code for all: a collaborative learning site for developers

Gregory Bramble
Georgia Institute of Technology
Atlanta, GA
gbramble3@gatech.edu

ABSTRACT
In the somewhat crowded space of online collaborative learning for software developers, what is an effective tool for sharing code in such a way that increases developer productivity and maintains a high level of quality in the community? My solution is *code for all*, a website where developers directly share code snippets which they license themselves.

Author Keywords
collaborative learning; developers; sharing; attribution; licensing; code sharing; collaborative development; productivity; quality.

ACM Classification Keywords
H.5.0. Information interfaces and presentation (e.g., HCI): General.

INTRODUCTION
My work up until this point, along with peer feedback I have received, has shown me that indeed the space of online collaborative learning is a popular field with many competitors. Part of the challenge in this field is to compare and contrast different techniques for online collaborative learning, because there are so many.

Al-Jarrah, et. al. provide a good summary of the basic concepts in collaborative learning, which is as follows: "Collaborative Virtual Environments are multi-user computer-based systems which support geographically dispersed users; Collaborative Virtual Environments provide the user the ability to collaborate and communicate in a number of different ways; Collaborative Virtual Environments are virtual environments—i.e., they are based on a virtual space or world, where all activities are performed; Each user is explicitly represented within the virtual environment and visible to other users by means of embodiment; Each user is independent from the others and has the ability to make any movement within the virtual environment independently." [2] From these basic concepts, in this paper, we will go a little deeper into them as well as use them to compare and contrast different collaborative learning software systems, and ultimately explore my tool as well.

WHY CODE FOR ALL?
The first advantage in my tool is also relevant as a basic concept in collaborative online learning systems, and is a deepening of the concept given by Al-Jarrah; "Each user is explicitly represented within the virtual environment and visible to other users by means of embodiment". Similar to the other tools, my tool should have an engaged community, which is rewarded for participation in a public way, which grades the given user's reputation. Dobakhshari, et. Al showed in a slightly different context how using reputation to enhance performance can be very effective, writing, "We show that by using past behavior in determining present payments, the operator can both incentivize higher effort as well as more frequent truthtelling by the sensor and decrease the required verification frequency." [1] The first basic concept of online collaborative learning, therefore, is that the community should be engaged. Stack Overflow, for its part, succeeds in this by effectively incentivizing its community to grow by giving users who are prolific in posting good solutions recognition for various things they achieve in the site, called their reputation score.

It is hard to discuss any online collaborative setting without mentioning Facebook and Twitter, just because their communities have been so effective for sharing so many different sorts of things. While they do not incentivize participation by directly giving a reputation score, like StackOverflow does, I would argue that both sites actually do have an implicit reputation score built in, because the sites are so effective in the way they allow users to share content and also give users an effective mechanism for giving feedback on the content. Quora is also a question and answer forum website, though it is not specifically geared to a developer community, it does delve into all topics including technology. While not a direct competitor to a code-sharing site, it does have a Top Writers program to motivate users to be reliable members of the community. Therefore, Quora also effectively gives the community feedback to keep it engaged.

Having covered the first basic principle of online collaborative learning systems, which is facilitating an engaged community, we must explore the next basic principle. This principle is explained by Awedh et. al. in their paper when they discuss prior research on Collaborative Learning: "Through the process of collaboration in a collaborative learning setting, students are able to efficiently obtain huge amounts of information, which is useful to students in generating new ideas for effective learning." [3] The principle is, that, a successful collaborative learning system should provide a huge amount of information to the learners. It is not hard to see that a forum is a great way to provide this information, and all of the tools we have explored, as well as the one I am

proposing, are in this style where users publicly post a lot of information, which is indexed and becomes searchable by other users.

The final basic principle of successful collaborative learning systems we will explore here is, that the content itself must be engaging. As Taraman et. al. concluded in their very informative paper: "One of the main findings of the research is that learning outcome and contribution is best predicted by the extent to which content is engaging (the variable is different from zero with p-value < 0.001 for both the model and the coefficient of the variable). Also, the learning outcome in terms of Shapely values and content engagement is significantly different from zero (p-value for shapely values < 0.01)." [4] So, a successful collaborative learning system must not only engage its community, but its content itself must be engaging. While certainly all of the tools we have analyzed here have some content which is engaging and some which is not, one of the major advantages I am proposing in my tool is that my users will just directly provide properly licensed code snippets, so that developers can just directly use it. I believe that this is a differentiator, which will make my tool have more engaging content than the content on a site like Stack Overflow.

THE SOLUTION: KEY ASPECT OF THE DESIGN

Code for all will allow developers to easily license the code they share, and inform consumers of that license. This is the key idea of this project, that users can post code as answers to questions, and automatically attach with that code snippet a license for the code, so that users can feel free to just directly use it (of course following the terms of the license). This straightforward and clear approach to licensing is further endorsed by Ons Mlouki, Khomh, and Antoniol in their paper *Stack Overflow: A Code Laundering Platform?*, in which they show that often times developers take code from Stack Overflow without proper attribution. They claim: "These findings suggest that developers do not pay enough attention to copyright terms when reusing code from Stack Overflow or sharing code on Stack Overflow." [5]

EXISTING SOLUTIONS: RECAP

Stack Overflow: The major player in this space is Stack Overflow, which is a forum for developers to ask and answer questions, and more broadly Stack Exchange, which expands the same technology to other domains. Stack Overflow effectively incentivizes its community to grow by giving users who are prolific in posting good solutions recognition for various things they achieve in the site, like their reputation score. At the end of the day, Stack Overflow is really just a simple question and answer system, however one with a really nice and user-friendly website.

GitHubGist: GitHubGist is a more recent entry into this space, and not one that is necessarily focused on building a centralized community. Rather, the tool allows users to quickly share code and potentially publish it publicly. It is extremely simple to use, and allows users to very quickly share files and code snippets, and also gives a search engine facility to search for gists. The potential for a community is definitely there, but it is not clear if GitHub plans to have a mechanism for users to ask questions that can be answered with code, or not. Currently, that does not seem like it is part of their plan. However, GithubGists are quite close to what I am proposing, and as Gu, et. Al. show in their paper "BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain" GithubGists are becoming standard mechanisms for sharing critical software in the machine learning community. [6]

Quora: Quora is also a question and answer forum website, though it is not specifically geared to a developer community, it does delve into all topics including technology. While not a direct competitor to a code sharing site, it has many features which have made it wildly popular, including having users register with their full name rather than a screenname (to enhance credibility), including a proprietary algorithm to rank quality of answers (rather than simple upvoting/downvoting, though this is part of their algorithm), and a Top Writers program to motivate users to be reliable members of the community. Its popularity is documented by Maity et Al. when they write "Quora is one of the most popular community Q&A sites of recent times." [7]

DESIGN

For my project design, I built a collaborative web application where users contribute actual code in answer to other users' software engineering questions. My web application will facilitate this exchange and help the community by providing clear licensing for submitted code (users will choose a license when they contribute code and this license will be shown with the code), reputation points for users who contribute quality code often, and a simple, searchable interface where developers can quickly access the code they need.

THE TOOL

In this section, I will describe the tool I built and refer to the screenshots below in order to better describe the tool.

Refer to Figure 1 to better understand the main layout of *code for all*. The main page was built to highlight the three main areas in which developers may ask questions and contribute code; features, bugs, and installs. The layout is minimalist and allows users to quickly move to the section of interest.

In Figure 2, you can find one of the three pages, and in this example I show the features page. Here you can see a list of all of the questions about features that users have asked. This page is both searchable, and available to users even if they are not logged in. Moreover, the site is easily searchable and index-able, so these individual pages could easily end up in search engine results.

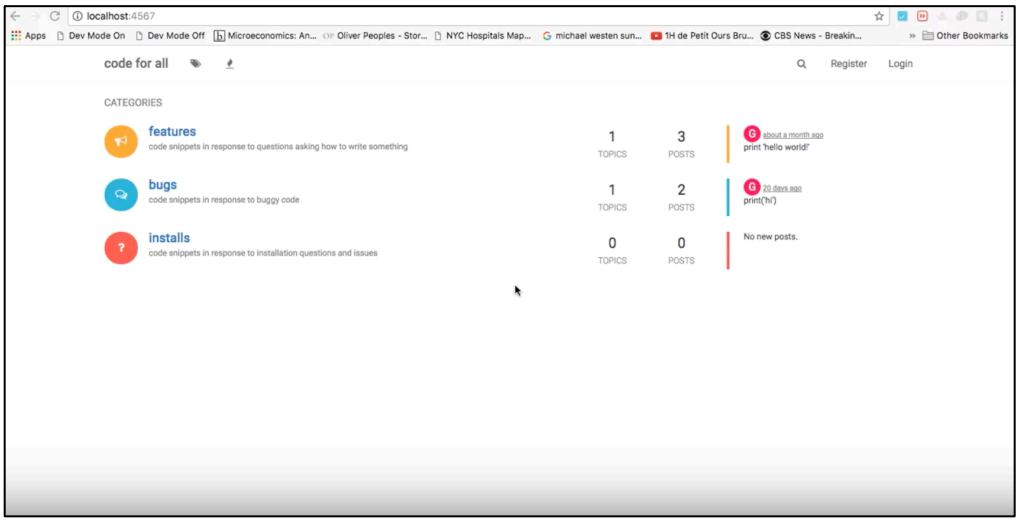


Figure 1. ***code for all*** **main page. Shows categories for submission of features, bugs, and installs.**

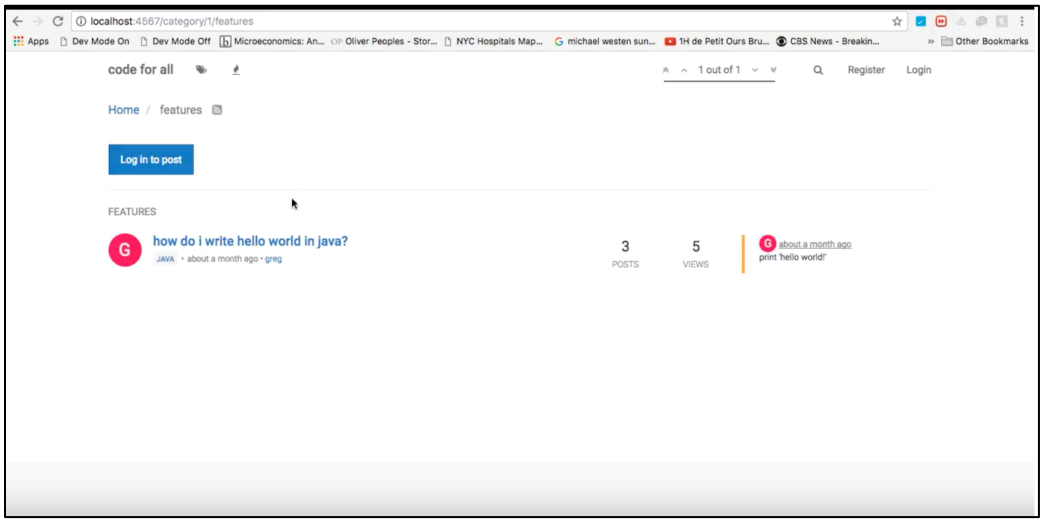


Figure 2. ***code for all*** **features page. Shows list of features about which users have asked questions.**

In Figure 3 below, you can find the most key page in the site, which is the question and code contribution page. In this page, you have an asked question, along with all possible responses, which are solely code. They are highlighted and indented for easy copying, and include the license prescribed by the contributor.

In Figure 4 below, see the login page. This simple page is important because users will sign up and be eligible for reputation scores based on their contributions, which research has shown is key to creating a successful online collaborative community.

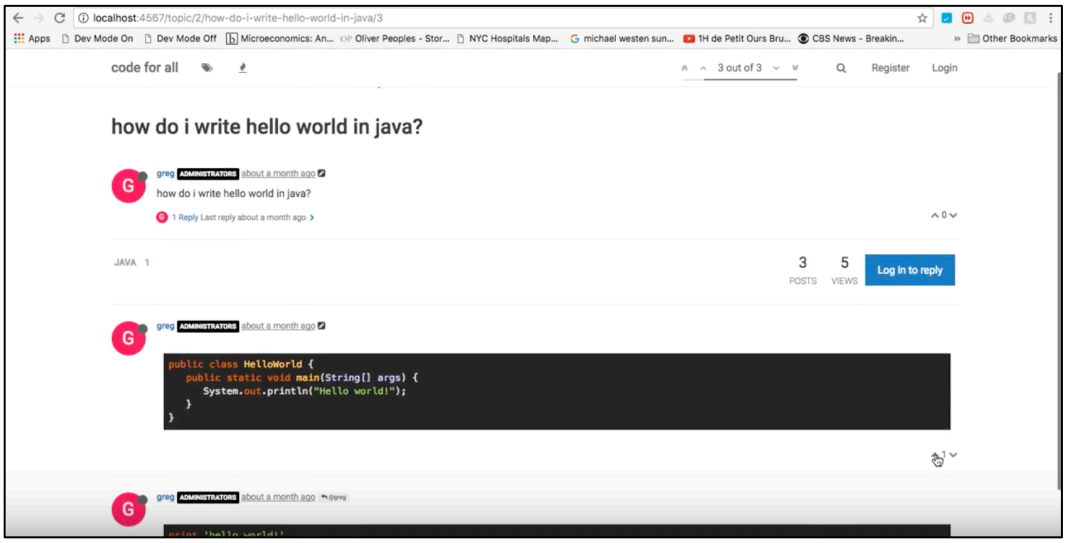


Figure 3. Sample feature question, along with submitted code snippets.

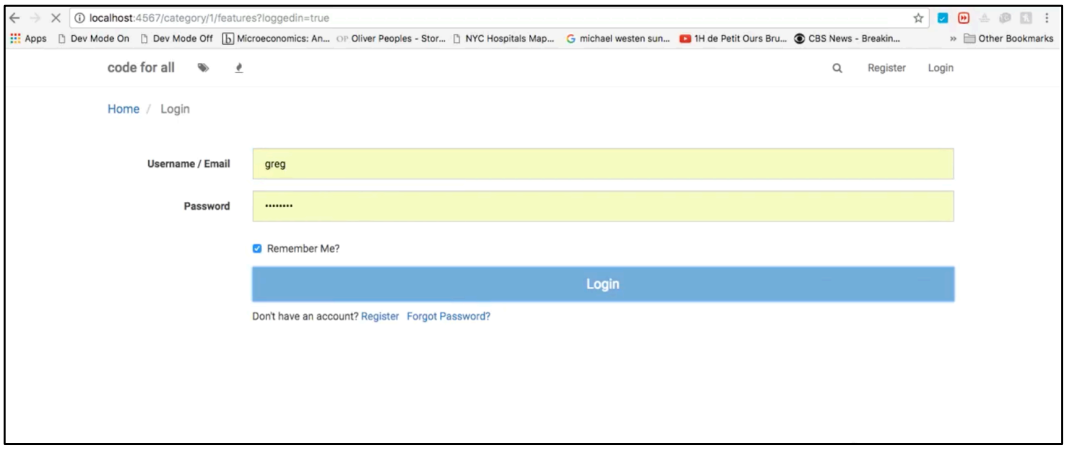


Figure 4. *code for all* login page.

In Figure 5 below, you can find a sample bug question, along with submitted code snippets. Interestingly, although the underlying reason for the correct answer in this question is that the user is accidentally using Python 2 syntax instead of Python 3, the page does not explain this. We found and based on feedback confirmed that indeed this is the most useful presentation for developers.

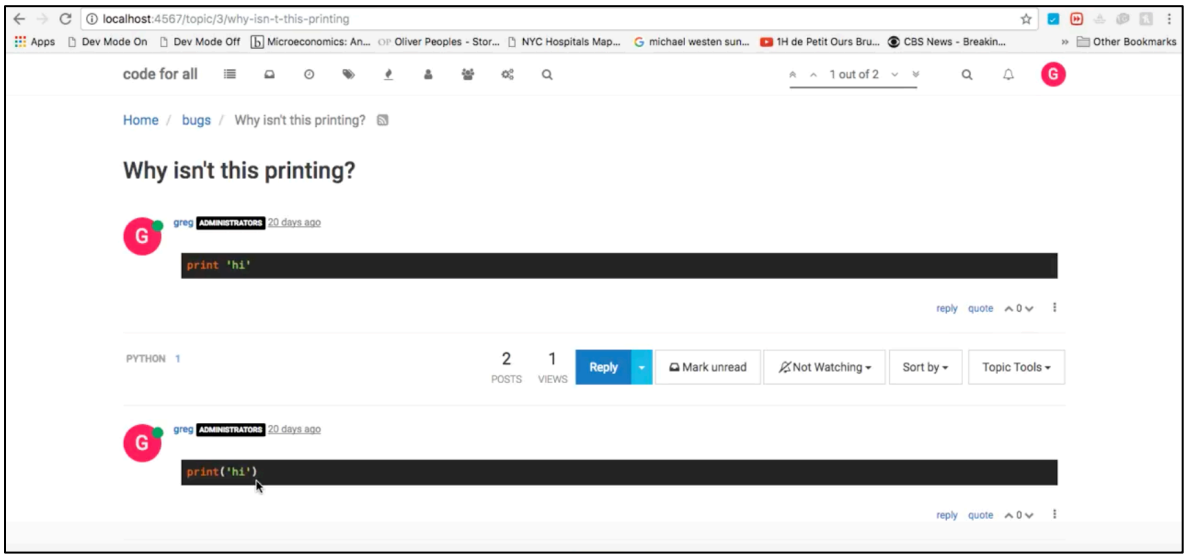
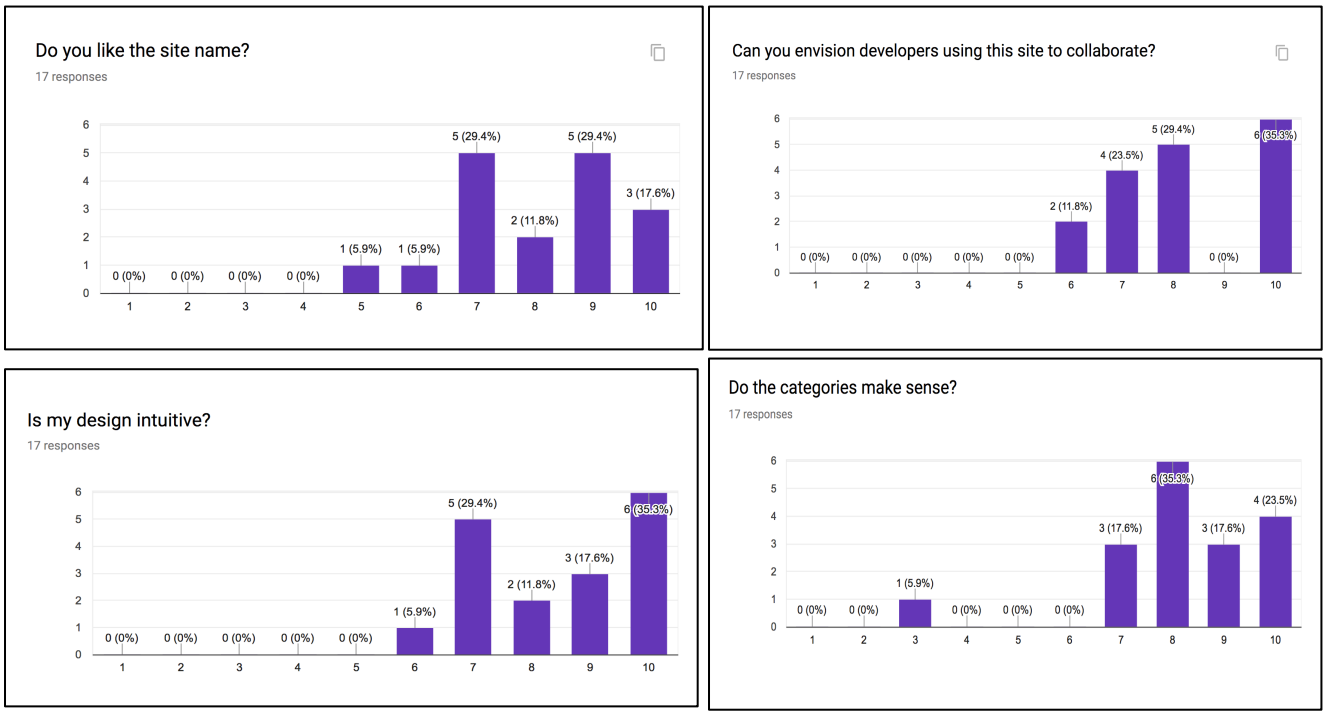


Figure 5. Sample bug question, along with submitted code snippets.

MARKET EXPLORATION SURVEY

Finally, we show the results of our market exploration survey, which shows how developers reacted to the tool. Overall, users rated the tool better than Stack Overflow, and generally the feedback was extremely positive. I will leave out the full comments for brevity, but please review the Figure 6, below, to understand how users believe a tool like this could have a major impact on the developer community.



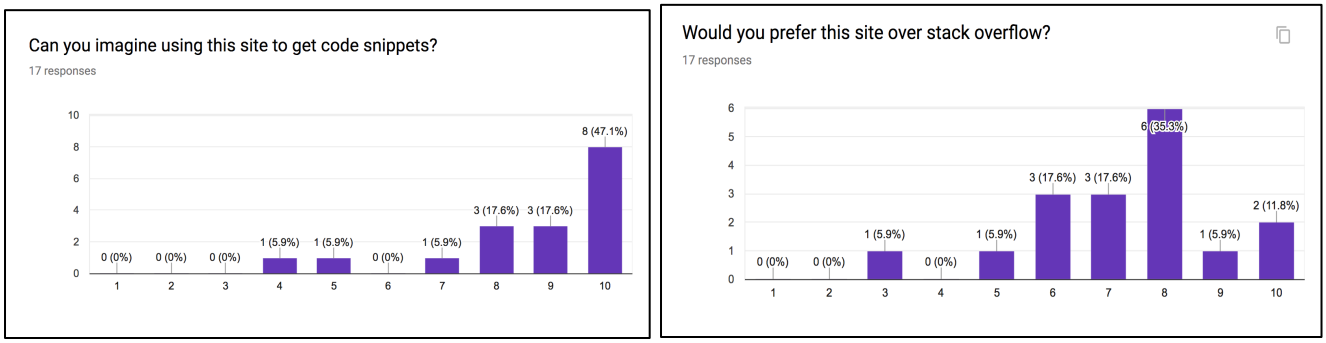


Figure 6. ***code for all*** **market exploration survey**

CONCLUSION

I will conclude with some quotes from my peers when asked "Why or why would you not prefer this site over Stack Overflow?" "I think this design focuses on code snippets rather than text explanations. This helps people looking for real answers to their problems." "Stack Overflow is popular because it has an active user base. Your largest challenge will be in getting an active vibrant user base." "Because your design is intuitive."

ACKNOWLEDGMENTS

Thank you so much to my mentor, Brandon King, to David Joyner, for creating the class, and for all of my wonderful classmates for their excellent feedback.

REFERENCES

1. Dobakhshari, Donya G., et al. "A reputation-Based contract for repeated crowdsensing with costly verification." *2017 American Control Conference (ACC)*, 2017, doi:10.23919/acc.2017.7963769.

2. Al-Jarrah, Ahmad, et. al. (2017, March 15). "The Collaborative Virtual Affinity Group Model: Principles and Design" Retrieved October 1, 2017, from https://arxiv.org/pdf/1703.04917.pdf

3. Awedh, Mohammad, et. al. (2014, December 4). "Using Socrative and Smartphones for the support of collaborative learning" Retrieved October 1, 2017, from https://arxiv.org/pdf/1501.01276.pdf

4. Taraman, Sara, et. al. "Employing Game theory and Multilevel Analysis to Predict the Factors that Affect Collaborative Learning Outcomes: An Empirical Study" Retrieved October 1, 2017, from https://arxiv.org/pdf/1610.05075.pdf

5. Ons Mlouki, L., Khomh, F., & Antoniol, G. (2017, March 11). Stack Overflow: A Code Laundering Platform? Retrieved September 17, 2017, from https://arxiv.org/pdf/1703.03897.pdf

6. Gu, Tianyu, et al. "BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain." 2017, https://arxiv.org/pdf/1708.06733.pdf

7. Maity, Suman. Kharb, A., & Mukherjee, A. (2017, March 11). "Language Use Matters: Analysis of the Linguistic Structure of Question Texts Can Characterize Answerability in Quora" Retrieved September 24, 2017, from https://arxiv.org/pdf/1703.04001.pdf