The main idea behind my approach is to get correct tags for a particular question, i.e to built an auto-tag feature for every question asked.

On applying this auto-tag feature we would be benefitted in 2 ways:

1. If correct tags are identified then the search space can be reduced to a great extent and results obtained would be faster and accurate. The auto-tagging feature of our project would help the user in identifying his/her interest more properly as the aim of the problem statement suggests.
2. The stack-overflow could identify the user's correct interest and can suggest similar questions related to the user's searched questions.

The current approach in the auto tag:-

Whenever a search is made in StackOverflow then the site tries to find the questions similar which has the same tags as the question asked. Like in the example-” how to run node app.js?”, fetches 426 results with tags which are useful and which are not useful like “discord”, ”bots” etc.

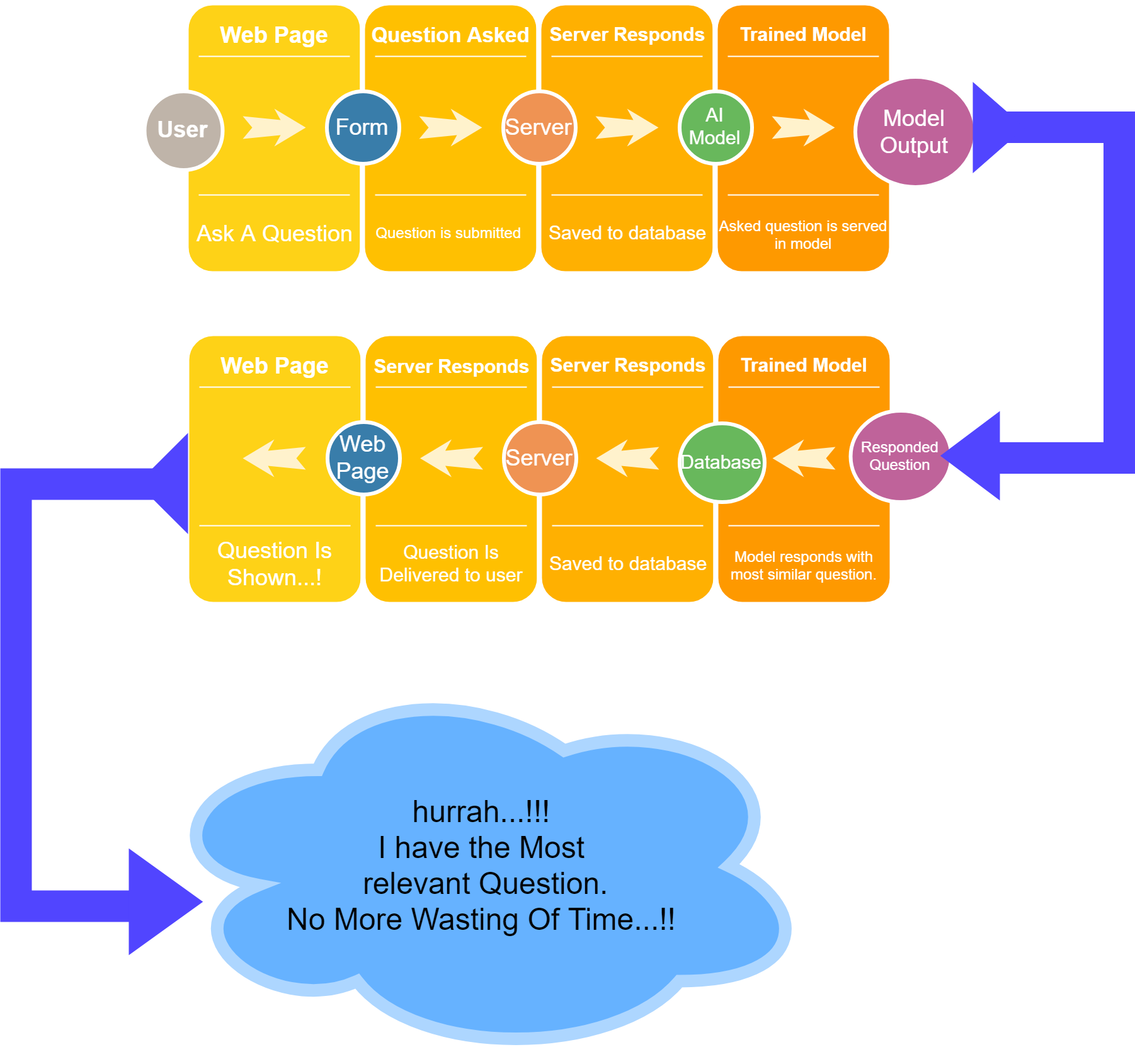
My main focus while solving this problem statement was on finding the tags correctly which can define the user search space completely. Hence by using deep learning, I have been successful to a great extent, which would define the user search space correctly.

**The above concept and it’s execution have been displayed by three flowcharts:**

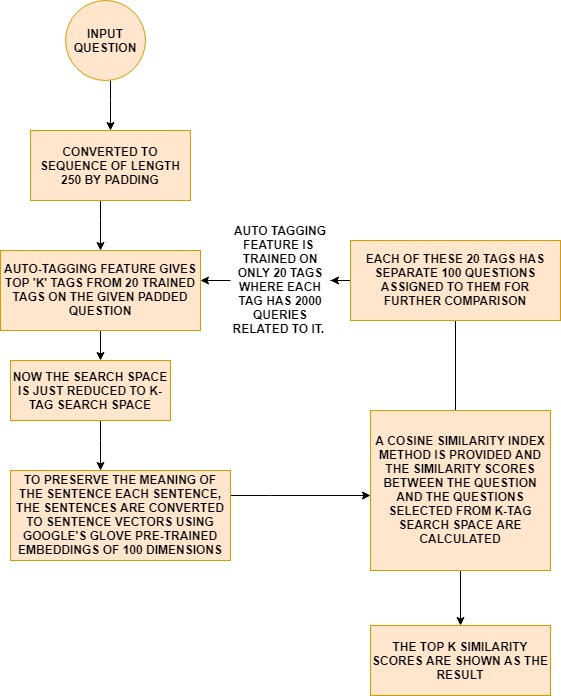
**Fig 1:- This flowchart displays how the User Interface has been defined**

**Fig 2:- This flowchart displays the basic overview of how the mechanism works after the question has been input.**

**Fig 3:- This flowchart displays the auto-tag feature which is the heart of the out developed solution.**

**FIG:1**

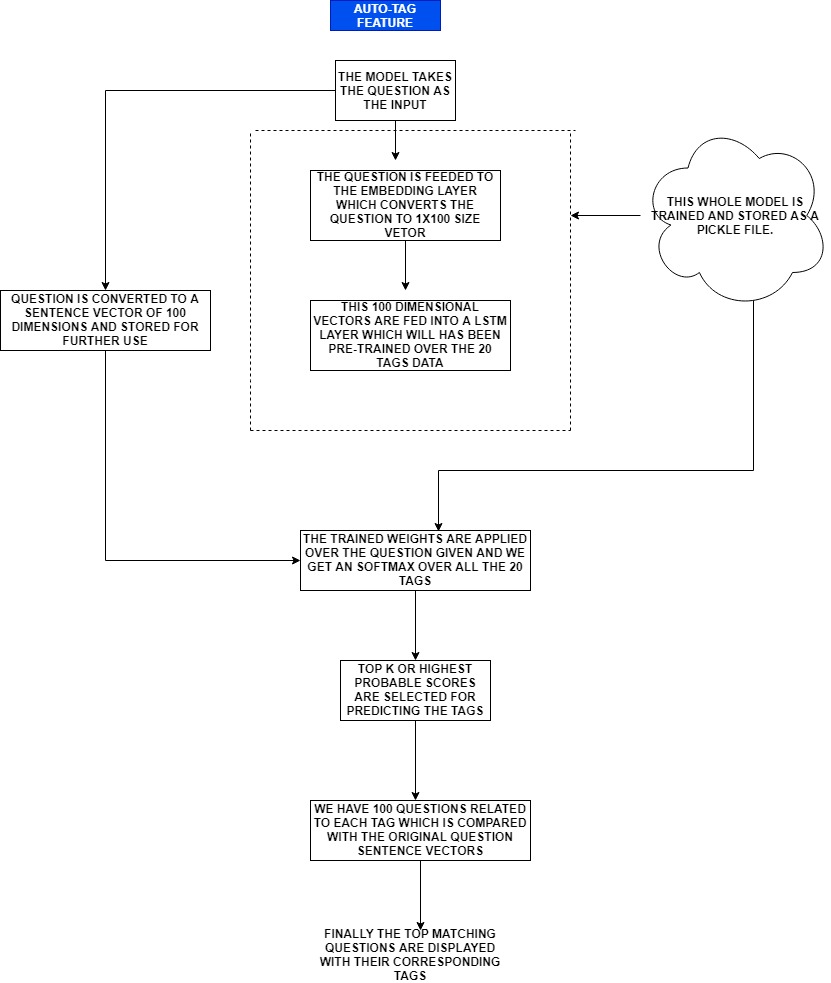
**USER INTERFACE**



The above flow-chart shows the basic approach we have applied to solve the problem, except the auto-tag part which is the heart of our solution has been displayed below.

**FIG2:**

**BASIC APPROACH TO THE SOLUTION**



**FIG3:**

**AUTO-TAG FEATURE**(USING LSTM)

As we all know machine modeling starts from data even saying more precisely “RELEVANT DATA”.

So for that we have scrapped most answered and most upvoted questions on the given topic from stackoverflow.

Using front end we greet the user with our landing page and give an option to ask the question on any topic on which we have trained our model.

Based on the users question we process the data via our backend system and store in a database for future reference.

The question asked is processed and returns the most similar question from stackoverflow database questions on that given topic.

The processed question is then shown back to the user so that he/she can ask the most relevant question in spite of typing random keywords.