What are the cons of a web based application

Asked 16 years, 3 months ago Modified 14 years, 5 months ago

Viewed 5k times

▲

4

▼

I am going to write a database application for the camp I work for. I am thinking about writing it in C# with a Windows GUI interface but using a browser as the application is seeming more and more appelaing for various reasons. What I am wondering is why someone would not choose to write an application as a web application. Ex. The back button can cause you some trouble. Are there other things that ayone can think of?

database web-applications browser

Share

Improve this question

Follow

edited Sep 24, 2008 at 21:58

Jake McGraw

56.1k ● 10 ● 51 ● 63

asked Sep 21, 2008 at 20:23

Matt

1,902 ● 2 ● 17 ● 19

Be more specific. Give us the situation, and we could tell you what the downsides of YOUR prospective webapp might be.
– Dustman Sep 21, 2008 at 22:49

13 Answers

Sorted by: Highest score (default)

▲
9
▼
✓

There are plenty of cons:

- Speed and responsiveness tend to be significantly worse
- Complicated UI widgets (such as tree controls) are harder to do
- Rendering graphics of any kind is pretty tricky, 3D graphics is even harder
- You have to mess around with logins
- A centralised server means clients always need network access
- Security restrictions may cause you trouble
- Browser incompatibilities can cause a lot of extra work
- UI conventions are less well-defined on the web - users may find it harder to use
- Client-side storage is limited

The question is.. do enough of those apply to your project to make web the wrong choice?

Share Improve this answer

Follow

answered Sep 21, 2008 at 20:40

Dan

63.3k ● 10 ● 66 ● 80

I was going to suggest that HTML5 offline cache means you don't have to have network access to use a web app, nor have responsiveness issues, but then I remembered that IE is always there to ruin the fun. Chrome frame is a patchwork solution for that, but it's not ideal. – Joeri Sebrechts Jul 18, 2010 at 18:15

i was really curious about how drawing can be done in a web app. Is there a built in canvas control of some kind? is drawing done through whatever the user's browser uses as to draw with? I assume directx and opengl are not usable? what is usable? like do web frameworks have the typical listbox, treeview style controls that i would have in WPF? – James Joshua Street Nov 16, 2017 at 3:07

2

One thing that was not mentioned here is the level of complexity and knowledge required to generate a good web application. The problem being unless you are doing something very simple, there is no "Single" knowledge or technology that goes into these applications. For example if you were to write an application for some client server platform.. you may develop in Java or C++. For a complex web application you may have to have expertise in Java, Java Script, HTML, Flash, CSS, Ajax, SQL, J2EE.. etc. Also the components of a web based application are also more numerous, Web Application Server, HTTP Server, Database, Browser.. are tipical components but there could be more.. a client server app is tipical just what it says.. a client application and a Server application. My experience and personal preference is not web based .. web based is great for many things. But even though I am an IT Architect for a leading company that is completely emersed in Web Apps as the solution for everything... The cons are many still.. I do thing the technology will evolve and the cons will go away over time though.

Share Improve this answer Follow

answered Jul 14, 2010 at 13:54

Vito

21 ● 1

1

Essentially the real limitations are only through of the platform, being the browser. If you have to account for all browsers in current use that can be a pain due to varying degrees of standards in each of them.

If have control of the which browser to use, that is everyone is on computers that you control on site, and say you install firefox on all of them, you could then leverage the latest Javascript and CSS standards to their fullest in your content delivery.

[edit] You could also look into options like the adobe integrated runtime or "AIR" as an option allowing you to code the front-end with traditional browser based options like xhtml/css/javascript, flash/flex and have the backend hooked up to your database online, only also providing functionality of a traditional desktop app at the same time.

Share Improve this answer

Follow

answered Sep 21, 2008 at 20:29

ethyreal

3,669 ● 2 ● 23 ● 25

1

The biggest difference and drawback I see with web applications is state management. Since the web is, by nature, stateless every thing you want to maintain has to be sent back and forth from the server with every request and response. How to efficiently store and retrieve it in a matter with respect to page size and performance is hard to do at times. Also the fact that there is no real standard (at least not that everyone adheres to) for browsers makes consistency really..........fun.

Share Improve this answer

Follow

answered Sep 21, 2008 at 23:08

Marcus King

1,677 ● 3 ● 15 ● 20

0

You need to have a network access to the server that you are going to have the web application on (if there are going to be multiple users for the application - which is typically the case).

Actually, there are more pros than cons - if you can give some details about your application, we could help a little more...

Share Improve this answer

Follow

answered Sep 21, 2008 at 20:25

Vaibhav

11.4k ● 11 ● 53 ● 71

0

It completely depends on the requirements of your project. For the most part, there isn't much web applications cannot do these days. Admittedly, certain applications do belong on the desktop as browsers (while currently advancing, and rapidly), still are not quite there yet. From the advent of applications such as Google Docs, Gmail

There isn't much you -cannot- do on the web. If you're creating a World of Warcraft competitor however, the web is most certainly not the optimal solution. Again, unfortunately we'd need more insight on the application you're building for the camp. The best part about the web is that anyone with a browser can use your application.

Share Improve this answer

Follow

answered Sep 21, 2008 at 20:31

anon

0

Web applications delegate processing to a remote machine. Depending on the amount of processing, this can be a con. Consider a photo editor that's a web app.

Web applications also can't deal with a whole lot of data going back and forth to and from a client. You can watch video online.. when it's compressed. It will be awhile before we see any web-based video editing software.

Browser compatibility is also a hassle. You can't control the look-and-feel of the application 100%.

Vaibhav has a good point. What's your application?

Share Improve this answer Follow

answered Sep 21, 2008 at 20:32

easeout

8,736 ● 5 ● 45 ● 51

0

A major one is down time for migrations... users will not expect the application to be down, ever, but realistically it will have to be down for major upgrades. When doing this with a desktop application, the user (or end-user systems admin) is in control of when upgrades happen; with an online app, they're not.

For applications which have large data, performance can be a major problem as you're storing a large number of users' data centrally, which means the IO performance will not be as good as it would be if you gave them all a laptop.

In general scalability gives problems for a server-based app. Desktop applications scale really well.

Share Improve this answer Follow

answered Sep 21, 2008 at 20:37

MarkR

63.5k ● 15 ● 119 ● 154

0

You can do an awful lot with a web-based app, but it is a lot easier to do certain things with a thick client:

- Performance: You get simple access to the full power of the client's CPU.
- Responsiveness: Interactivity is fast and easy.
- Graphics: You can easily use graphics libraries such as DirectX and OpenGL to create fast impressive graphics.

- Work with local files
- Peer-to-peer

Share Improve this answer

Follow

answered Sep 21, 2008 at 20:39

Matt Howells

41.3k ● 20 ● 85 ● 104

0

Deciding whether a web application is a good approach depends on what you are trying to achieve. However here are some more general cons of web applications:

- Real integration with desktop apps (e.g. Outlook) is impossible
- Drag and drop between your app and the desktop / other running apps

Share Improve this answer

Follow

answered Sep 21, 2008 at 22:25

daremon

4,884 ● 2 ● 29 ● 27

0

With a web application, there are more privacy concerns, when you are storing user data on your servers. You have to make sure that you don't loose/disclose it and your users have to be comfortable with the idea of storing that data on your servers.

Apart from that, there are many security problems, like Man-in-the-middle attacks, XSS or SQL injections.

You also need to make sure that you have enough computing power and bandwidth at hand.

Share Improve this answer

Follow

answered Sep 21, 2008 at 22:36

Torsten Marek

86.3k ● 21 ● 93 ● 98

0

"Ex. The back button can cause you some trouble."

You'll have to be specific on this. A lot of people make fundamental mistakes in their web applications and introduce bugs in how they handle transactions. If you do not use "Redirect after Post" (also known as Post-Redirect-Get, PRG design), then you've created a bug which appears as a problem with the back button.

A blanket statement that the back button in trouble is unlikely to be true. A specific example would clarify your specific question on this.

Share Improve this answer

Follow

answered Sep 22, 2008 at 2:03

S.Lott

391k ● 82 ● 517 ● 788

This is pretty irritatingly flippant. You seem to suggest "Oh, of course there is never any problems with the back button, you're an idiot." That's just moronic. Stating the "back button can cause you trouble" is hardly the same has "OMG the back button is impossible to solve!" – BobbyShaftoe Dec 17, 2008 at 23:18

@BobbyShaftoe: sorry you saw my request for details as flippant. I've never had trouble, and I don't know what (other than the standard redirect-after-post) they're talking about. And if they're talking about redirect-after-post, it seems solved to me. I'm asking for help. – S.Lott Dec 18, 2008 at 0:14

0

The back button really is not that much of an issue if you design your application correctly. You can use AJAX to manipulate parts of the current page, without adding items into the browser history (since the page itself wont change).

The biggest issue with designing web applications has to do with state, and the challenges that need to be programmed around. With a desktop application, state is easy to handle, you can leave a database connection opened, lock the record and wait for the user to make the changes and commit. With a web application, you could lock the record...but then what if the user closes the browser? These things must be overcome in the design of your application.

When designing a web application, make sure that each trip to the server "stands alone" and provides a complete answer. Always re-initialize your variables before performing any work and never assume anything. One of the challenges I ran into once was pulling "pages" of grid data back to the user. In a real busy system, with record additions/modifications happening in real time, the user navigation from page to page would vary greatly, sometimes even resulting in viewing the same set of a few records as new additions were added in-front of the query.

Share Improve this answer Follow

answered Sep 22, 2008 at 2:25

skamradt
15.5k ● 3 ● 39 ● 57

Well, that may be true, but there are entire Javascript frameworks for dealing with history and the "back button" under AJAX. – BobbyShaftoe Dec 17, 2008 at 23:20