

NullPointerException - Your questions. Answered.

About my project

What is my project?

My project is a website for developers to ask questions related to all aspects of programming and to answer other developers' questions. Each user's profile will allow them to customise the type of questions they want to answer and will include an option to receive a weekly email with popular questions. The website will not just be a question and answer site but a community for developers to learn and share expertise. It will also include social options such as direct messaging, points for helpfulness, trophies for various milestones and a profile page for each user. This page will help employers to recognise talent which will encourage developers to contribute to the site. The website will also include a page for most common languages with useful links to official documentation and tutorials. If I have time I might build an Android app and a PC program for it to allow a bigger audience.

Why is it needed?

I believe that my project is needed because I believe that all the major programming question and answer websites have issues, but if the best features were taken from each of them, then the resulting site would become a necessity for developers. Another major reason for this project is the lack of a large community for developers to use. Many developers share the same sense of humour and the same way of thinking. The site will also allow employers to advertise jobs and will only show them to developers who would find them relevant. Developers could set their job status to "looking for job" in order to receive notifications when their perfect job comes up.

Who will be interested?

First time developers starting their career to experienced developers pushing the boundaries of their field.

My requirements to complete the project

To develop my project, I will use my laptop as it is powerful and portable as well as my home PCs. The project will be developed using a mixture of PHP (for the backend) and JavaScript (for connecting the frontend with the database). I will be using apache as my web server because it's free, I've used it in the past so I'm familiar with it and it meets all my needs for this project. The finished project will run on a Linux server for several reasons. The first is that Linux is the industry standard for servers due to its stability, low level access and being free. I also use Linux at home which means I'm familiar with how it works. Finally, it works well with all the software I need for the project, such as PHP and apache etc. The website will need a database, so I'll be using MySQL Server. Again, this is because I've used it in previous projects so I'm familiar with it, and because it's second most popular database engine.

User requirements to access / use my project?

Internet connection, web browser and standard PC / laptop. Also, possible Android and PC app.

How am I going to test it?

I am going to be running tests myself to test that the basics are working but I am going to give the website to a class and challenge them to do everything and anything they can to break it as well as test everything they can. This will allow me to test much more than if it was just me. Also, I will use a

NullPointerException - Your questions. Answered.

bot website to put high load on the site to stress test the server to see how it works with high demand.

Section 1 - Analysis

Part (A) - Computational Amenability

My project requires the use of a computer for several reasons, the first being that one of the main ways for users to access it is via the website. This allows for a global audience as anyone with an internet connection and a web browser can access and contribute to it which enriches the experience for the user. Another reason why I believe that my project requires the use of a computer is that it's the fastest method of communication across the world which allow for a much better user experience as they do not have to wait for a physical representation of their communication to be sent manually around the world. Using a computer also allows me to create an intuitive graphical user interface to allow users to easily navigate and use the features of my project. Also, the amount of data that would need to be stored and processed in order for the project to work would be so large that using physical processing methods such as paper would be incredibly slow and inefficient and so a computer is the only viable method of running the backend of the project. Overall, I cannot think of a viable way that a project of this scale and with features such as these could be built any way other than digitally.

Part (B) – Features

Simple breakdown of features:

- Question/Answer system for programmers of all skill levels to give and receive help
- Social features including posting, image sharing, content rating/voting system and a friend/profile system.
- Career features for both employers and employees including a “looking for job” system
- Android app if I have time to make it
- Tutorial pages for common/popular languages and links to documentation

Question/Answer system:

Programming can be very difficult in some situations, no matter how experienced or proficient a developer is. Even professional programmers resort to question and answer websites such as Stack Overflow daily. One feature of my project is a similar system because there are no major alternatives to Stack Overflow and I believe that more features could be added on top of the basic Question/Answer system. The type of questions people receive on their question page can be customised via a settings page on the user's profile. Users will also have the option to receive popular questions by email once a week.

Social features:

NullPointerException - Your questions. Answered.

Due to the lack of a proper community for programmers and developers, and the fact that most people programming for fun or doing it for a career share the same interests and sense of humour, I thought it would be a good idea to include various social features in my site. This will differentiate it from various other programming question and answer sites such as Stack Overflow or Super User and in the process, create a large and unique community for developers all over the world. Social features on the site include direct messaging with other users, profile pages, a dynamic and community ran points system for asking good questions and providing in depth and clear answers to other user's questions, trophies for reaching certain milestones on the site and a posting system for developers to post content which dynamically pushes new, highly rated content to the top of the page.

Career features:

I thought my project needed a career feature as programming jobs can be very specific and it can be hard for employers to target jobs towards the perfect people. The site will include options to help people find their perfect programming job as well as help employers get their jobs seen by the right people. It will include a "looking for job" tag which users can activate when they are looking for a new job. The type of jobs that appear on the user's "jobs" page will be based off one of the settings page on the user's profile which will allow them to enter their requirements and skillset and target viable job offers to them. Career finding features go perfectly with a Question/Answer website as user's can demonstrate their skills by answering questions and collecting trophies and points to prove to any possible employer how good they are. The site will also include extra functions such as uploading their CV to a special part of their profile page and notifications when a likely job is advertised. There will also be an "employer" setting on user's pages to allow employers to advertise new jobs. These features will attempt to match people with their perfect development job so will encourage both employers and employees to use the site.

Android App:

Websites don't always work perfectly on the increasingly popular mobile devices so if I have time after completing my project, I will make an android app for the project with all the same features but offering better performance, appearance and scaling than simply using a web browser on a phone.

Tutorial pages:

Programming can be very difficult to learn, so my project will contain a learning area, complete with tutorial pages for mainstream and popular languages, documentation for said languages, worked example programs and guides written by the community. This will help people to learn new languages and help experienced developers who still frequently have to check something while working. The community written guides and worked examples programs will reward points and trophies making users look more attractive to employers and also encourage the community to create new content to develop the site further. Adding in links to the official documentation for mainstream languages can save developers lots of time.

NullPointerException - Your questions. Answered.

Part (C) – Stakeholders

Categories of user:

There are 3 main categories of user that I would expect to use the website:

- Beginner / Novice developers
- Experienced developers
- Employers

Beginner / Novice developers:

Starting programming can be a daunting task and new developers are always looking for websites that could help them. My project will help new developers in several ways. The first way the website will help new developers is the Question and Answer feature. New developers will be able to ask questions that can be answered by more experienced members of the programming community. These questions can range from logic and general language help to questions about error messages and will provide essential support for novice developers. Another way it will benefit and appeal to new developers is that it includes a tutorial zone, complete with worked examples and links to other places to get help. This will really help people driving to learn how to program, as worked examples are incredibly useful as they show the entire process of creating a program as well as step by step instructions and explanations. The links to the documentation and other places to get help are also useful as they simply show new developers where the resources they need to get help are.

Novice developers would require the site to:

- Be easy to navigate/use
- Contain well explained tutorials
- Help them to learn to program

Novice developers would need the site to be easy to navigate/use because the learning curve in programming is so steep anyway. This would allow them to focus more on programming and less on learning how to use the site that could be the most beneficial to them while programming.

They would also require the site to contain well explained tutorials because if the tutorials were lacking in detail or hard to understand, then there wouldn't be any point in having them. The tutorials and worked examples are on the site to help people learn to program and if they didn't work as intended then they would just be wasting space on the server.

Finally, novice developers would require the site to help them learn to program. This is the main requirement for this group of users as I don't want the site to be just another programming question and answer website, but a community for developers to learn, teach, recruit and communicate.

Experienced developers:

I would expect experienced developers to use the site mainly for reference and for the career features, as well as for using their experience to answer questions. However, I wouldn't be surprised

NullPointerException - Your questions. Answered.

if they still asked questions as new developers are less likely to ask complicated and advanced questions. I believe that the career features of the site are more likely to appeal to experienced developers as they are the most likely to be looking for a job in the programming industry. The links to the documentation for mainstream languages will also be incredibly useful for experienced developers as they are likely to be referring to documentation many times each day.

Experienced developers would require the site to:

- Have an intuitive and effective question and answer system
- Have up to date links to the documentation for a large number of mainstream languages
- Provide a careers system that actively helps experienced developers find the right job

Experienced developers would require the site to have an intuitive and effective question and answer system because I believe that they would be the main source of answers on the site. They have the most experience and knowledge but if the method for asking and answering questions was clunky or unintuitive then they would be less likely to provide answers to newer developer's questions.

Also, experienced developers would require the site to have up to date links to the documentation for mainstream languages because if the links were out of date, or incomplete, then that area of the site would be completely pointless.

Lastly, experienced developers would require the site to provide a careers system that actively helps to find them the right job as otherwise the careers system would be pointless. The careers system needs to automatically do all the difficult work and simply provide users with relevant jobs as soon as they are published. The system should also not provide users with jobs once they are no longer vacant.

Employers:

I think that employers will almost exclusively use the careers features of the site as I believe it to be the perfect way of recruiting the right people for the job. The set of features that will be on the site will allow employers to get their jobs seen by the right developers to avoid wasting time interviewing people unsuitable for the job.

Employers would require the site to:

- Get their job seen by the right people
- Provide analytics to show who their job is seen by and where
- Find a user who is eventually recruited.

Primarily, employers would require the site to get their job seen by the right people as there is a high chance that if the job is seen by suitable candidates then one of them will be recruited to do the job.

Employers would also require the site to provide analytics as to who their job is seen by and where their job is advertised. This would provide them with useful insight, and also prove that they are advertising their job in the right place.

NullPointerException - Your questions. Answered.

Last of all, employers would want the site to find a user who is eventually recruited to do the job. This is the reason employers advertise jobs and if the site fails to do this then why would people advertise jobs on it.

Part (D) – Research and existing solutions

As there are three distinct features in my project (question/answer, careers, social), I researched two examples of each of the features and analysed the good and the bad bits of each one to get a better understanding of how to develop my own solution and what issues current solutions have.

Feature one - Question and Answer:

The most popular and best-known programming question and answer site currently in existence is Stack Overflow (<https://stackoverflow.com>). This is the first existing solution I chose to research. The other existing question and answer solution I chose to analyse was Code Ranch (<https://coderanch.com>). Code Ranch is a less well-known Question and Answer site but seems to be very different to Stack Overflow, so I thought it would be an appropriate choice.

Stack Overflow:

Pros:

I believe that Stack Overflow's biggest asset is its huge community of over 50 million developers, a statistic that it proudly boasts on its homepage. Having such a large community allows every question to have a high chance of being answered by an experienced developer. The huge community also means that there is a massive amount of content on the site as you have a large number of people asking questions all the time.

Another good thing about Stack Overflow is its reputation system. The reputation system in Stack Overflow controls and limits what each user can do which adds a benefit to asking good questions and answering politely and clearly. The system works by locking certain features until a user has a certain reputation. This system acts as both a reward for being helpful, and as a way to combat spam. By locking features such as up and downvoting comments until a moderate amount of reputation has been reached, they are ensuring that people don't create multiple accounts to either increase their own reputation or to decrease the reputation of others.

Stack Overflow also has a primitive career system called "Developer Jobs" which lists both jobs and companies. It filters jobs to the requirements on your profile page. It also features "Featured jobs" which are pinned to the top of the page. I'm not sure if companies have to pay to list jobs on the site although I would assume that they do. One thing that I particularly like about the career system on Stack Overflow is the

Cons:

The homepage of Stack Overflow varies depending on whether you have a user account. Without an account, half of the homepage is taken up by a blue box prompting you to create an account. The rest of the page is filled with a large list of questions that appear to be the most recent questions

NullPointerException - Your questions. Answered.

asked on the site. The homepage is unintuitive and doesn't make it particularly easy to find specific questions to answer. A minor issue I found with the homepage is that the buttons used to filter the questions on the homepage ("Interesting", "Featured", "Hot", "Week" and "Month") require the page to reload when they are clicked. This could be done without a page reload using asynchronous JavaScript (AJAX) which would improve user experience.

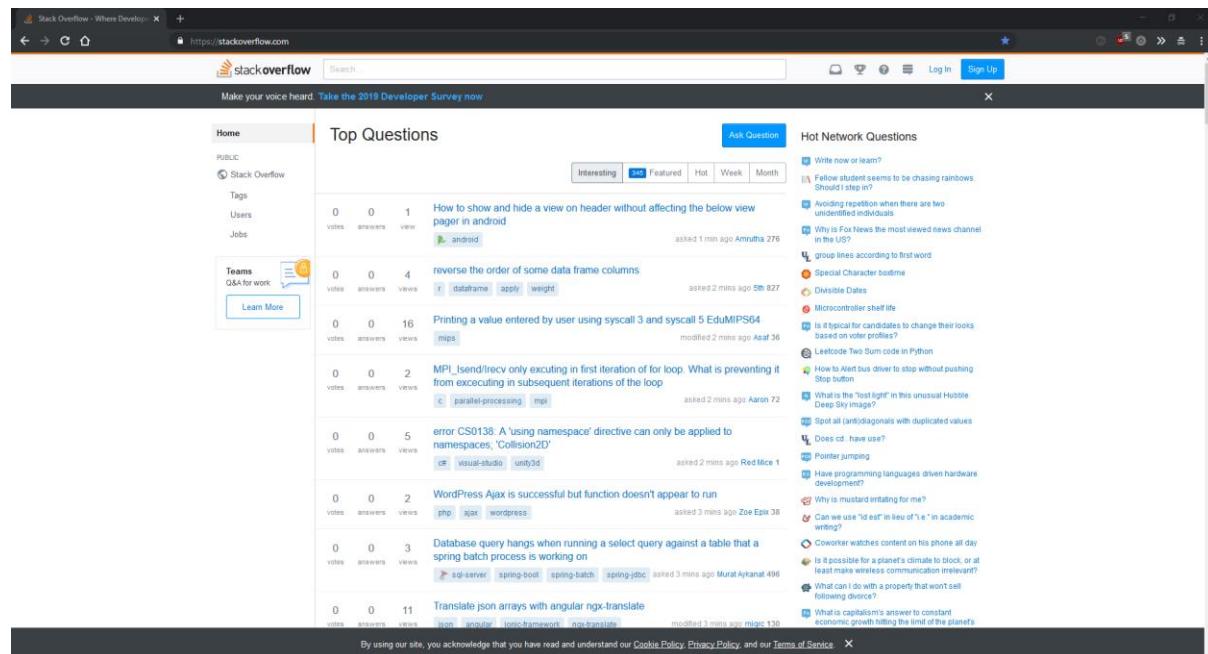
After logging in, the homepage becomes a little bit better as there is no big blue box taking up half the screen. The questions on the homepage are filtered to the languages and topics you have set on your profile, but the homepage is still not very intuitive as the entire page is just a long list of questions. I'm not sure how this system could be improved or how to could be presented better but I think it just looks confusing and messy. Making the homepage look both nice and intuitive could end up being one of the most difficult parts of the entire project.

While the reputation system is good, it does have some issues for newer members. For example, you cannot comment on another user's question or answer without a modest amount of reputation. This can make it difficult for it for newer members are they would have to add a new answer just to say thank you to someone who was helpful. They also can't upvote helpful content such as an answer to their question without a small amount of reputation which they wouldn't be able to do as a brand-new user. This is a more minor complaint but still can affect the general user experience.

Overall:

Overall, I think Stack Overflow is the closest existing solution to the project I'm going to create. It has both a career system and a question/answer system. Its strongest aspect is a large and passionate user base, but the lacks a user-friendly interface and strong social features.

Screenshots:



NullPointerException - Your questions. Answered.

The screenshot shows the Stack Overflow homepage with the 'Top Questions' section highlighted. The page features a sidebar with links for Home, PUBLIC, Stack Overflow, Tags, Users, and Jobs. A 'Teams' section is also present. The main content area displays several questions with their titles, vote counts, and answer counts. For example, the first question is "Why conversion between default system datetime to PersianCalendar is not working?" with 1 answer and 28 views. The second question is "How to compare if a string is in a list and have each item in that list be read in lower()?" with 1 answer and 10 views. The third question is "Any strategies for locking down Jenkins credentials to a shared-library?" with 0 answers and 2 views. The fourth question is "How to return javascript/jquery alert function to python file?" with 0 answers and 5 views. The fifth question is "Convert STR column to Dictionary" with 1 answer and 8 views. The sixth question is "Google Firestore python library not working in AWS Lambda" with 0 answers and 3 views. The seventh question is "Tensorflow: Replicating dynamic_mn behaviour with raw_mn" with 0 answers and 31 views. The eighth question is "Cookies and sometimes sessions disappear on subdomain" with 0 answers and 2 views. The right side of the page includes a 'BLOG' section with news items like "Our 2019 Developer Survey is Open to Coders Everywhere!" and "Data science (ml) January 2019 and views to answers". There are also sections for 'FEATURED ON META', 'HOT META POSTS', 'Watched Tags', and 'Ignored Tags'.

The screenshot shows the 'Ask a Question' page on Stack Overflow. The left sidebar includes links for Home, PUBLIC, Stack Overflow, Tags, Users, and Jobs. The main form for asking a question is visible, with fields for 'Title' (containing "What's your programming question? Be specific."), 'Body' (containing a rich text editor placeholder), 'Tags' (containing "e.g. (windows cmd-server string)"), and a 'Post Your Question' button. To the right, a 'How to Ask' panel provides guidance: "Is your question about programming?", "We prefer questions that can be answered, not just discussed.", "Provide details. Share your research.", "If your question is about this website, ask it on meta instead.", and links to "visit the help center »" and "asking help »". At the bottom, there are footer links for Stack Overflow (Questions, Jobs, Developer Jobs Directory), Products (Teams, Talent, Engagement), Company (About, Press, Work Here), Stack Exchange Network (Technology, Life / Arts), and social media links (Blog, Facebook, Twitter, LinkedIn).

NullPointerException - Your questions. Answered.

The screenshot shows the Stack Overflow Jobs page. On the left, there's a sidebar with links for Home, PUBLIC, Stack Overflow, Tags, Users, and Jobs. Under Jobs, there are sub-links for Teams (Q&A for work) and a 'Learn More' button. The main content area features a banner stating 'Developers first. You'll never receive recruiter spam or see fake job listings on our site.' Below this is a search bar with fields for 'Search all jobs', 'Located anywhere', and a 'Create alert' button. A 'Developer Story' sidebar on the right encourages users to create stories to accelerate matches. The main content displays three job listings:

- Java Engineer - Online & Omni-channel** at Sainsbury's - Manchester, UK. Published 1w ago. Tags: java, spring, hibernate, maven.
- Contract JavaScript developer (small team, in-house products, growth business)** at FLG & Flow XO - Manchester, UK. Published yesterday. Tags: javascript, reactjs, elasticsearch, node.js, redux.
- Junior Test Engineer- Permanent- Manchester City Center** at Sorted Group Limited - Manchester, UK. Published 1w ago. Tags: testing, manual-testing, web-api-testing, continuous-delivery.

At the bottom, a footer bar includes a link to the 'Cookie Policy', 'Privacy Policy', and 'Terms of Service'.

The screenshot shows the Stack Overflow login page. The top navigation bar includes links for Log In and Sign Up. A banner at the top encourages users to take the 2019 Developer Survey. The main content area features a message stating 'Stack Overflow is part of the StackExchange network of 173 Q&A communities.' Below this is a login form with fields for Google/Facebook sign-in, Email (you@example.org), and Password. A 'Log in' button is present. At the bottom of the form, links for 'Don't have an account? Sign up' and 'Are you an employer? Sign up on Talent' are provided. The footer includes the Stack Overflow logo, links to Questions, Jobs, and the Developer Jobs Directory under the 'STACK OVERFLOW' section, and links to Products, Company, and the Stack Exchange Network under the 'PRODUCTS' and 'COMPANY' sections. The footer also includes links for Blog, Facebook, Twitter, and LinkedIn.

NullPointerException - Your questions. Answered.

This screenshot shows Olli Radlett's profile on Stack Overflow. The profile page includes sections for Reputation (86), Badges (5), and Impact (-839). It lists 8 answers, 0 questions, and 12 tags. The answers section shows a list of posts related to Python and Tkinter. The tags section includes python, tkinter, file, eclipse, loops, java, and jdk.

This screenshot shows a question titled "How to compare if a string is in a list and have each item in that list be read in lower()?" on Stack Overflow. The question has 1 answer and 0 comments. The accepted answer provides code to check if a new username is available by comparing it against a list of existing users in both lower() and title() formats. The code uses Python and prints a message if the username is available or suggests another one. The question has been viewed 15 times and is active today.

NullPointerException - Your questions. Answered.

The screenshot shows the Stack Overflow homepage with the title "All Questions". There are 17,088,083 questions listed. The first few questions are:

- Google Firestore python library not working in AWS Lambda
- C# Desktop Application: Code obfuscation: Code obfuscated by Net Reactor was easily deobfuscated.
- How to compare if a string is in a list and have each item in that list be read in lower()?
- Cookies and sometimes sessions disappear on subdomain
- Custom Module - Page not found

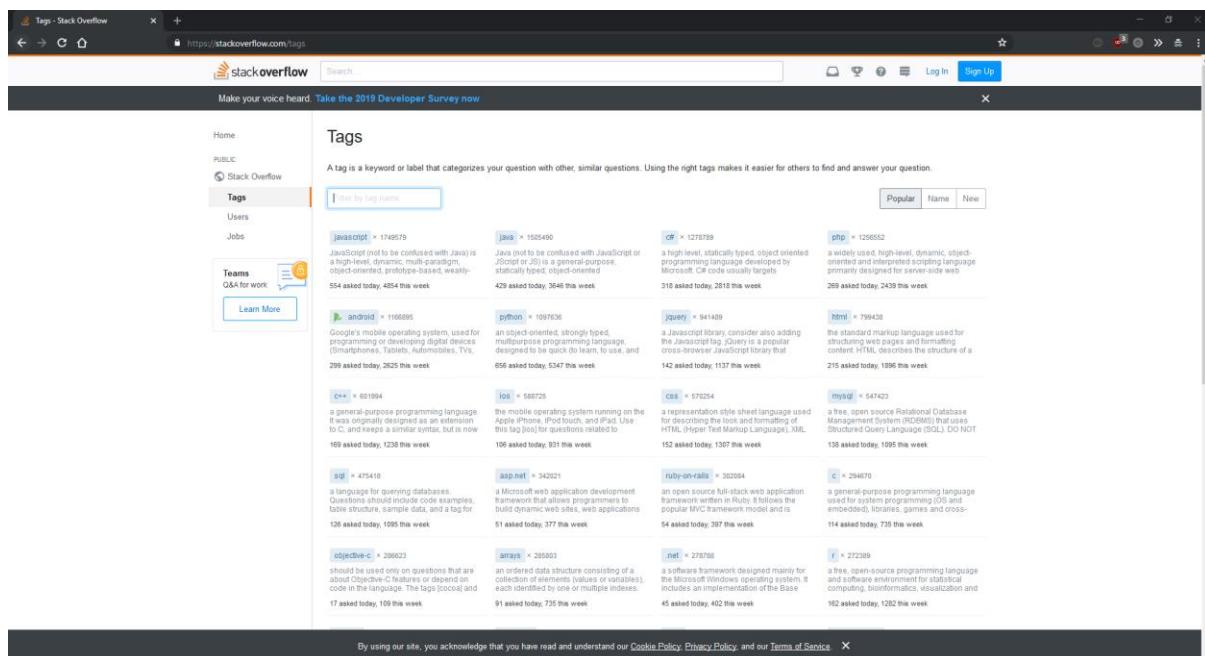
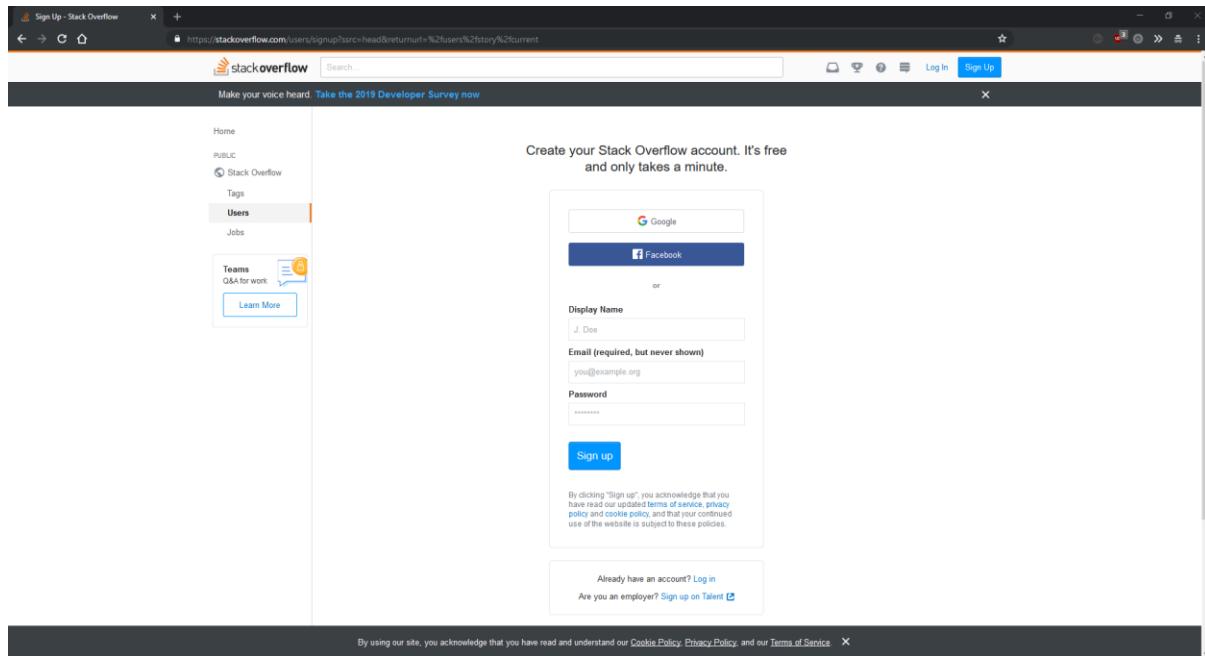
On the right side, there are sections for "Related Tags" (including javascript, java, c, php, android, python, jquery, html, c++, ios) and "Hot Network Questions". A sidebar on the left includes links for Home, PUBLIC, Stack Overflow, Tags, Users, Jobs, Teams, and Learn More.

The screenshot shows the Stack Overflow homepage with the title "Questions tagged [java]". There are 1,506,494 questions listed. The first few questions are:

- Why is it faster to process a sorted array than an unsorted array?
- Why is subtracting these two times (in 1927) giving a strange result?
- Is Java "pass-by-reference" or "pass-by-value"?
- Avoiding != null statements

On the right side, there are sections for "FEATURED ON META" (Take the 2019 Developer Survey, Data science time! January 2019 and views to answers), "HOT META POSTS" (What is the process for proposing a "soft lock" on a question?), "Watched Tags" (java), and "Ignored Tags". A sidebar on the left includes links for Home, PUBLIC, Stack Overflow, Tags, Users, Jobs, Teams, and Learn More.

NullPointerException - Your questions. Answered.



Code Ranch:

Pros:

The homepage of Code Ranch is much stronger than Stack Overflow with a much smaller sign up button and easy to use buttons to navigate to specific languages. There are no questions on the front page which makes it look cleaner.

NullPointerException - Your questions. Answered.

The other good part of Code Ranch is that each category contains a “Best this year” and a “Best this month” column. Whilst this could be move into a better location, it’s a good way of showing the top content without endless hierarchical lists.

Code Ranch also includes a careers system, which consists of forum posts with job offers. This is way more primitive than Stack Overflow’s system, but is better than nothing.

Cons:

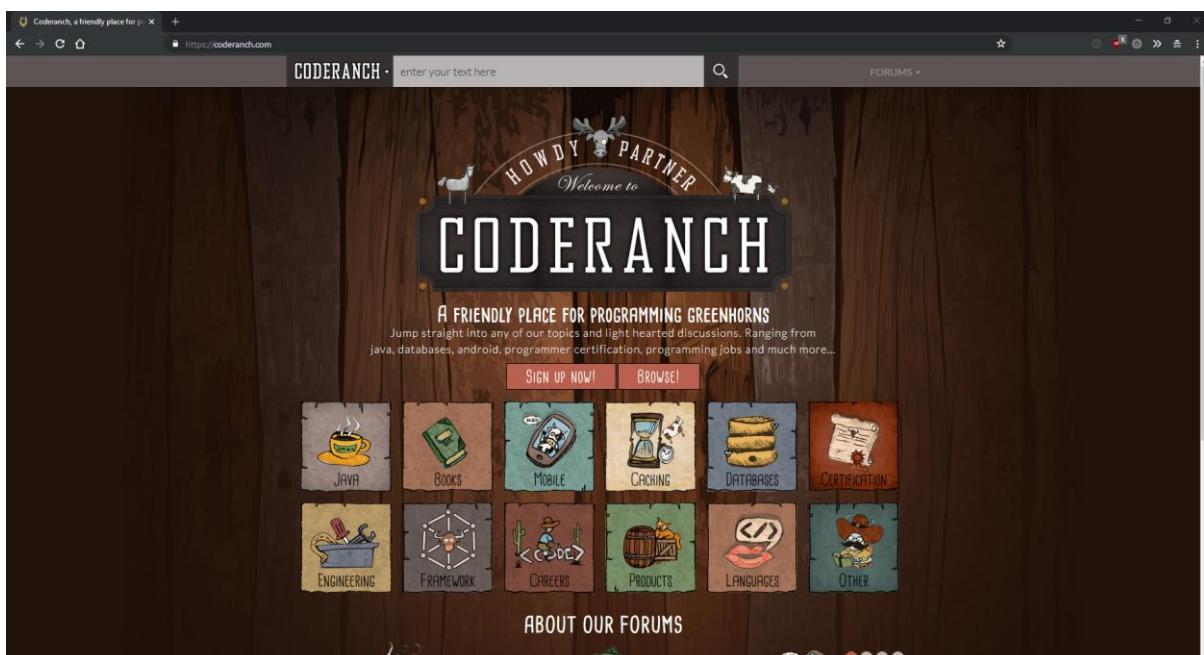
Code Ranch is split up into distinct categories for each topic. Some examples of topics on Code Ranch are “Java”, “Databases”, “Frameworks”. Each category contains subcategories. This keeps the homepage looking nice but makes navigating past the homepage difficult. An example of the subcategories is to get to the page on C++ you have to click on the languages button on the homepage then find the small navigation menu on languages page and click C/C++. Without knowing how to get to where you want to go originally it can be confusing to find things.

Code Ranch takes a more forum style approach to a question and answer site which I don’t think is suitable. The pages aren’t very clear and comprise of 4 to 5 columns with a mixture of questions and posts in a forum style. This makes each page confusing to look at and is surprising after such a well-designed homepage. This is the opposite of Stack Overflow, which suffers from a poorly designed homepage but a good question system.

Overall:

Overall, I think Code Ranch is not as polished as Stack Overflow but has some good features that I might make use of in my project. It suffers mainly from a forum style interface.

Screenshots:



NullPointerException - Your questions. Answered.

The screenshot shows a reply message to a topic titled "Given a LinkedHashSet of anagrams, print each set of anagrams alphabetically and on a new line." The message was posted by Ganesh Patekar on 28/10/2018 at 20:01:59. The message content is: "Congratulations Kevin McKeon."

At the bottom of the message area, there are several checkboxes for message options:

- Disable HTML in this message
- Disable BB Code in this message
- Disable smilies in this message
- Append Signature (Signatures can be modified on "My Profile" page)
- Notify when a reply is posted

Below the message area, there is a "Review message" section with tabs for "Author" and "Message".

The screenshot shows a new topic message on the Big Moose Saloon forum. The message is a book giveaway announcement: "This week's book giveaway is in the Beginning Java forum. We're giving away four copies of Get Programming with Java (MEAP only) and have Peggy Fisher on-line! See this thread for details."

At the top of the message area, there is a "New Topic" header. Below it, there is a "Forum" dropdown menu with the placeholder "... Select a forum from this list ...".

At the bottom of the message area, there are several checkboxes for message options:

- Disable HTML in this message
- Disable BB Code in this message
- Disable smilies in this message
- Append Signature (Signatures can be modified on "My Profile" page)
- Emoticons [What do these icons mean?](#)

NullPointerException - Your questions. Answered.

This screenshot shows the homepage of the Java Forums at the Big Moose website. The header includes links for FAQs, Search, Recent Topics, Flagged Topics, Hot Topics, Best Topics, Register / Login, and a book giveaway announcement for "Get Programming with Java (MEAP only)". The main content area features a sidebar with categories like programming forums, Java, Mobile, Certification, Databases, Caching, Books, Engineering, Micro Controllers, OS, Languages, Paradigms, IDEs, and Build Tools. Below the sidebar are three columns of forum topics: "Best this year" (e.g., "How to deal effectively with short 10 day sprints without feeling pressure?"), "Best this month" (e.g., "How to stay up-to-date with Java?"), and "Most recent" (e.g., "Should I go for the test or?"). A "Forums" sidebar on the right lists categories such as CAREERS, JOBS OFFERED, JOBS WANTED, JOBS DISCUSSION, and SOFT SKILLS.

This screenshot shows a specific forum thread titled "Welcome Peggy Fisher" in the "Beginning Java" category. The thread has 26 replies and was posted 1 hour ago. The post content is: "Welcome Peggy Fisher". The sidebar on the left shows the same categories as the previous screenshot, and the sidebar on the right shows the "Top Forums" section with various Java-related categories like JAVA, Beginning Java, Java in General, NIO2 and Try With Resources, Lambdas and Streams, Jigsaw and Modules, Services, JSP, JSF, EJB and other Jakarta Java EE Technologies, Web Services, Swing / AWT / SWT, Sockets and Internet Protocols, Threads and Synchronization, Performance, I/O and Streams, Other JSE/JEE APIs, Game Development, Distributed Java, Portals and Portlets, SOA and ESB, JNLP and Web Start, and JavaFX.

NullPointerException - Your questions. Answered.

A screenshot of a web browser showing a user profile page. The URL is <https://coderanch.com/u/576791/tempAccountFirstName-tempAccountLastName>. The page has a wooden background with a moose head icon. At the top, there's a banner for a book giveaway in the [Beginning Java](#) forum. The main content area shows a placeholder image for the user's profile picture, followed by the user's name: **tempAccountFirstName tempAccountLastName** (Greenhorn). Below the name, it says "RECENT POSTS BY TEMPACCOUNTFIRSTNAME TEMPACCOUNTLASTNAME". There are three zero values for posts, threads, and cows. A note indicates that the user hasn't created any posts yet. A link to "print one" is provided. At the bottom, there's a "Cows and Likes" section.

A screenshot of a web browser showing a forum topic page. The URL is <https://coderanch.com/t/699878/java/LinkedHashSet-anagrams-print-set-anagrams>. The page has a wooden background with a moose head icon. At the top, there's a banner for a book giveaway in the [Beginning Java](#) forum. The main content area shows a forum topic titled "Given a LinkedHashSet of anagrams, print each set of anagrams alphabetically and on a new line." by Kevin McKeon. The post was made 3 months ago and has 3 replies. The post text asks for help with traversing a LinkedHashSet of anagrams. Below the post, another user, Lilituras Vilida, has responded with a reply. The sidebar on the left lists various programming forums like Java, Mobile, Certification, Databases, Caching, Books, Engineering, Micro Controllers, DS, Languages, Paradigms, and IDEs.

NullPointerException - Your questions. Answered.

The screenshot shows a web browser window displaying the 'Big Moose Saloon' forum at https://coderanch.com/forums/search/search/?fmatch=type-all&sort_by=time&groupByTopic=true&q=java. The page title is 'Search Results: 60000 threads containing 671099 posts'. The main content area displays three search results:

- Convert multiple Java Beans to JSON** in Beginning Java
A post by Junily Laiac from today at 11:58:23. It discusses converting multiple Java Beans to JSON. The message includes a link to the Gson User Guide and a note about handling null fields.
- How to fix/implement equals method in Java?** in Beginning Java
A post by Junily Laiac from today at 11:47:34. It explains how to implement the equals method correctly, mentioning a quote from Samantha Rupasingha about checking if coordinates are equal.
- To get the path from DB on JSP by UseBean** in JSP
A post by Andrey Dmitriev from today at 10:44:45. It asks how to get the path from a database using a JSP useBean. The message includes code snippets and a note from BearBibeault.

At the bottom of the page, there is a navigation bar with links for 'Search', 'FAQs', 'Recent Topics', 'Flagged Topics', 'Hot Topics', 'Zero Replies', 'Register / Login', and a 'View message' link.

The screenshot shows a web browser window displaying the 'Big Moose Saloon' forum at <https://coderanch.com/forums/user/login>. The page title is 'log in or sign up to Coderanch'. The main content area displays a 'Create a new Coderanch account' form:

This week's book giveaway is in the [Beginning Java](#) forum. We're giving away four copies of [Get Programming with Java \(MEAP only\)](#) and have Peggy Fisher on-line! See [this thread](#) for details.

Win a copy of [Get Programming with Java \(MEAP only\)](#) this week in the [Beginning Java](#) forum!

Create a new Coderanch account

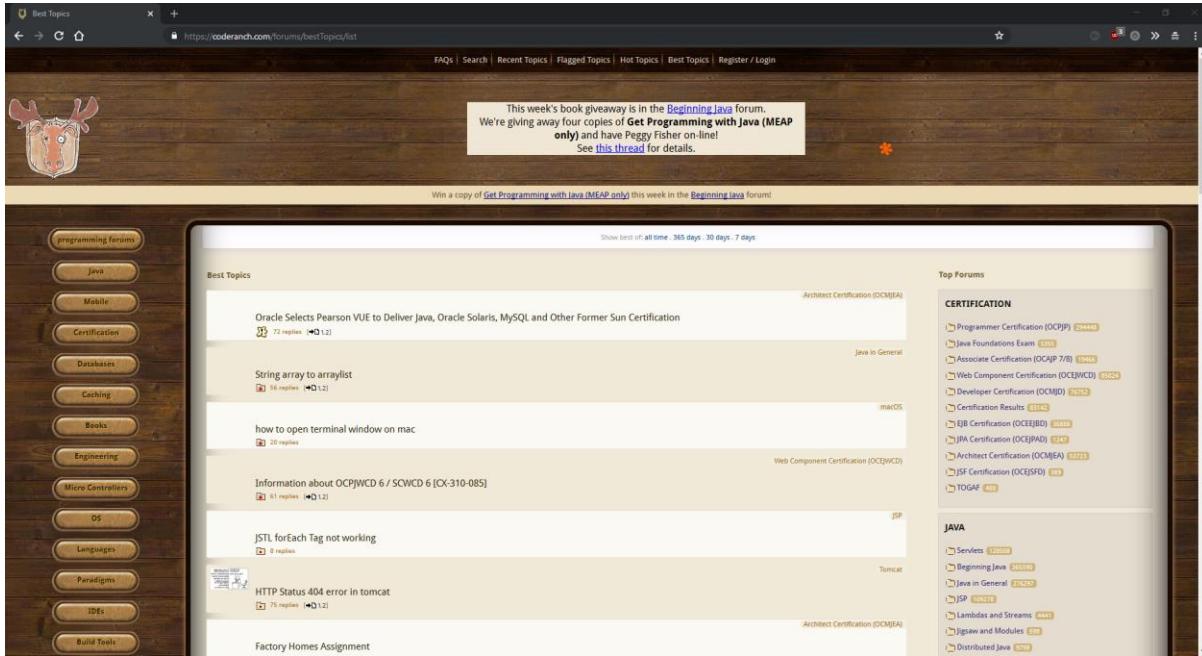
Already have a Coderanch account? [Log in](#)

First name:
Last name:
Email address:
Password:
 Stay logged in
 I lost my password | Activate my account
 Disable autocomplete to please my password manager

I'm not a robot 
reCAPTCHA
Privacy - Terms

I agree to be nice when posting to Coderanch.com | [Sign up](#)

NullPointerException - Your questions. Answered.



Feature two – Careers

To research this feature, I found two career websites, one specialising in programming and one more general job finding site so that I could get a good understanding of how a successful job finding site works. The programming job site I used was GitHub Jobs (<https://jobs.github.com>). The more general job finding site was Indeed (<https://indeed.co.uk>). From my other research, indeed appears to be one of the top job finding websites available.

GitHub Jobs:

Pros:

GitHub Jobs is an extension to GitHub – the most popular open source community so the website is likely to have lots of traffic which means employers are likely to get their job listings seen.

GitHub Jobs mainly consists of two input fields. The first – “Job Description” – allows you to enter keywords for the type of job you are looking for and the second – “Location” simply lets you filter jobs to a specific location. There is also a checkbox for whether you want only full-time jobs to appear. The rest of the page contains a couple of features jobs, a featured company and links to recent popular searches. The website is in the style of a search engine, so it’s intuitive and easy to use.

Employers have to pay \$450 for a 30-day listing so the website is likely to make money.

Cons:

\$450 is a lot of money for a 30-day listing, so some employers might not be able to afford to list on the site. This means that there might not be as much variety on the jobs listed.

NullPointerException - Your questions. Answered.

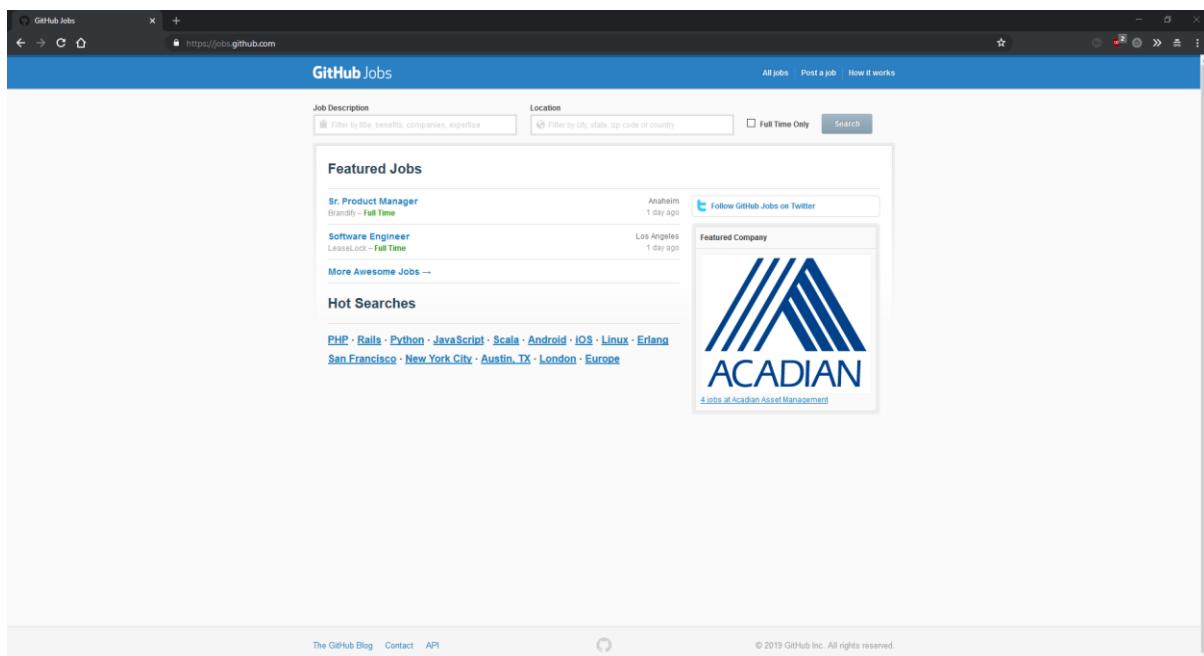
The website is simple but doesn't have any advanced features such as accounts or regular emails with filtered or new jobs. This means that users might miss out on new jobs that have just been published. Even Stack Overflow has regular emails when relevant jobs are published. This is definitely a feature I will add to my project to provide a better user experience.

At the time I accessed the website (16/11/17 – 16:06) there were only 177 jobs on the entire website. This is an especially small number considering that this site is international. This means that there are only 177 jobs listed on GitHub Jobs in the entire world. This is probably the biggest flaw of the site.

Overall:

Overall, the website seems a little basic and the lack of jobs really limits the site. The search engine feature is nice, but the lack of user accounts and email system hold the site back.

Screenshots:



NullPointerException - Your questions. Answered.

The screenshot shows the GitHub Jobs landing page. At the top, there's a header with the GitHub logo and navigation links for 'All jobs', 'Post a job', and 'How it works'. Below the header, a section titled 'How GitHub Jobs Works' explains the process: creating and previewing a listing, paying with major credit cards (\$450.00 per listing), and having the listing go live immediately. It also includes a contact email for questions: jobs@github.com.

The screenshot shows a specific job listing for a 'Sr. Product Manager' position at 'Brandify'. The job title is 'Sr. Product Manager' located in Anaheim. The listing details include:

- Title:** Product Manager / Sr. Product Manager
- Department:** Product Management
- Reports To:** Dept. Head
- Start Date:** ASAP

The job description highlights that Brandify is an industry-leading provider of location-based digital marketing solutions that helps brands connect to their customers. The responsibilities involve assisting in identifying product development needs, establishing and maintaining deep understanding of the product offering, user interface, and technical architecture, and working closely with the operations team to provide product training and implementation support. The job also requires experience with product management processes, including feature prioritization, user research, and market analysis.

Job Requirements:

- Minimum 3-5 years of experience working as a product manager, preferably in a SaaS company within local or regional digital marketing verticals
- Bachelor's degree in Engineering, Business or other relevant field
- Excellent knowledge of agile product processes and related product lifecycle management
- Solid written and verbal communication skills
- Familiar with all Microsoft's business programs
- Takes initiative and strong at following up
- Driven to overcome and succeed
- Exceptional customer advocate and internal ambassador

Confidential Information: This position has access to confidential information such as backend technology and client non-disclosure documents.

Physical Demands: The physical demands described here are representative of those that must be met by an employee to successfully perform the essential functions of this job. Reasonable accommodations may be made to enable individuals with disabilities to perform the essential functions.

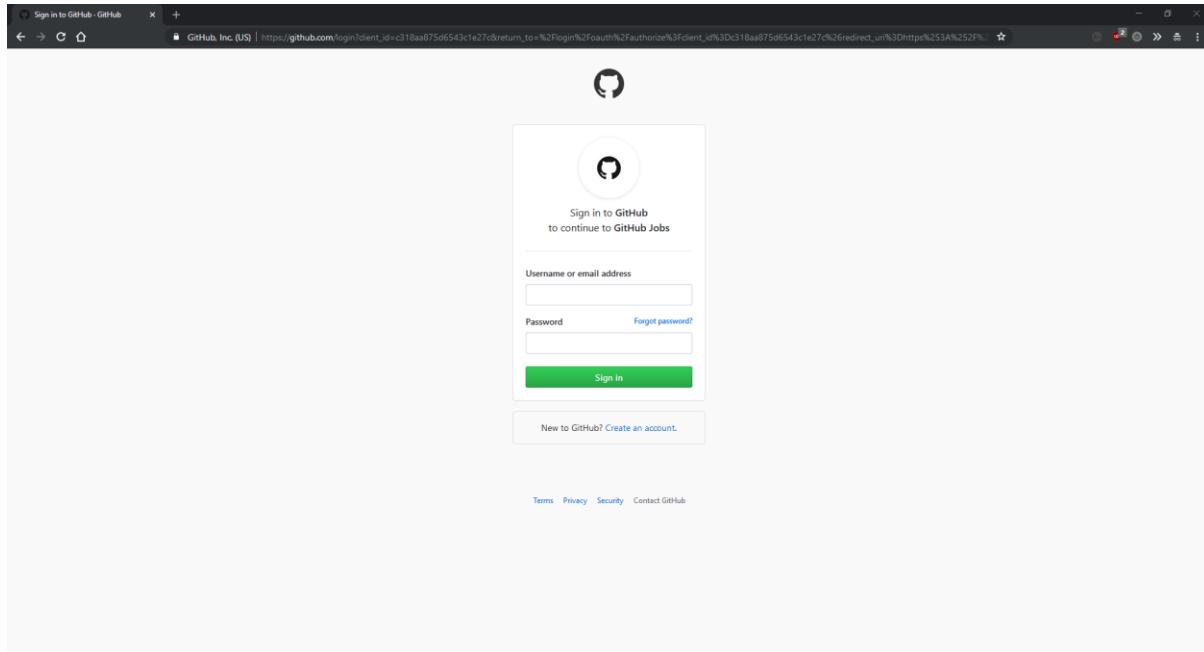
NullPointerException - Your questions. Answered.

The screenshot shows the GitHub Jobs posting interface. At the top, it says '\$450.00 per listing — listings run for 30 days (How does it work?)'. Below that are buttons for 'Create', 'Preview', and 'Publish'. A note says 'Must fill required fields to continue'. The main form includes fields for 'Job title' (with a red asterisk), 'Email (for receipt)' (with a red asterisk), 'Job description' (with a red asterisk), 'Company name' (with a red asterisk), 'Upload Company Logo' (with a red asterisk), 'Company website' (with a red asterisk), and 'How to apply' (with a red asterisk). There is also a 'Browse...' button for logo uploads. A note at the bottom says 'Formatted with Markdown'. At the bottom of the page, there are links for 'The GitHub Blog', 'Contact', and 'API', along with a copyright notice: '© 2019 GitHub Inc. All rights reserved.'

The screenshot shows the GitHub Jobs search results page. The URL is https://jobs.github.com/positions?utf8=%E2%9C%93&description=java&location=. The page title is 'GitHub Jobs'. It features a search bar with 'Job Description' set to 'java' and 'Location' set to 'Hunt Valley, MD'. There is a checkbox for 'Full Time Only' and a 'Search' button. Below the search bar, it says 'Showing 1 - 50 of 147 jobs'. The results list 147 job postings, each with a title, company, location, and posting date. Some examples include:

Job Title	Company	Location	Posted
Lead Software Developer	Naylor Association Solutions	Hunt Valley, MD	about 18 hours ago
Python + Django Software Engineer	Peregrine (Contract) - Full Time	Indianapolis, IN	about 18 hours ago
Human Interface/Front-End Engineer	Apple Inc. - Full Time	Austin, TX	2 days ago
Siri - Software Engineer/Data Scientist	Apple Inc. - Full Time	Cupertino, CA	2 days ago
Siri - Speech Frameworks Engineer	Apple Inc. - Full Time	Cupertino, CA	2 days ago
Senior Full Stack Software Engineer	GoCoder - Full Time	Paris	2 days ago
Softwareentwickler Ruby (w/mid)	teamet GROUP - Full Time	Hamburg	2 days ago
Software Engineer Backend and/or Frontend	adidas - Full Time	Zaragoza (Spain)	2 days ago
Senior Full Stack Web Developer	Datto - Full Time	Norwalk, CT	3 days ago
JavaScript WordPress Developer	ThemeFusion - Full Time	Remote	3 days ago
Senior Software Engineer	Bitcoin - Full Time	Japan	3 days ago
Experienced Frontend or Backend Software Engineer	Pirate Studios - Full Time	Bristol, United Kingdom	4 days ago
Front End Developer	IMPLAN - Full Time	Huntersville, NC	4 days ago
Software Developer		Remote	

NullPointerException - Your questions. Answered.



Indeed:

Pros:

Like GitHub Jobs, Indeed comprises of mainly two input fields in a search engine style approach. The naming of the input fields makes the website very intuitive. The input fields are simply called “what” and “where” which lends itself nicely to the overall design of the page. It is very simple and has no unnecessary details unlike GitHub Jobs which has unrelated “features jobs” and a “featured company”.

Unlike GitHub Jobs, Indeed does not have a shortage of jobs. When I accessed the website (16/11/17 – 16:21) there were 229,691 new jobs added in just the last 7 days. This means that there is a huge choice of jobs which means more people can find the job they want. It’s worth bearing in mind however, that Indeed finds any job, but GitHub Jobs exclusively finds programming jobs which helps to explain some of the difference in numbers.

Indeed doesn’t just allow on employers publishing their jobs to its website, but also appears to search through other job sites, newspapers, associations and company career pages. This is a nice feature but not something I am likely to be able to do for my project.

Indeed does have an account system which allows you to upload your CV. This is a good feature that I was already planning on adding to my project.

Indeed features a company ratings page which allows users to rate their employers as well as salaries lookup page which finds the average salary for various jobs and companies. This is a very nice feature, but it is very unlikely that I will be able to do anything like this as I won’t have the user base or the statistics to do anything worthwhile.

NullPointerException - Your questions. Answered.

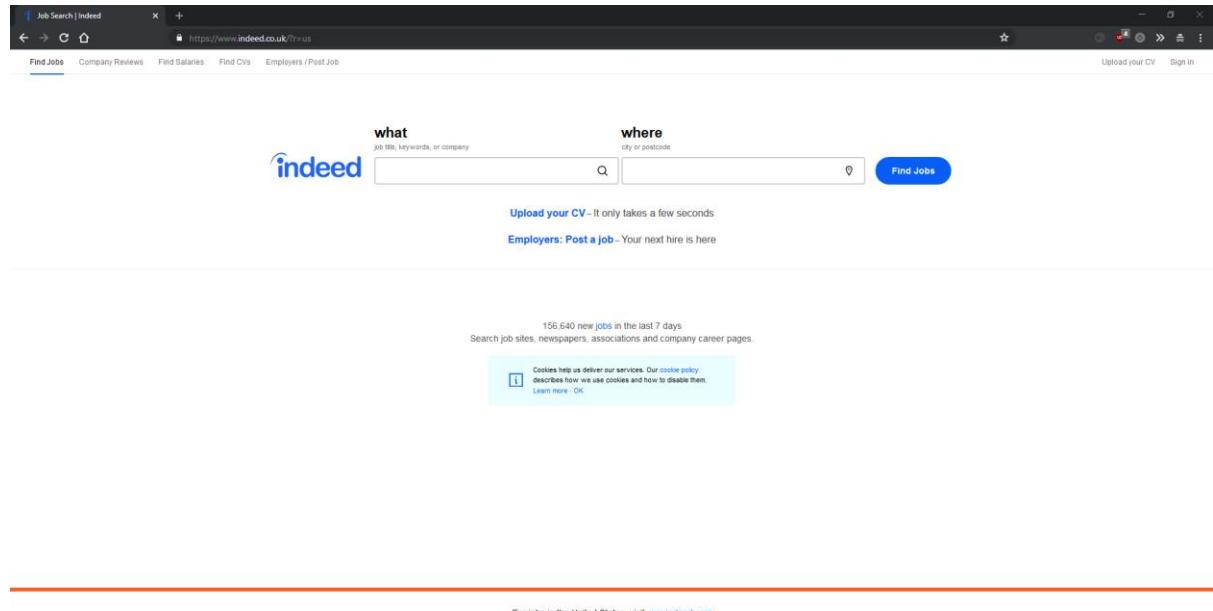
Cons:

The “Advanced Job Search” contains lots of input fields with weird validation rules that could be complicated for some users to use effectively. In my project I am going to try and avoid this kind of overcomplicated interface.

Overall:

Indeed is a much more polished system than GitHub Jobs but contains lots of features that I will be unable to create. It benefits from its extremely high user count allowing it to take averages and present statistics to inform users. In terms of features, it is a much more complete tool than GitHub Jobs. Its only downside is an overcomplicated and unintuitive advanced search menu.

Screenshots:



NullPointerException - Your questions. Answered.

Indeed search results for "Programmer jobs in Derbyshire":

Sort by: relevance - date

Distance: within 25 miles

Salary Estimate: £25,000 - £30,000 a year

Job Type: Permanent

Location: Nottingham NG7, Sheffield, nationwide

Company: Sumo Digital (25)

Job Title: Drone Show Choreography Programmer (C++)

Description: As Verity Studios' drones come to stages worldwide and production volumes rise, we are seeking to complement our productions team with a drone show choreographer.

Responsibilities:

- Program drone motion and light choreographies for world best-known artists and the biggest live events
- Work with the client's creative teams to bring drone show segments to life under tight deadlines, in the office and the event's sites
- Design, develop, and test new drone motion elements and lighting effects
- Contribute to the development of our drone choreography design tools

Qualifications:

- Solid programming ability (preferably C/C++)
- Initiative, independence, and creative thinking
- Understanding of the basic math behind the representation of shapes in 3D space
- Excellent team and communication skills
- Good English skills, written and spoken
- Willingness to travel
- Knowledge of 3D animation and visualization software such as Blender are a plus
- Experience in show programming (lighting, automation, etc.) or artistic background are a plus

Benefits:

Indeed search results for "Programmer jobs in Derbyshire":

Sort by: relevance - date

Distance: within 25 miles

Salary Estimate: £25,000 - £30,000 a year

Job Type: Permanent

Location: Nottingham NG7, Sheffield, nationwide

Company: Sumo Digital (25)

Job Title: Drone Show Choreography Programmer (C++)

Description: As Verity Studios' drones come to stages worldwide and production volumes rise, we are seeking to complement our productions team with a drone show choreographer.

Get new jobs for this search by email

My email: _____

Also get an email with jobs recommended just for me

Send me new jobs

By creating a job alert or receiving recommended jobs, you agree to our Terms. You can change your consent settings at any time by unsubscribing or as detailed in our terms.

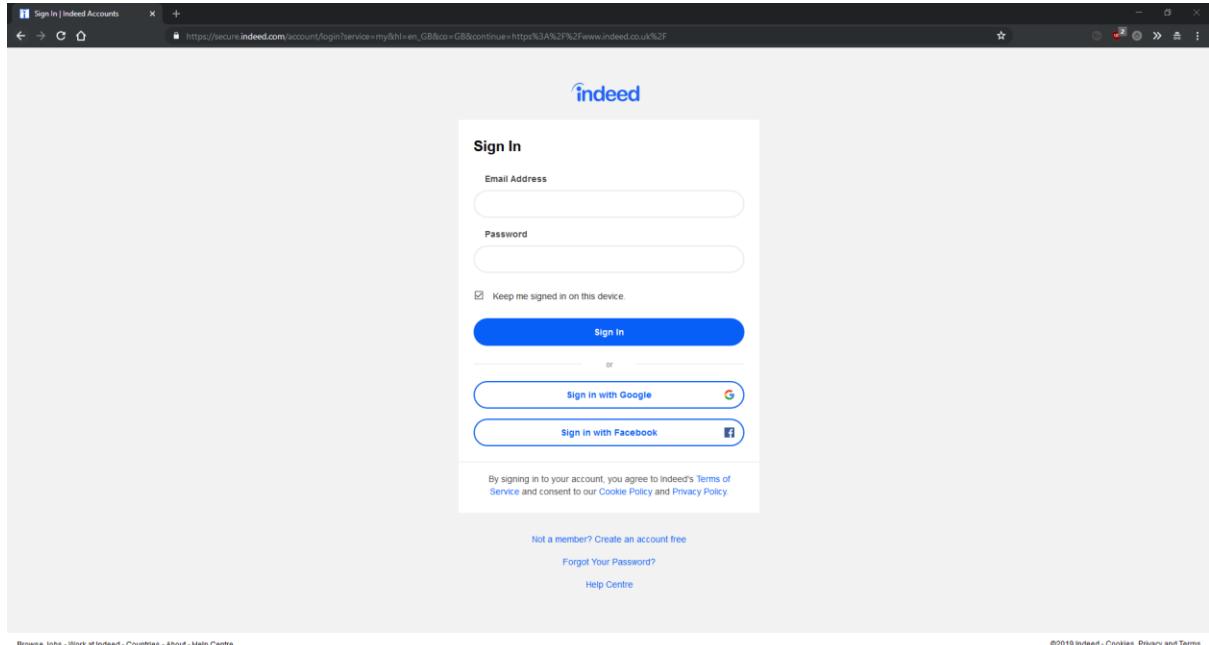
Programmer salaries in England

£34,969 per year

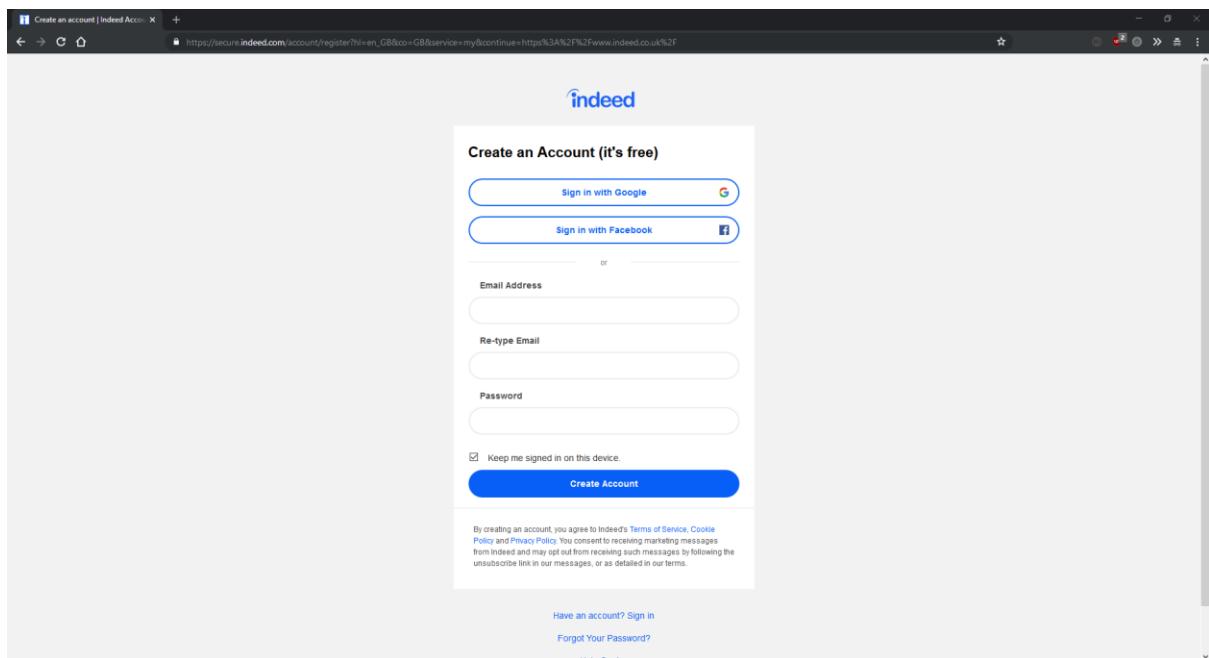
Based on 3,411 salaries

Programmer salaries by company in England

NullPointerException - Your questions. Answered.

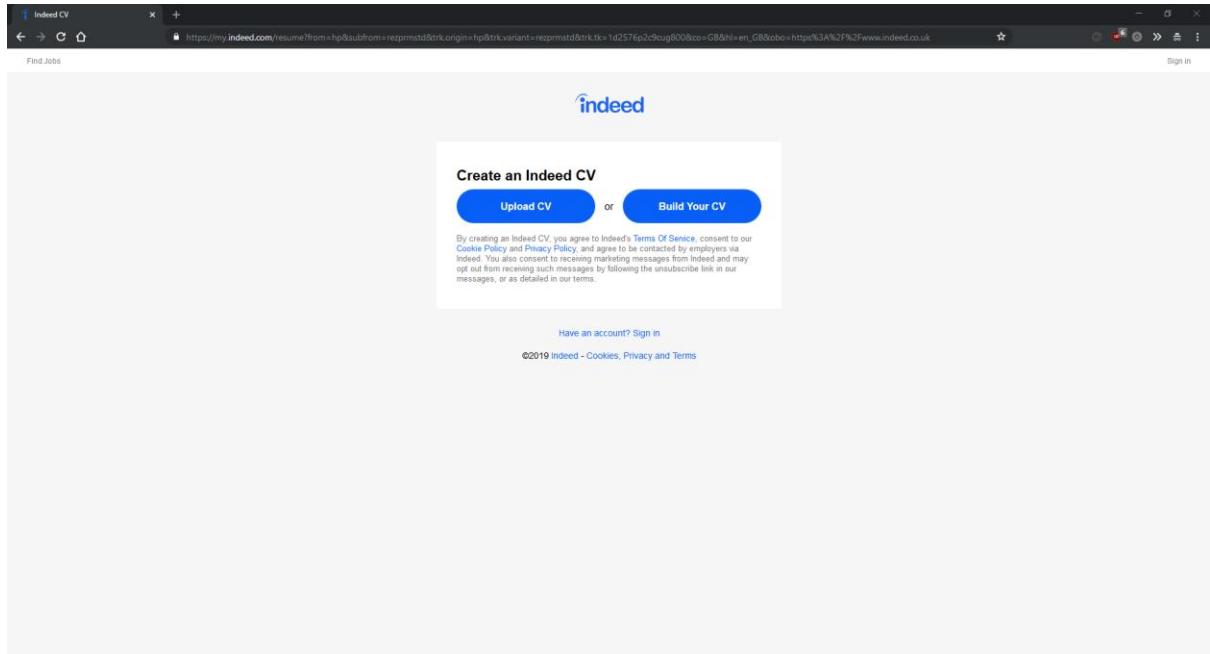


The screenshot shows the 'Sign In' page of the Indeed website. At the top, there are fields for 'Email Address' and 'Password'. Below these is a checkbox for 'Keep me signed in on this device'. A large blue 'Sign In' button is centered. Below it, a horizontal line separates the sign-in section from social login options: 'Sign in with Google' (with a Google logo) and 'Sign in with Facebook' (with a Facebook logo). At the bottom of the page, a note states: 'By signing in to your account, you agree to Indeed's Terms of Service and consent to our Cookie Policy and Privacy Policy.' Navigation links at the bottom include 'Not a member? Create an account free', 'Forgot Your Password?', and 'Help Centre'. The URL in the address bar is https://secure.indeed.com/account/login?service=my&hi=en_GB&co=GB&continue=https%3A%2F%2Fwww.indeed.co.uk%2F.



The screenshot shows the 'Create an Account (it's free)' page of the Indeed website. It features social login options ('Sign in with Google' and 'Sign in with Facebook') followed by a 'or' link. Below this are fields for 'Email Address' and 'Re-type Email'. A 'Password' field is also present. A checkbox for 'Keep me signed in on this device' is available. A large blue 'Create Account' button is at the bottom. A note at the bottom states: 'By creating an account, you agree to Indeed's Terms of Service, Cookies Policy and Privacy Policy. You consent to receiving marketing messages from Indeed and may opt out from receiving such messages by following the unsubscribe link in our messages, or as detailed in our terms.' Navigation links at the bottom include 'Have an account? Sign in', 'Forgot Your Password?', and 'Help Centre'. The URL in the address bar is https://secure.indeed.com/account/register?hi=en_GB&co=GB&service=my&continue=https%3A%2F%2Fwww.indeed.co.uk%2F.

NullPointerException - Your questions. Answered.



Feature three – Social

The closes website I know of to the social features I want to add to my project is Reddit (<https://reddit.com>). Reddit has a huge user base and it splits its content into user created “subreddits”. The features that I want to add in my project that are also in reddit are the direct messaging, the trophy system, the upvote system and possibly something similar to the posting system, although I am undecided about that at the moment. The other website I chose to analyse for its social features was Discord (<https://discordapp.com>) because like reddit, its split up into user created categories (“servers”). The reason I chose two sites with user created categories is because they both have user created programming categories which helped me visualise what kind of social features I wanted to add in my project. This also allowed me to analyse the specific programming communities on this site as well as the site as a whole to see how these differ, and if the site succeeds in providing a community for programmers.

Reddit:

Pros:

Reddit is a huge website with a massive user base. Like all the other websites with a huge user base that I've analysed, this works in its favour. The large number of active users actively creates and drives the content meaning that every day the content on the homepage is different keeping the site dynamic and interesting for the user.

The homepage of reddit looks basic but is intuitive and functional. There is no need for a more modern user interface. Like most websites, the content on the homepage of Reddit changes if you are logged in or not, however the change is more drastic on Reddit. While not logged in, there is the usual “Sign up here” box, although it takes up less of the page than websites such as Stack Overflow. This is a good thing because it makes it easier for people without an account to use the site while they decide if they want to make an account. It also makes the homepage easier to use.

NullPointerException - Your questions. Answered.

When you have created an account, you can subscribe to different user created communities. Your homepage then comprises of the top content from each of your communities every day. This is a nice feature, and is similar to something I was thinking of doing with my project. The feature I am planning on adding is a settings page to customise the languages and types of question/content that appear on your homepage. This will allow users to have as much relevant content as possible to improve their user experience.

Cons:

While not logged in, the homepage of Reddit shows you the top trending user driven content from your country. This is an advanced and technical feature, but I don't believe it is suited to a social type of website. While some users would enjoy looking at content popular in their country, it doesn't display as much variety as a universal feed, and some users might miss seeing content from all around the world.

The general look and feel of Reddit varies significantly on different platforms. The iOS app and Android app are completely different and neither look like the website. I don't like this as they almost feel like different programs

The direct messaging works in more a forum style way which doesn't make it easy and simple to use. The direct messaging system I am trying to make will hopefully be more intuitive.

Overall:

Overall, Reddit specialises in communities and user driven content, its main advantages are its large number of active users and its strong homepage. Some of the features seem unnecessary such as the top content from your country or area, but overall, it's a good example of some of the features I want to add to my project.

Screenshots:

NullPointerException - Your questions. Answered.

The screenshot shows the Reddit homepage with a banner at the top reading "Make your voice heard". Below the banner, there are several post cards. One post from r/news discusses a tentative deal to reopen the government. Another from r/AskReddit asks what the most memorable thing found in trash is. A third from r/UpliftingNews talks about a paralyzed human treated with stem cells regaining movement. A fourth from r/pics shows a cherry blossom tunnel in Japan. On the right side, there's a sidebar for "r/popular" and a section for "TRENDING COMMUNITIES" featuring subreddits like r/Threadkillers, r/Retirementpsychology, r/YouTubeables, r/Documentaries, and r/AnthemTheGame. At the bottom, there's a cookie consent banner and links for the Reddit app, careers, press, blog, help, and more.

This screenshot shows a list of trending posts on the Reddit homepage. The posts include: "Danger Zone - Please change 'Beta' to 'Bravo'" by /u/BetaIsForBabies, "Aah yes, C+++, my favorite soft drink" by /u/lemonade, "Fernando Alonso has defeated Rodney Sandstorm" by /u/lemonade, "Did you hear about the Irishman who was impervious to bullets?" by /u/lemonade, "An African practice of 'ironing' a girl's chest with a hot stone to delay breast formation is spreading in the UK, with anecdotal evidence of dozens of recent cases, a Guardian investigation has established" by /u/lemonade, "This guy's cigarette magic trick" by /u/lemonade, "All this rubbish stopped from going into the ocean by adding a simple net. (Contents at a glance were numerous cans and bottles, 8 ball pond balls, 1 coat hanger, 1 flip flop/thing - right foot size 9)" by /u/HalftimeCove, "Yeah maybe not. I'll just go inside" by /u/lemonade, "For the next 24 hours, anything that you lick becomes yours. What do you do during the lickocalypse?" by /u/HyperHyperObstinate, "Keanu Reeves as Kenshi from Mortal Kombat" by /u/RonnyGerbil, "Cherry Blossom Tunnel in Japan" by /u/BettingBrooke, and "We all made fun of Prof. Oak having to ask if you're a boy or a girl. Turns out he was just well ahead of his time." by /u>ShowersThoughts. The right side of the page features a "reddit premium" section, a "Create your own subreddit" button, and a "account activity" link.

NullPointerException - Your questions. Answered.

This screenshot shows the Reddit 'submit' page. At the top, there's a navigation bar with links like 'DASHBOARD', 'HOME', 'POPULAR', 'ALL', 'RANDOM', 'USERS', 'FRIENDS', 'MOD', 'MESSAGEQUEUE', 'SAVED', 'EDIT', and 'preferences'. Below the navigation is a search bar and a 'reddit premium' advertisement. The main area is titled 'submit to reddit' and has two tabs: 'link' and 'text'. A yellow box contains instructions: 'You are submitting a text-based post. Speak your mind. A title is required, but expanding further in the text field is not. Beginning your title with "vote up" if it is violation of intergalactic law.' There are input fields for 'title' (with character count 0/300) and 'text' (with placeholder '(optional)'). Below these are sections for 'choose where to post' (radio buttons for 'Your profile' and 'A subreddit'), 'subscribed subreddits' (a long list of subreddits), and 'options' (checkbox for 'send replies to my inbox'). At the bottom is a note: 'please be mindful of reddit's content policy and practice good reddiquette.'

This screenshot shows the Reddit homepage ('r/popular'). At the top, there's a banner with the text 'Make your voice heard' and 'Vote and be a part of what the Internet is talking about!'. Below the banner, there's a 'Sign in' modal window. The main content area shows several trending posts: 'Lawmakers', 'Garbage in the trash', 'First pants now reg', and 'Cherry Blossom Tunnel in Japan'. On the right side, there are 'SUBSCRIBE' buttons for various subreddits. At the bottom, there's a cookie consent message: 'Cookies help us deliver our Services. By using our Services or clicking I agree, you agree to our use of cookies. Learn More.' with 'I AGREE' and 'Privacy Policy' buttons, and a copyright notice: '© 2013 REEDIT, INC. All rights reserved.'

NullPointerException - Your questions. Answered.

This screenshot shows a Reddit post titled "Google asks Supreme Court to overrule disastrous ruling on API copyrights" from the r/programming subreddit. The post has 2,091 points and was submitted 19 hours ago by user blerkut. It has 330 comments. The post discusses the absurdity of copyrighting APIs and the fragmentation of Java APIs. A user named Gamemality responds with a detailed explanation of the issue. Another user, roothorick, adds a link to a JavaGiant article. A user named ZehulanRacanahan points out that works from 1923 entered the public domain at the beginning of this year. A user named ia_it_controversial notes the irony of corporate lobbying. A user named paulgrnt999 expresses admiration for works from 1983. The post ends with a note that the lifetime of the author precludes making other uses of the work during the lifetime of the entertained. The sidebar on the right shows statistics for the programming subreddit, including 1,608,462 readers and 2,175 users now. It also features a "Get Reddit Premium" button and a "Guidelines" section.

This screenshot shows the front page of Reddit. The main banner says "Make your voice heard" and encourages users to vote and comment. Below the banner, there is a sidebar with various subreddits and their counts: r/news (42.9k), Lawmak (22.0k), r/AskReddit (11.0k), r/Uplift (10.0k), and r/pics (16.5k). The main content area shows a large image of a cherry blossom tunnel in Japan. A modal window is open, prompting users to "Join the worldwide conversation." It asks for an email address and provides a "NEXT" button. The sidebar on the right includes links for "GET PREMIUM", "SUBSCRIBE", and "SIGN UP". The footer contains legal and privacy information, including a cookie consent notice and links to "About", "Careers", "Press", "Advertiser", "Blog", "Help", "The Reddit App", "Reddit Coins", "Reddit Premium", and "Reddit Gifts".

NullPointerException - Your questions. Answered.

The screenshot shows a list of 20 posts from the 'Programming' subreddit. The posts cover various topics related to NullPointerException, such as Supreme Court rulings, basic Java concepts, and specific code snippets. The interface includes a sidebar for 'programming' with a 'Get Reddit Premium' button, a 'Guidelines' section, and a 'Info' section with subreddit statistics and rules. The main content area shows post titles, submission times, and a snippet of each post's content.

Post Title	Submitted By	Post Content Snippet
Google asks Supreme Court to overrule disastrous ruling on API copyrights	(antechnica.com)	submitted 19 hours ago by antechnica
How BASIC was started	(poludna.com)	submitted 2 days ago by poludna
Crypto failures in 7-Zip	(theadreaderapp.com) <	submitted 1 day ago by Leungc
Apple is indeed patenting Swift features	(forums.swift.org)	submitted 1 day ago by Koenigstein
Fourier Transform drawing with two functions	(youtu.be)	submitted 14 hours ago by XallBombe
Software Patents Should Require the Code to be Open Source	(tortugatech.wordpress.com)	submitted 18 hours ago by grusenoff
Advanced Wildcard Patterns Most People Don't Know	(appodeeles.com)	submitted 18 hours ago by grusenoff
CockroachDB's Consistency Model	(cockroachdb.com)	submitted 13 hours ago by grusenoff
Practical Deep Learning for Coders 2019 - fast.ai	(fast.ai)	submitted 5 hours ago by cheefuwhy
It's Time to Move on from Two Phase Commit	(dmamusinga.blogspot.com)	submitted 1 hour ago by dmamusinga
Who is responding with HTTP Status 418 "I'M A TEAPOT" in one of their projects?	(httpstatususe.com)	submitted 20 hours ago by adamavryola
Desktop to Mobile - Developing for Multiple Platforms without Losing Your Mind	(youtube.com)	submitted 2 hours ago by palem
Altair becomes environment friendly	(XKojMedia) <	submitted 12 minutes ago by imstache
Finally created Arkit: 360 Degree Video Portal #xcode #swift	(m.youtube.com)	comment share save hide give award report crosspost

Discord:

Pros:

Discord is primarily an instant messaging and VOIP solution, but some of the features are good examples of things I am going to add in my project. The instant messaging system in discord is fluid and intuitive and features a simple markdown which is another thing I'm considering adding to my project. For example, if you surround a String with a * on each end, the text is formatted in italics. This removes the need for formatting buttons which can be messy, and is also fast for experienced users. This markdown could be very beneficial for my project, and I will probably add it in my own project, extending the markdown to include font sizes (big, medium, normal), code markdown (with language specific syntax highlighting) and more. This will not only make the site more user friendly, but improve the general look and feel of it, making it look more professional.

One feature that discord does that most people wouldn't even glance at, is the messages coming through when a message is sent, without the need to refresh the page. This is expected of every single instant messaging application, and the vast majority of people wouldn't even consider it. However, it will require some thought when it comes to my project, as the majority of the backend will likely be coded in PHP, which is server side and backend only. This means that it does everything on page load, and cannot modify the contents of a page on demand. Doing this will require the use of AJAX (Asynchronous JavaScript and XML), which not only allows me to poll the server for messages every x seconds, but also allows me to dynamically refresh certain parts of the webpage without reloading the entire page. This means that I could have the message feed <div> refresh itself every x seconds to check for new messages, without having to reload the page. Whilst this is expected from every single Instant Messaging solution, it is still a major consideration when designing any messaging application. Discord does this very well, as its primary objective is to be an instant messaging program.

Discord also supports the use of user created bot accounts. These can perform a multitude of tasks, but most useful of all is auto-moderation. Discord provides an API for simple integration with their

NullPointerException - Your questions. Answered.

services, and users can write bot accounts to automate tasks. Obviously I wouldn't create a public API with full access to my site, as it wouldn't make much sense, and it would have security issues. However auto moderation could be a useful addition to my site, and research into existing discord auto moderation bots would provide me with the exact features required by an auto moderator.

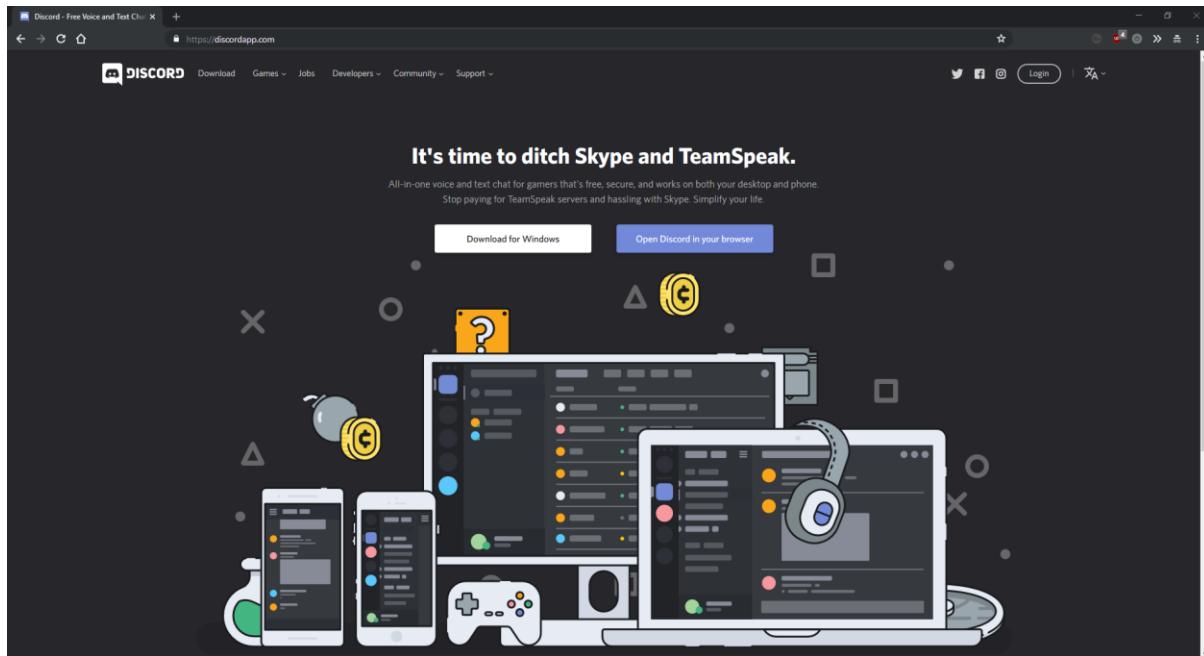
Cons:

Discord has great instant messaging features, but is a completely different site to mine. While it does most of its intended features very well, it would be pointless to talk about them here, as they are so dissimilar to the intended features of my site.

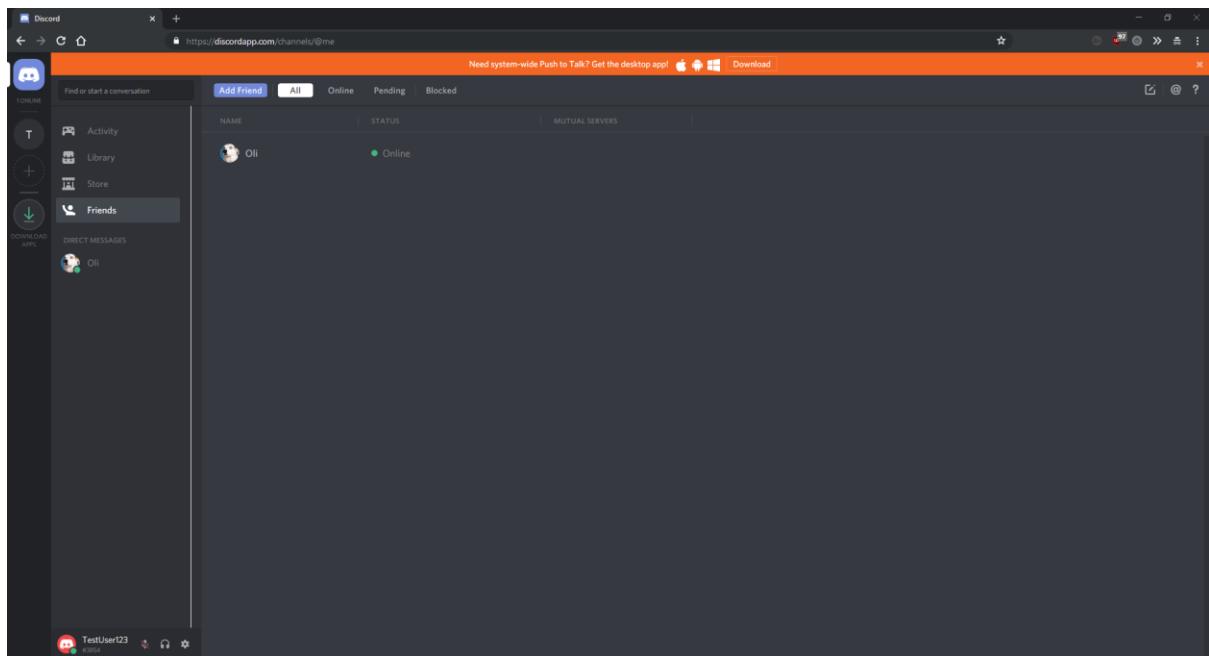
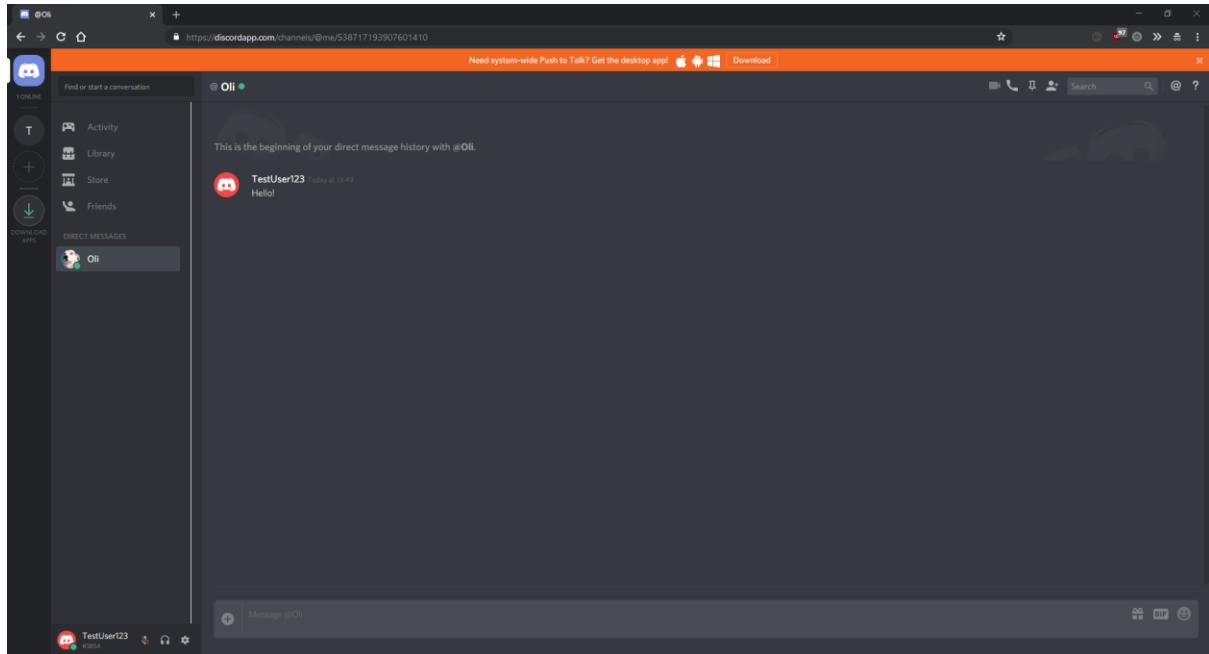
Overall:

Whilst Discord is a completely different site to mine, with a completely different purpose and design, some of its features, such as its markdown and instant messaging, could be very useful in my project, and have given me some good ideas.

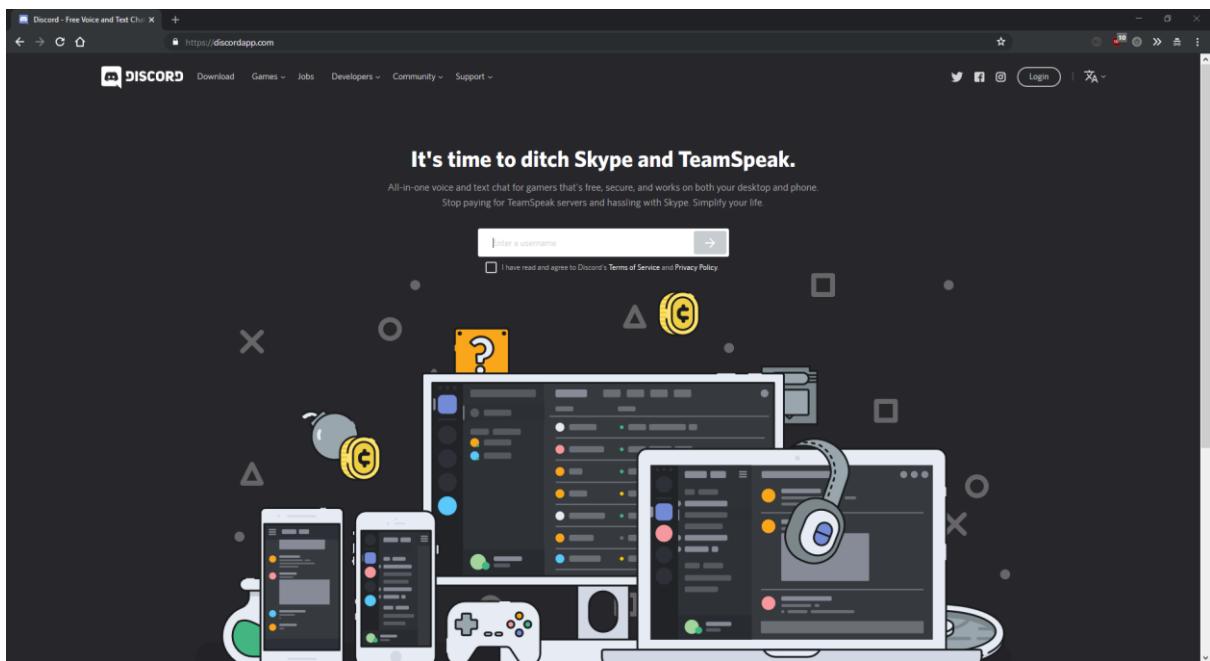
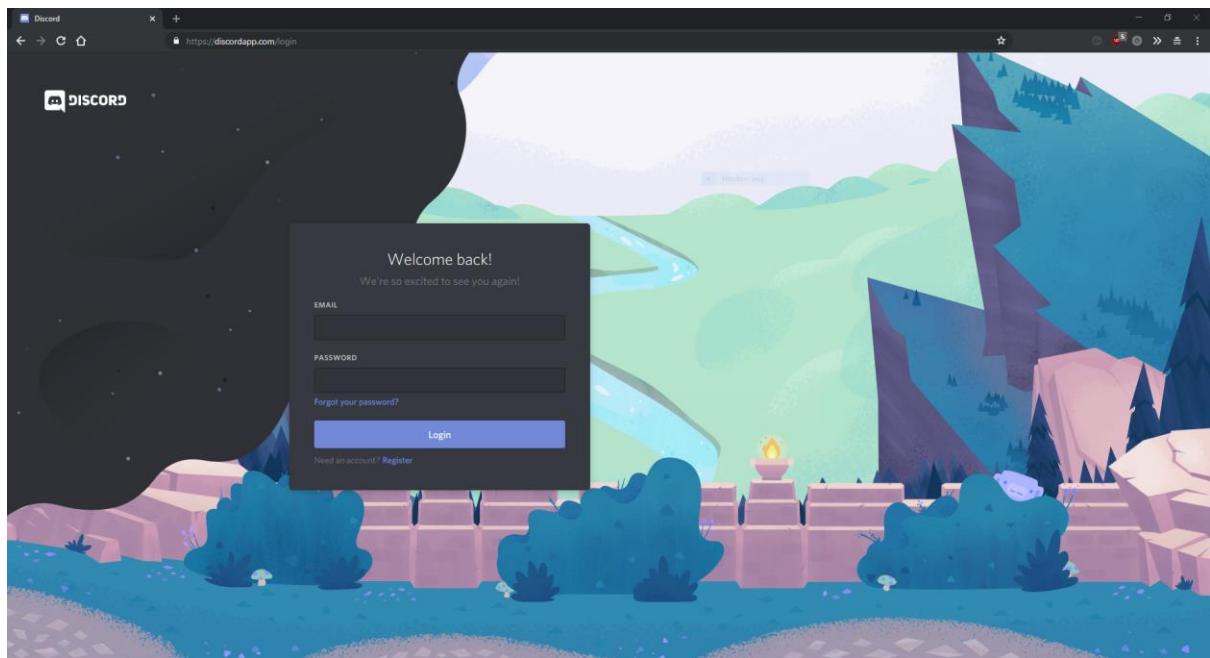
Screenshots:



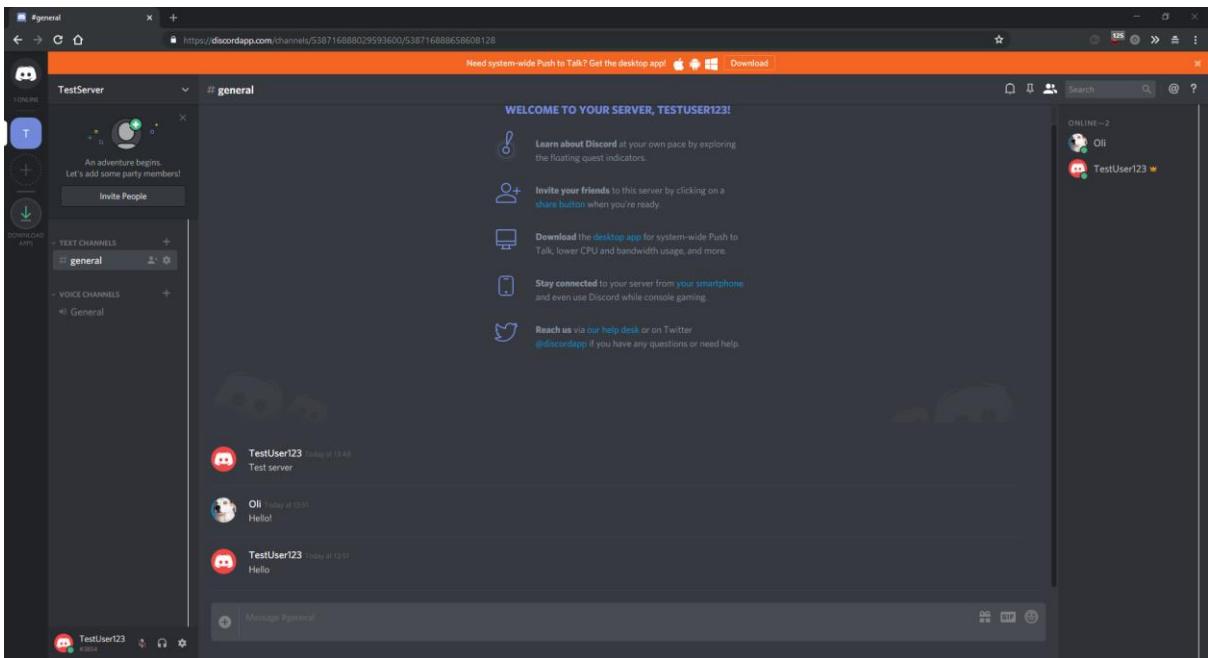
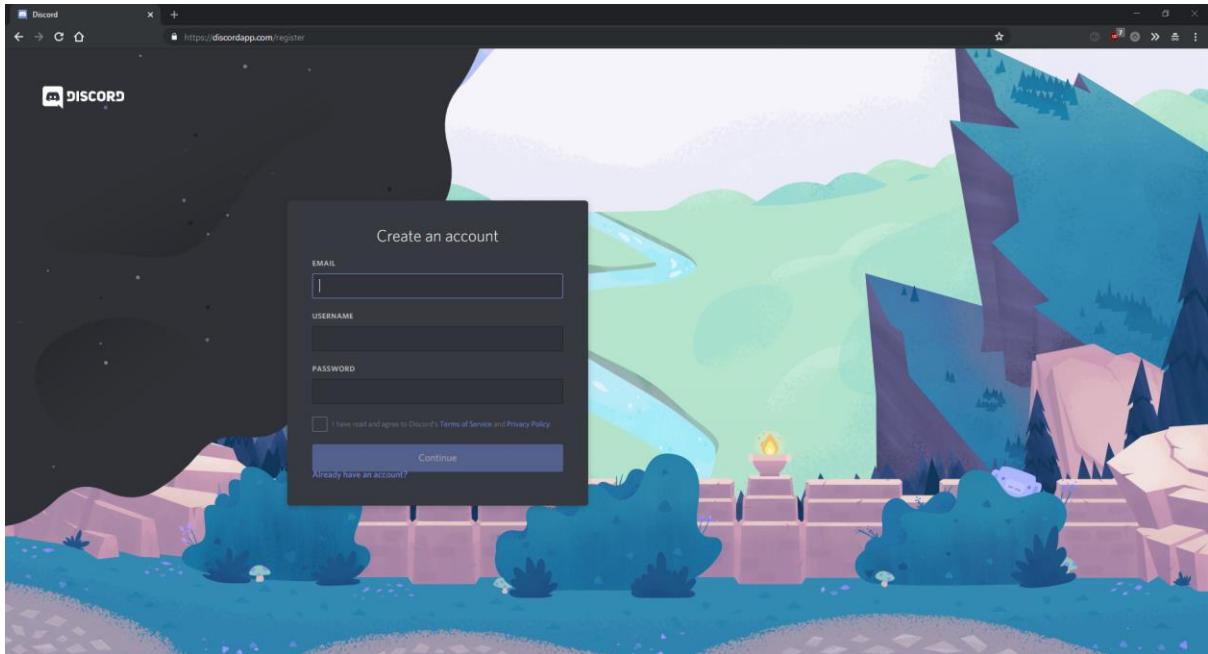
NullPointerException - Your questions. Answered.



NullPointerException - Your questions. Answered.



NullPointerException - Your questions. Answered.



NullPointerException - Your questions. Answered.

The screenshot shows the 'TestServer' settings page in the Discord app. The left sidebar has sections for 'TESTSERVER' (Overview, Moderation, Audit Log, Roles, Integrations, Emoji, Webhooks, Widget), 'USER MANAGEMENT' (Members, Invites, Bans), and 'Delete Server'. The main area is titled 'SERVER OVERVIEW' with a placeholder image for the server icon. It includes fields for 'SERVER NAME' (TestServer), 'SERVER REGION' (Western Europe), 'AFK CHANNEL' (No AFK Channel), 'AFK TIMEOUT' (5 minutes), and a 'NEW MEMBER MESSAGES CHANNEL' dropdown set to '# general'. Below these are 'DEFAULT NOTIFICATION SETTINGS' with options for 'All Messages' (checked) and 'Only @mentions'.

The screenshot shows the 'My Account' settings page in the Discord app. The left sidebar has sections for 'USER SETTINGS' (My Account, Privacy & Safety, Authorized Apps, Connections, Billing, Discord Nitro, HypoSquad), 'APP SETTINGS' (Text & Images, Appearance, Streamer Mode, Language), 'Change Log', and 'Log Out'. The main area is titled 'MY ACCOUNT' and shows the user's information: USERNAME (TestUser123#3854) and EMAIL (redacted). There is an 'Edit' button next to the email field. Below this is a section for 'TWO-FACTOR AUTHENTICATION' with a link to 'Enable Two-Factor Auth'.

NullPointerException - Your questions. Answered.

Part (E) – Requirements

My Requirements

In order to create my project, I will require a computer/laptop to do all the programming on. I will need full admin access on this computer, as well as no internet restrictions. It will also need to be fairly powerful as it will have to run several demanding programs simultaneously, as well as having to compile large amounts of code without a large wait. It will need at least 8GB of RAM and a decent modern CPU. I am going to use my personal laptop to fulfil this requirement, as it is more portable than a desktop computer, which will allow me to work both at home and at school. It is also powerful which means that it can multitask effectively and run demanding software with ease. Lastly, it is already set up with all the programming software and runtimes that I will need to program my project. This will save time and allow me to concentrate on the actual development. The only issue is that the school network blocks some websites that I will need to access to carry out my project. To solve this, I have set up a VPN on my laptop to allow me unrestricted internet access, even on limited access networks.

I will also require a large number of people to test the robustness of my project to test for bugs. For this, I will use a class of about 30 younger students and ask them to explore every feature and function to test for bugs. This is a far faster way of testing for bugs than going through the entire site myself, and clicking every button and testing every type of input. This will save me a vast amount of time and allow me to spend more time and effort actually developing the solution rather than just testing.

I will also need a remote Linux virtual machine with the specifications below:

- Ubuntu server
- At least 2GB of RAM
- Decent CPU
- Enough storage space to archive content and store a large database (Approx. 20GB)

The remote computer will require the software below:

- MySQL Server
- apache
- openssh
- git
- PHP7
- PHP MySQLi extension

To fulfil this requirement, I will use a Digital Ocean remote virtual machine running Ubuntu server.

User requirements

To access my website, users will require a computer/laptop with an internet connection and a web browser. To improve the user experience, their screen should have an aspect ratio of 16:9 in order to display the content in the best way. Similarly, the screen should be at least 720p. If I have enough time to develop an android app, the user will require a minimum of Android Marshmallow. I will be

NullPointerException - Your questions. Answered.

able to provide further information about user requirements after I have finished developing and optimising the project.

I aim to support most popular web browsers, but Internet Explorer requires “interesting” ways of doing things meaning I would have to code twice for almost everything. For this reason, I probably won’t support Internet Explorer. Other than that, however, I plan on supporting most popular web browsers.

Initial user requirements

Website

- Modern web browser with JavaScript support (e.g. Firefox, Google Chrome, ...)
- Reasonably good computer/laptop/phone (Speed of computer only required to run browser, not website)
- Stable internet connection

Android app

- Reasonably good Android phone running at least API level 23 (Android Marshmallow)
- Modern web browser app with JavaScript support (e.g. Google Chrome)
- Stable internet connection or mobile data connection

Part (F) – Limitations off my proposed solution

Limitations that I can't solve

All the limitations with my project that I cannot solve involve the user base to some extent. The number of users visiting my site is not something I can directly control, and therefore have the biggest impact on my site.

The biggest limitation of my project will be the user base. Websites like StackOverflow and Reddit thrive off their huge active user base, while it is unlikely I will ever have any users at all, since I am spending no money or time on advertising or increasing product visibility. This means that features like my “hot” questions will have very little impact on the site, as they require a large and active to continually make and vote on content. Limitation caused by the lack of a large user base is that my website relies on user driven content, which means that without users, there will be very little content on my site. For development purposes, I will create some fake user accounts and fake posts to demo the site, but the lack of users means that my website would never be able to function as a business.

Another limitation to my solution is time. The tutorial section of my website would take me a large amount of time to simply write all the tutorials, sample programs, and to gather links. Because of this it is likely that I will only support a few of the more common languages to demo the functionality, but if I had more time, or more developers working on the project with me, I would support at minimum all the mainstream languages, and preferably more.

Another limitation to my project is that, in the unlikely event that my project gains traction and gets a large number of users, my server wouldn’t be able to cope with the high demand caused by lots of network requests. This would be easily fixable but would cost a lot of money, which would mean I would have to take a more business oriented approach, instead of just making a website I would enjoy using.

NullPointerException - Your questions. Answered.

Limitations that I can solve

Most of the limitations with my website that I will be able to solve simply involve spending more time on various features that would often a separate developer. This includes things like security and database optimisation.

One limitation that I can solve is security. Hobby projects and other small websites often ignore security issues in order to focus on UI and UX (User interface and User Experience). However, my preferred sector of programming is backend, which means I will spend the time securing my website to the best of my ability, including password hashing, external database configuration file, and IP (Internet Protocol) Address blacklisting. This will differentiate my website from other small projects of a similar nature, and give potential user's the confidence to use my site.

Another fixable limitation of my site is the speed of the database. While I am unable to use a powerful server for my project, I can at least ensure my database design complies with popular database principles including ACID (for reliability and constancy), and normalisation (For speed and efficiency). This will ensure that I make the most of the hardware available to me, and I have no bottlenecks caused by lazy database design and use.

Part (G) – Success criteria

The main success criteria of my project is: "Can I produce a useable and intuitive solution that provides users with an alternative to StackOverflow, while providing additional social and career features not found in mainstream programming sites".

The success criteria for my project are:

- User friendly and intuitive
- Functional and robust question and answer system
- Functional and professional career system
- Functional tutorial zone
- Fast and reliable database and backend
- Secure backend

User friendly and intuitive:

My website should be as user friendly and intuitive as possible, allowing users of all skill levels to be able to interact and use my site with ease. This can be quantified by testing the finished prototype with a class of a lower year, to assess the interface and how user friendly it is. The test will probably involve a survey and a task list for the students to complete, where the more tasks completed, the more user friendly the website is.

Functional question and answer system:

The question and answer system is probably the largest and most important aspect of the site, and having a functional and robust question and answer system is almost certainly the most important

NullPointerException - Your questions. Answered.

success criteria for the project. In this instance, functional will be defined as a complete system that performs all of the functions below:

- User can ask questions
- Users can view asked questions
- Users can search for questions
- Users can filter questions by the three categories "Hot", "Top" and "New"
- Users can tag questions with the language they are about
- Users can filter questions by their language tag
- Users can search for questions by their language tag
- Users can comment on questions
- Users can edit comments
- Users can vote on questions
- Users can report inappropriate questions
- Users can save their favourite questions
- Users can answer questions
- Users can edit answers
- Users can vote on answers
- Users can accept answers
- Users can report answers
- Users can comment on answers
- Users can report comments

All these functions above can and will be in the form of unit tests. Unit testing my solution will allow me to test every single one of these functions. Unit testing is when you create a program to you're the functions in your program and compare the output to a manually specified and worked out result. If the expected output matches the anticipated output, the unit test is passed. If not, it is failed, and the unit test framework will produce some kind of stack trace. Some unit test frameworks that I might be using are PHP Unit, Selenium, FrontPage, or a mixture of all three.

Functional and professional careers system

The careers system on my website has to work, matching users with jobs and allowing employers to advertise vacant positions. For this criteria, functional will be defined as a complete system that performs the functions below:

- Employers can create adverts for jobs

NullPointerException - Your questions. Answered.

- Employers can remove adverts for jobs
- Adverts will be taken down once the position is filled
- Employers can view analytics data for their advert
- Employers can tag jobs
- Employers can specify job location
- Users can report jobs
- Users can apply for jobs
- Users can have their job displayed on their profile
- Users can contact employers
- Users can search for jobs based on a search query
- Users can search for jobs based on tags
- Users can search for jobs based on location
- Users can view jobs on an interactive map

As with the above section, most of these can be quantified using unit test, using a variety of frameworks as mentioned above.

Aside from simply being functional, the careers system on my website needs to both look professional, and adhere to industry standards. This is vital to the success of this section, as no employer is going to advertise jobs on an unprofessional website. In this instance, the professionalism of my solution will be quantified by asking a group of students to score my website. If the average score is over 70% then I will consider this criteria a success.

Functional tutorial zone

There isn't much (if any) backend on the tutorial zone, so there is a lot less success criteria for it. The UI/UX needs to be polished and look good, and the actual tutorial content needs to be thorough and most importantly, correct. The individual success criteria are below:

- User friendly UI
- Good UX design
- Detailed and easy to use tutorials
- No dead links
- No spelling mistakes

The success criteria for the tutorial zone is mostly quantifiable by asking a group of students to test the website and score it on the above categories. An average of at least 70% will count as a success. In addition, the tutorial programs must be able to be ran on a computer without error and must produce the expected output.

NullPointerException - Your questions. Answered.

Fast and reliable database and backend

The database on my website must be fast, in order to provide a good user experience, as no one wants long delays while searching for a question or voting on an answer. This isn't quantifiable, however I will ask a group of students to tell me how fast the website feels. However, I am limited to a free web hosting site, so slow speeds are expected. This is out of my control. In terms of reliability, the database needs to handle any unexpected errors gracefully, and shouldn't break while a user is on the website. Again, this isn't very quantifiable, but I can ask a group of students to try and break the site, to see how it reacts to unexpected use.

Secure backend

The backend of my project should never be able to be breached under any circumstances. A lot of the security is out of my control, due again to using a free online web hosting service, but I must be able to at least protect my website from vulnerability's such as SQL injection and XSS, as well as security fundamentals such as password hashing, captcha and IP address blacklisting. Again, the security of my solution is hard to quantify, and asking people to test it may do more harm than good. I'll consider this success criteria achieved if none of my friends manage to gain access to the backend or expose security flaws before I've handed the project in!