



IIIT ‘Stack Overflow’

Scripting and Computer Environments (CSE505) Project

Overview

IIIT has many smart people around and some people that want answers to their questions. But we don't have a Q/A system (except Moodle) across disciplines and courses that can help the students get their questions answers. So let's make a StackOverflow clone tailor-made for IIIT.

Technologies Allowed :

HTML, CSS (Bootstrap/Any framework) and JavaScript/jQuery for the Front-End (Client Side). Flask (Python) for the Server Side and SQLite as the Database of choice. :)

A Piece of Advice :

This spec is not complete and adding your own creative elements and features and UI/design is always welcomed and appreciated and rewarded.

Although we are talking about making a StackOverflow clone, it's encouraged and appreciated that you create one with your own flavor. Go creative and try not to follow the design UI of StackOverflow and create your own nice looking front-end and UI. Creativity will always get better marks. That being said, please **DO NOT** copy the CSS from StackOverflow.

Note on Plagiarism :

This project is detailed and requires a good amount of effort. Thus plagiarism like copying code from others won't be tolerated and will fetch you a **ZERO** in this otherwise heavily weighted project. :)

Expected Features

I. Basic User Registration and Logging In/Out

Users should be able to register using their students or research mail IDs. (Need to check if any other type of IDs) are used. With registration, they also get a “unique” username. Users can login using the username using which they registered. There should be an option to remember the login so that whenever the site is opened again, the user is already logged in. (If the user logs out in a session then they need to login again).

Users should be able to upload a profile picture of their own (which will be used whenever their name is displayed with a question/answer)

II. Homepage and Search

Registered and Non Registered Users can get a homepage containing a list questions per page (example 10 per page). The list should contain the question title (described in the next section) and the number of answers to that question. Clicking on the question title will open the page containing the question and all answers.

(Bonus: Display the tags related to the questions too and clicking on a tag displays all the questions related to tag)

Only registered users must be able to answer a question/ask a question. But non registered users can still search and view questions.

There should be a search bar on top of the home page. Users can search for questions in that search bar (No need to implement advanced search, basic substring matching is fine).

III. Asking Questions and Getting Answers

The main idea behind stack overflow is to ask questions and write answers to those questions (no more distractions).

I. Questions

Any “registered user” can ask a question. The question must contain a title (having the gist of the question) and a details of that question.

Users must be able to add “tags” to their questions (0 to 5 tags). Those tags can be clicked to get all questions of that type.

The question must also contain the username of the person, along with their photo who asked it and the date-time of the question asked.

II. Answers


Any “registered user” can answer a question (users can write answers to their own questions too). There can be multiple answers for a question. Each answer must have the username and profile photo of the user who posted it.


Display the total number of answers for a question before the answers.


There are “Upvote” and “Downvote” buttons with answer where other users can upvote and downvote any given answer. The answer that has the most upvotes must be displayed on top. (Basically sort the answers using upvotes).

The person who asked the question gets an “Accept Answer” button using which they can mark an answer as “Accepted”. They can only mark one such answer.


(Bonus: Try to figure out a way to display “Accepted” answer on top even if its not the most upvoted answer)

 Good answers are voted up and **rise to the top**.
The best answers show up first so that they are always easy to find.

 The person who asked can mark one answer as “accepted”.
Accepting doesn't mean it's the best answer, it just means that it worked for the person who asked.


 3,857 ● 5 ● 31 ● 66

2 Answers


4


Swift code can be deployed to OS X 10.9 and iOS 7.0. It will usually c
versions.

answered Jun 3 '14 at 8:25




Greg Parker

7,316 ● 1 ● 16 ● 19


3

Apple has announced that Swift apps will be backward compatible wi
The WWDC app is written in Swift.

answered Jun 3 '14 at 0:03




Ben Gottlieb

79k ● 22 ● 168 ● 169

(Example Screenshot from StackOverflow. Need not reproduce exactly)

Below each answer, display the username of the person whose answer it is.

IV. Comments on Answers and Questions



Any users (including the one who asked or answered) can add multiple(0 or more) comments to answers and questions using the button/link provided.

The comment should also display the name of the person who commented it.

V. Reputation System

Users earn reputation whenever their answers are upvoted. (Like +10 for each upvote). Figure out a way to display the reputation of the user (in their login handle or where their name is displayed in answers/question)

Error Handling

Suffice to say, your code must stay running if it encounters normal runtime errors. Try to perform proper validation wherever needed.

Still Want More?

If you think you have better ideas or some creative designs for your project, by all means go creative and add your own features and flavour to this application. You'll surely be rewarded for the same.

Happy Scripting! Cheers!