**Name of the Problem Statement** : User Query on Stack Overflow

**Team Size and members** : 3 Members

* Aditya Kumar
* Hitesh Varma P.
* Rohan Banerjee

**Problem Statement**

The problem statement aims at building a solution that helps to find the right answers that are relevant to the developer issues on Stack Overflow.

A lot of content is present in form of stack overflow questions and answers, various studies point that developers face problems while development life cycles and they ask questions on stack overflow which gets answered by fellow developers across the globe. For a new developer to understand a concept or solve an issue, it could be very difficult to identify the problems. The proposed solution should help to identify most relevant questions to a query using text similarity including identify the matching tags and pick top relevant questions from stack overflow and identify top (k) solutions of the problem based on sentiment analysis of reviews of the given solutions on the Stack Overflow.

With our solution, we provide a simple interface which can be used to get the top k answers quickly. Along with that, we show how good the answers found actually are, with a confidence measure that is indicated by color and can be interpreted in a blink of the eye.

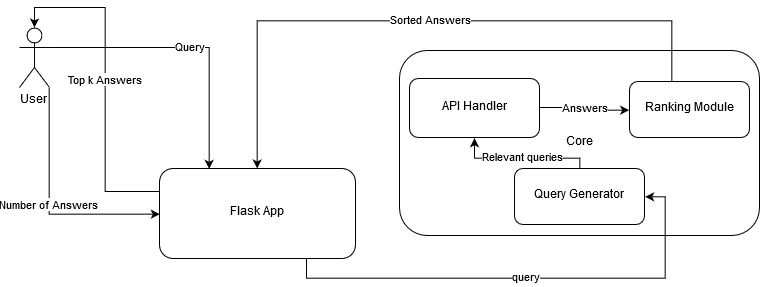
**Business Impact**

Our solution will be useful for boosting productivity of developers in businesses as they don’t have to search around for queries that have already been answered on Stack Overflow. Plus, the quality of the answers is interpreted for them and they don’t have to spend time sorting out the best answers. Boosted developer productivity means faster development and deployment of new technologies.

**Scope of Work**

* Data Fetching
  + The data we are using is fetched from “**stackexchange**”, contains a lot of un-processed information. The entire data used is solely from the API used.
* Text Processing
  + Since the first component in the solution is to analyze the text given by the user, we have filtered the text by removing stopping words and punctuations and acquired mainly nouns or verbs to search for results.
  + And we also allow the user to adjust the value of **k** while searching for a query.
* Result Ranking
  + The top **k** results from API search are ranked using a custom-made formula.
  + These top **k** results will then be given a number (confidence score) each which will be used to rank the results.
* Data Visualization
  + The ranked queries are to displayed in each division box. And there must be some difference between the best result box and the worst one.
  + The user if wishes, should be able to hide the result which is in display and also unhide.

**Technologies and Architectural Flow:**

****

This project was made using Python 3. Technologies used alongside are:

* NLTK library for Natural Language Processing.
* Flask, along with HTML, CSS and JS for building the web application.
* Py-Stackexchange library for dealing with the Stack Exchange API.
* Html2text library for displaying answers on the command line.
* Requests library for sending API requests.

**Role of Team Members**

1. **Aditya Kumar - Core modules and CLI script**
2. **Hitesh Varma P. - Flask Backend**
3. **Rohan Banerjee - Frontend**

Thank you