1. Team Name:

   CaptainMarvel

2. Problem Statement:

   Identification of User Query on Stack Overflow Using Semantic Search

3. Team Member

   Sunita Sen

   Role:

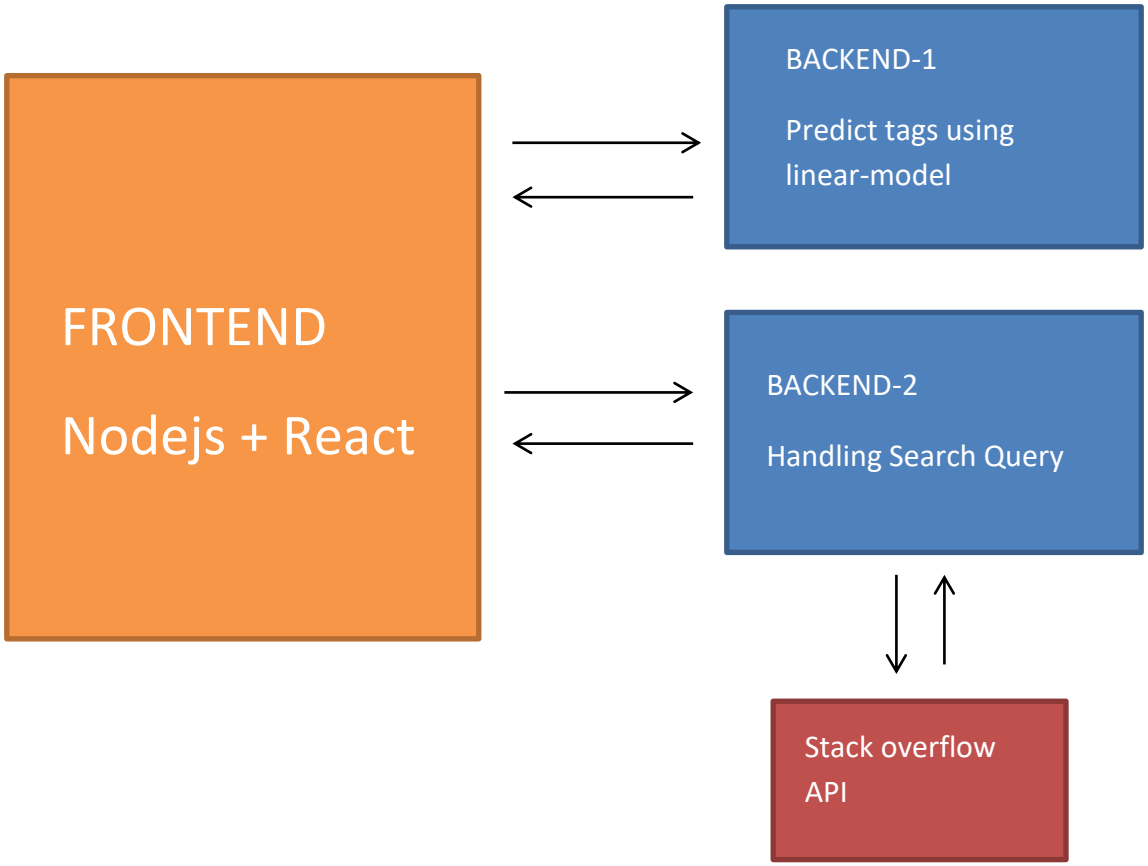
   i. Development of the Node.js application
   ii. Development of the backend of the web app.
   iii. Deployment of the app in cloud.

4. Uniqueness and Impact on business:

   Stack Overflow is a go-to website for almost all developer. In the given problem statement we had to identify top solutions for the search query. The way stack overflow does this, they produce solution of all queries at one go. However, this sometimes becomes irksome for users to sort out questions and solutions. In my solution, the top relevant questions and answers are provided in different tabs which will help the users to find the relevant question and the appropriate solution he/she wishes to view. Moreover, sometimes the solution also lies in the linked questions. Hence, top linked questions are also shown in a separate column.

   Also, based on the user query relevant tags is identified using linear-model. This allows the user to further read about the topic he/she has queries about. This is especially helpful for beginners.

5. Architechture:



6. Technology used:
    a. Frontend:
        i. Node.js – Express
        ii. React
    b. Backend
        i. Flask(Python)
        ii. Sklearn classifier model
    c. External API
        i. Stackexchange API- free version
    d. Hosting
        i. Heroku

7. Work done:
    a. Frontend:

The frontend of the application is made using react. It is used to display the relevant questions and answer for the query given.

b. Tag Predictor:

Here we are using python's NLTK library and linear classifier to identify relevant tags in the user query. This return the tags predicted(if any) for the query. It accepts flask request.

c. Search Handling:

It is made using flask. It is responsible for processing the query given and request the stack overflow api. Here, we are using the free tier of stackexchange api(300 queries in a day).

8. Further Work to be done:
    a. Since we are using a third party API here, the response is relatively slow. Work can be done to improve this.
    b. The linear classifier gets trained for a very small data set. As a result it cannot predict tags for many queries. If it could have been trained for a larger dataset it would have been much more accurate.