/MEAN/MERN)?

37 responses



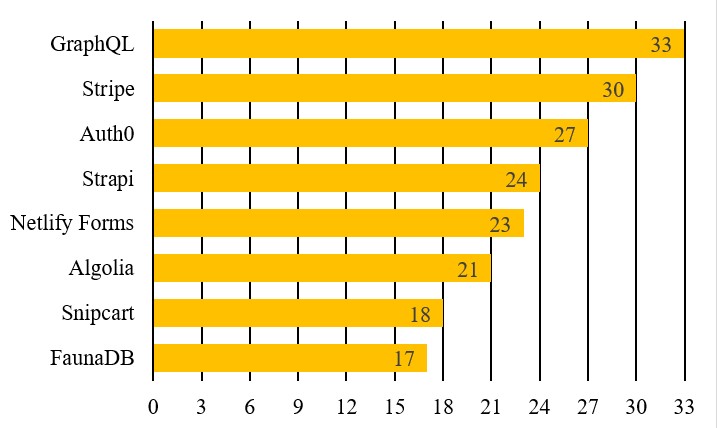
1. To which of the following stacks are you leaning towards? (1 – Jamstack, 3 – neutral, 5 – traditional stacks)

*37 responses*



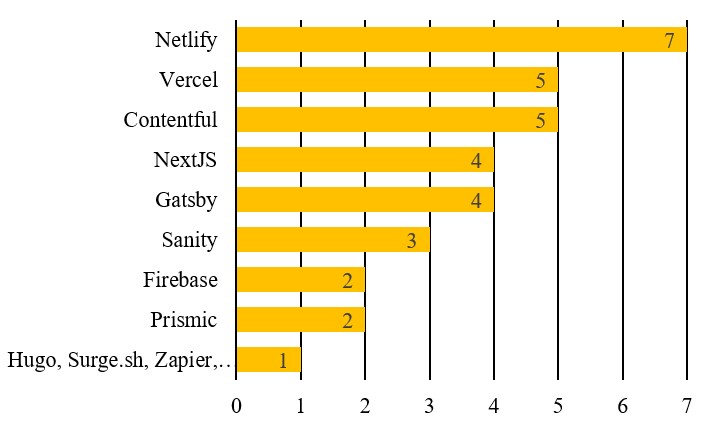
1. Which of the following APIs and tools have you heard of and find useful for Jamstack?

*37 responses*



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

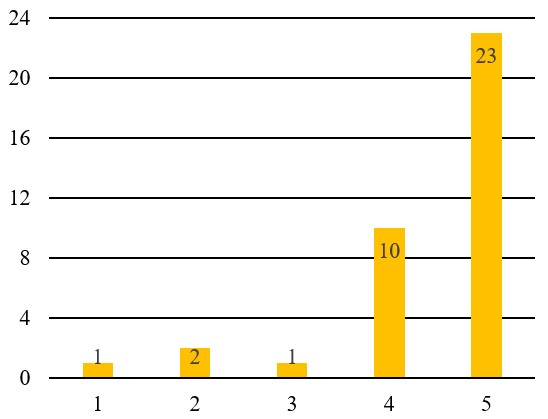
*32 responses*



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

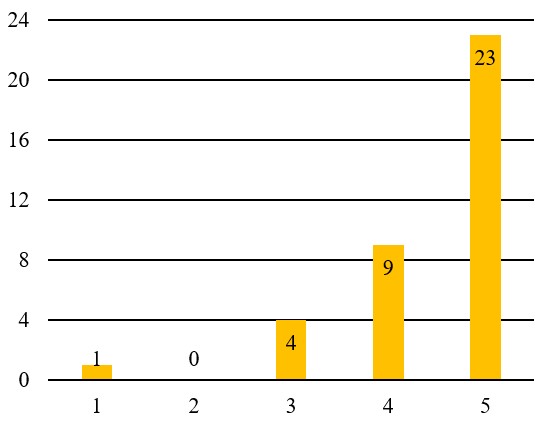
Performance - With static pages delivered through CDN the loading speed of the JAMstack website usually achieves the highest possible score on speed tests. There are three aspects that are usually used as metrics of website speed and they are: LCP (largest contentful paint), FID (first input delay),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

*37 responses*



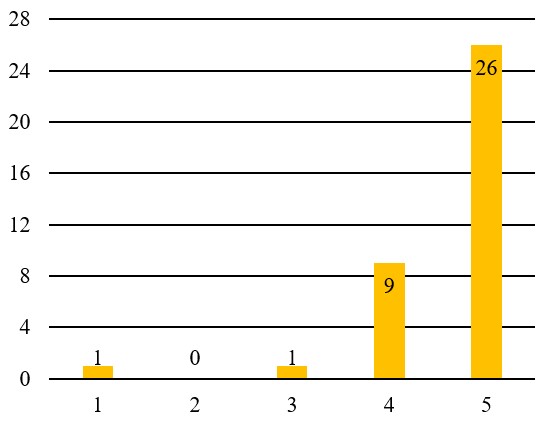
1. Security – Thanks to the static storage or CDN that serves your application, APIs that are read-only and accessible exclusively for building environment, attacks can be reduced to minimum if not even removed completely. Since the layer has no database, sites are not vulnerable to server-side include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

*37 responses*



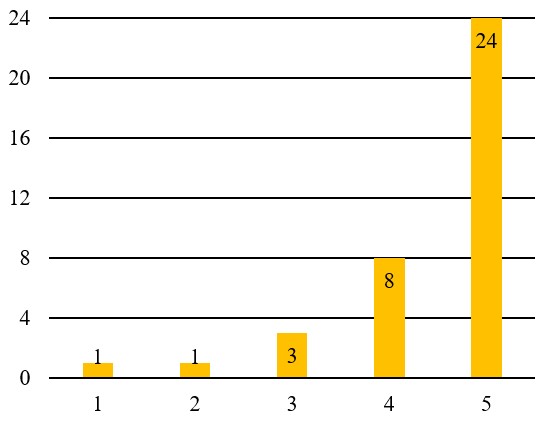
1. Flexibility - Jamstack uses third-party services more known as headless CMS for managing content on the website. Since headless CMS are separated from the front-end of the website it means that the API that is created with headless CMS can be used and easily displayed across multiple platforms.

*37 responses*



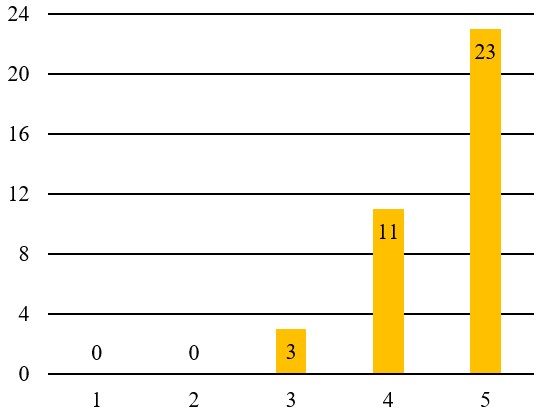
1. Pricing - The whole Jamstack website requires a small amount of resources and can be hosted on CDN using some free static hosting like Netlify or GitHub Pages which will reduce the cost.

*37 responses*



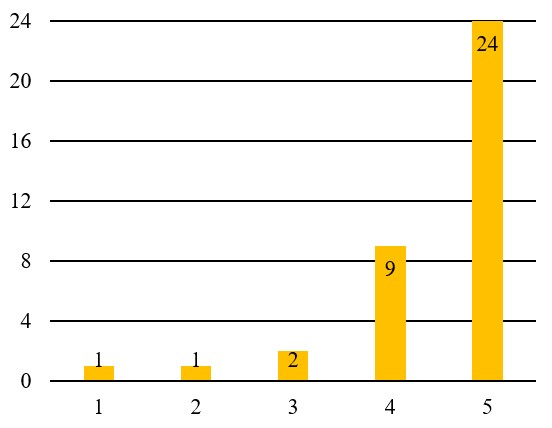
1. SEO - Jamstack page content is static which means that the speed of the website will be fast and indexing the pages of the website will be a lot easier in comparison with other web stacks which are the two most important factors for a good ranking on the internet.

*37 responses*



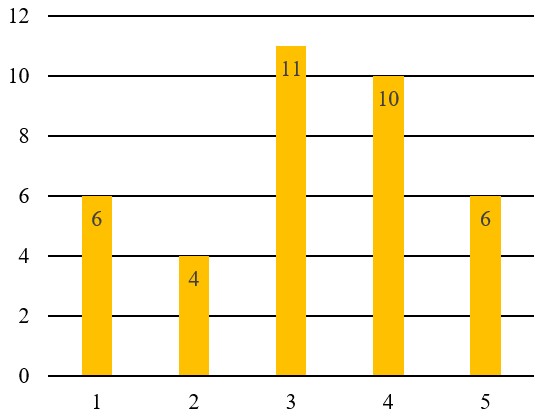
1. Developer experience - Jamstack developers have more freedom in choosing the technologies they will use in accordance with their knowledge, wishes and needs unlike other web stacks. The fact that front-end is separated from the backend makes developer experience far more enjoyable and easier because 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.

*37 responses*



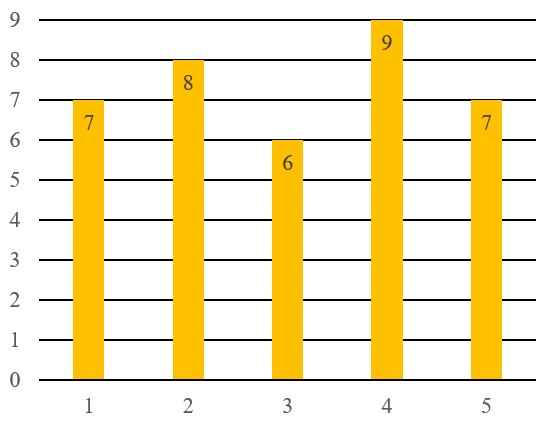
1. Jamstack limitations (1 – strongly disagree, 3 – neutral, 5 – strongly agree)
2. Dynamic functionalities - While Jamstack can be dynamic it is usually harder to implement some dynamic functionality and it mostly requires relying on third-party services for adding more complex features (For example 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).

*37 responses*



1. 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.

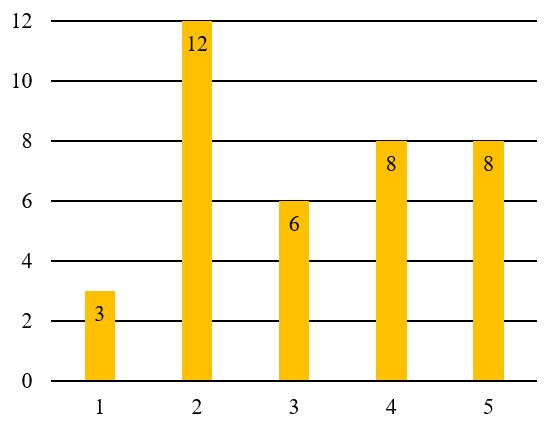
*37 responses*



1. Reliance on third-party systems

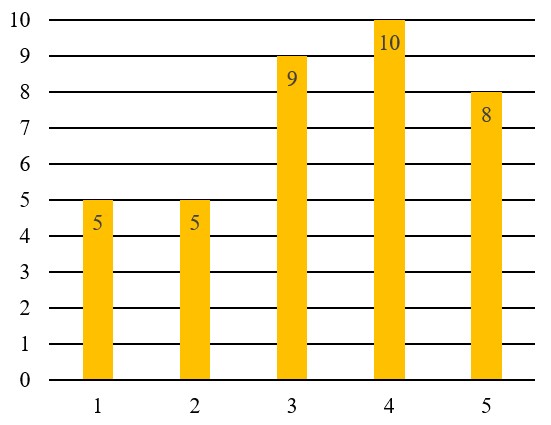
Availability - In case the system goes down, your website will feel the consequences because there is no communication with the API. Whether a solution will be reached and if the system will function again, is out of your hands and depends entirely on the provider.

*37 responses*



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

*37 responses*



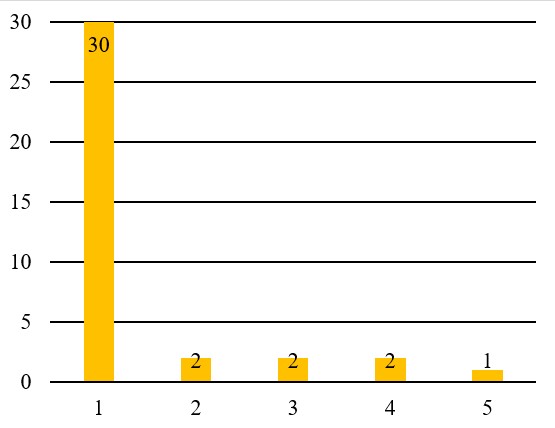
1. Did you encounter any other limitations or benefits that are not already listed while using JAMstack? Please describe them.

*24 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 compresss, lasy load and render (like a Cloudfare). 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. |
| 20. | Full control of data. Ability to move to different storage requirement (eg EC2) |
| 21. | no there are no limits |
| 22. | n/a |
| 23. | Interchangeability between backend services |
| 24. | n/a |

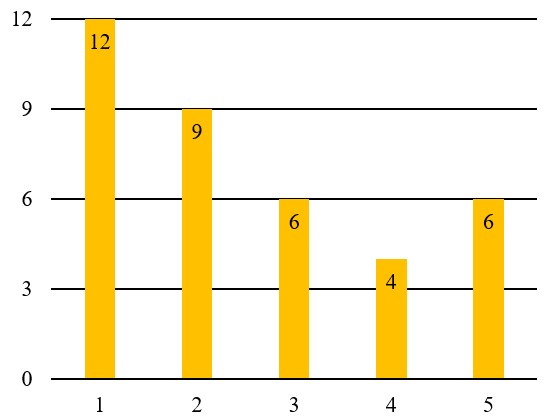
1. In your experience with JAMstack and traditional stacks, for which kind of site one would be a more suitable option than the other? (1 - Jamstack, 3 – neutral, 5 – traditional stacks)
2. Blog and Portfolio

*37 responses*



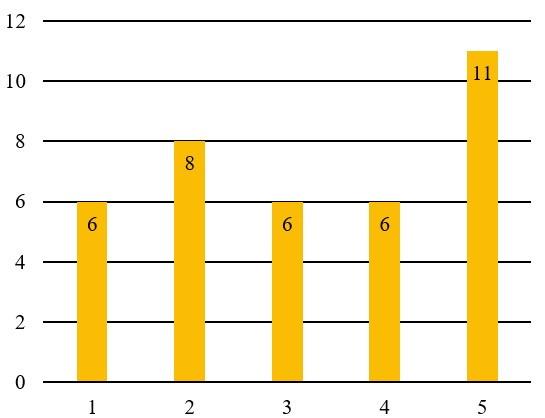
1. E-commerce

*37 responses*



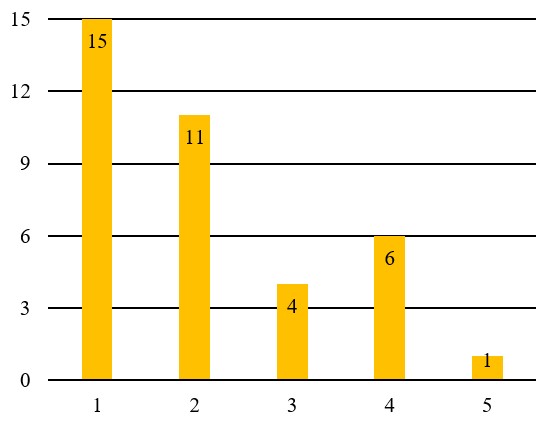
1. Social media

*37 responses*



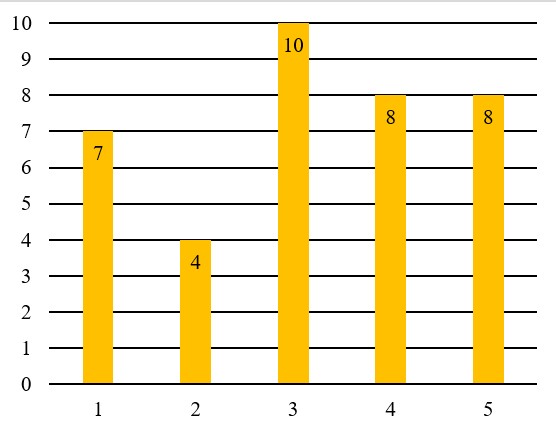
1. News and Magazine

*37 responses*



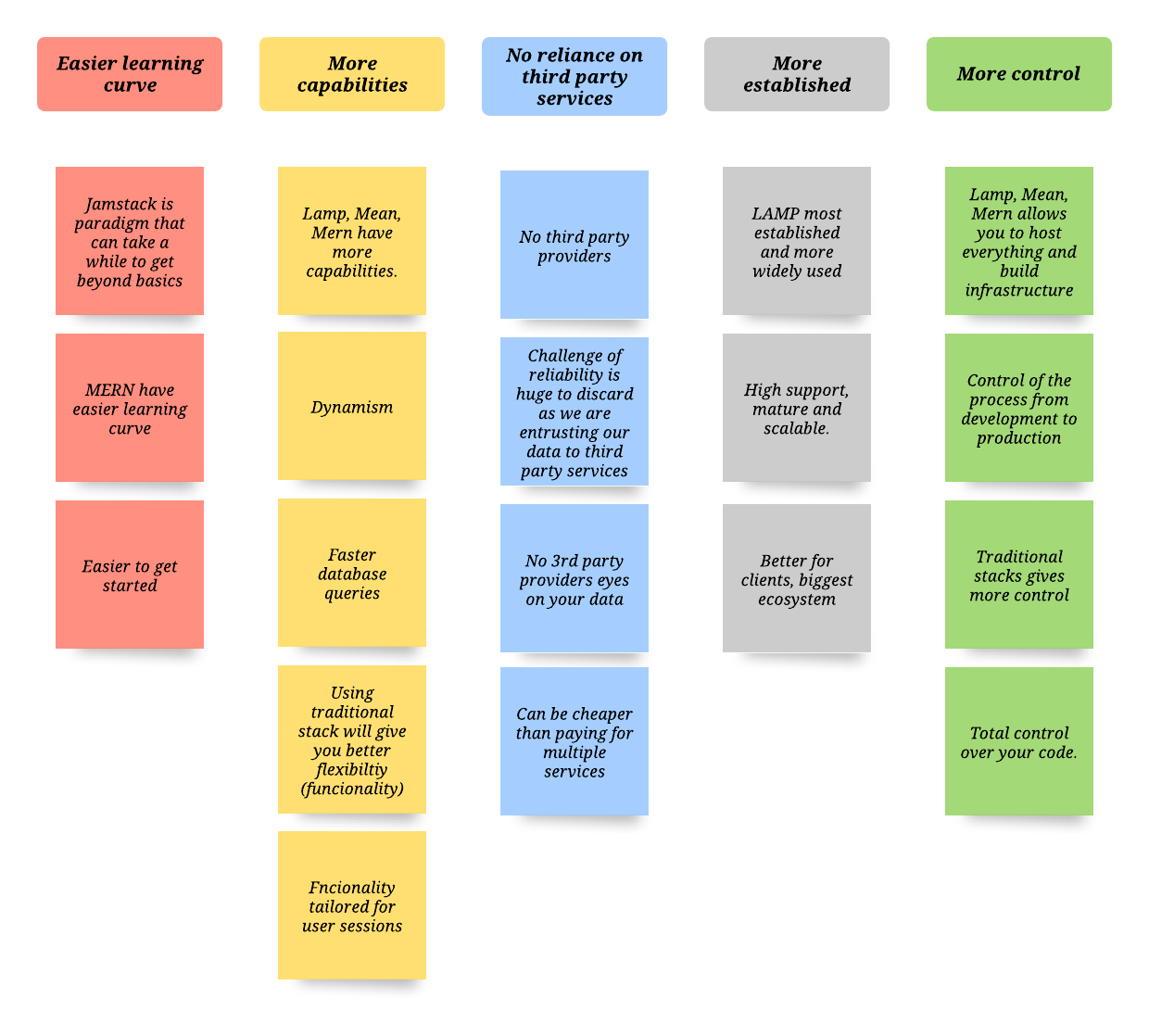
1. Media streaming

*37 responses*



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

*27 responses*



1. 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. (1 – Not at all, 3 – Neutral, 5 - Definately)

*28 responses*

|  |  |
| --- | --- |
| **Respondent** | **Answers** |
| 1. | I do think JAMstack will grow exponentially over the next few years for all the reasons we have already discussed - cost, speed of rendering, flexibility, CDN delivery, and also it feels like a return to classic web development (i.e., writing clean HTML, CSS, and JavaScript). |
| 2. | I think in the long run JAMstack with the help of technologies like serverless would not only overcome other traditional stacks but also become the standard for web development. Using serverless will help fill in spaces where having a traditional architecture seem to have an edge over JAMstack. JAMstack developers will become the new fullstack developers. Areas like performance are starting become larger areas of focus for the web development field |
| 3. | Yes. |
| 4. | I think not, because the learning curve is high and it lacks traditional methods for realtime problems. |
| 5. | It's growing, and it will keep growing. Probably will become the most used stack, yeah. Because it's simple, fast, and you can build almost anything with it. And it has a great community, lots of documentation, tutorials and it keeps evolving, it's not going to be deprecated anytime soon. |
| 6. | It depends on how you define JAMstack |
| 7. | I think Jamstack will be here to stay, but there is probably a place for both types. I hope to see Jamstack evolve so that platforms get better, ecosystems get bigger, integrations more tight, deploy times faster, etc. I think it will keep getting bigger over time. |
| 8. | I think it all depends on the scope of the site / app. Some scenarios I can think of where JAMstack will become de facto standard are marketing pages and e-commerce sites, blogs and information sites. |
| 9. | No. It will be the preferred solution for marketing websites, but it is not one size fits all. Every stack has its right purposes. |
| 10. | Yes, this is because of the simplicity and bar-of-entry into the JAMStack ecosystem. Comparing the simplicity, security, reliability, cost, and performance gains to traditional stacks makes this a worthwhile investment where the problem fits a JAMStack solution. |
| 11. | Nope, because it's not the best solution for everything, we still need other web stacks. |
| 12. | Definitely |
| 13. | Yes |
| 14. | Not in SSR (Server Side Rendering) heavy needed applications but yes in client side, SPA's, CMS's, and with the trend of decoupling and containerization progression + in combination toolings such as microservices, Progresive Web Apps, Decentralized Apps (D'Apps), Serverless fuctions, etc. It will continue to rise exponentially and it's place for a great deal of scenarios and situatiions. |
| 15. | Not sure it'll ever become a de facto standard, but it'll certainly not fade away. |
| 16. | I think it's currently gaining a lot of ground on today's market already! |
| 17. | As being a huge fan of React to start and later on become Full Stack, when web apps have more reactive parts, the developer experience becomes enjoyable. From a company's point of view, if something needs to be updated on the website, you dont need the cavalry, you just need your sales person to update the prices as they see fit. It is beautiful to connect and communicate APIs to you app and then "then" it, we can relate it to our way of living. |
| 18. | No, although the gap is closing. I think SSR dynamic sites are the happy medium. |
| 19. | I don't think so, but I'm not 100% sure :) Maybe after Jamstack will be another fresh and cool stack that will be better then Jam. IT is rapidly evolved. |
| 20. | A lot of the problems are being worked on. Like incremental builds in the new Gatsby v3 make large sites with thousands of pages feasible. |
| 21. | I think yes. Most of the web pages out there would be more performant on a JAMstack approach. |
| 22. | It will enable new approaches like render on demand |
| 23. | Yes, I think so. Most sites on the web are lightweight; a problem that JAMstack efficiently tackles. Additionally because most of these sites are built using word press, Jams stack offers a better, incremental, secure solution that is efficient and provides a good developer experience. Ideally, I like to look at JAMstack as an evolution from WordPress. |
| 24. | It’s just so obvious to move from monolith to edge delivered content |
| 25. | JAMStack is separation static from dynamic elements. You can use both worlds easily. |
| 26. | The point is that JAMstack is mainly related to sites, but the ways to interact with user/customers are many and in general is the headless approach that will win or better has already won over monolithic approaches |
| 27. | Still need a dev to get setup, less approachable to non-technical folks |
| 28. | Jamstack is the best alternative for WordPress and I think its use will definitely increase in the future! |