I.J. Information Technology and Computer Science, **2023, 4, 33-49**
Published Online on August 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2023.04.04

CodeUP: A Web Application for Collaborative Question-answering System

Yashi Agarwal
Jaypee Institute of Information Technology, A-10, Sector 62, NOIDA, India
E-mail: 18803033@mail.jiit.ac.in
ORCID iD: https://orcid.org/0000-0002-3777-8997

P. Raghu Vamsi*
Jaypee Institute of Information Technology, A-10, Sector 62, NOIDA, India
E-mail: prvonline@yahoo.co.in
*Corresponding author
ORCID iD: https://orcid.org/0000-0001-7726-6198

Siddhant Jain
Jaypee Institute of Information Technology, A-10, Sector 62, NOIDA, India
E-mail: 18803010@mail.jiit.ac.in
ORCID iD: https://orcid.org/0000-0002-6656-1121

Jayant Goel
Jaypee Institute of Information Technology, A-10, Sector 62, NOIDA, India
E-mail: 18103255@mail.jiit.ac.in
ORCID iD: https://orcid.org/0000-0002-5322-3414

Received: 01 September 2022; Revised: 13 January 2023; Accepted: 26 May 2023; Published: 08 August 2023

Abstract: The majority of collaborative learning and knowledge sharing (CLKS) platforms are built with numerous communication mediums, team and task management in mind. However, with the CLKS, the Question-Answering (QAs), User profile evaluation based on the quality of answers provided, and feeding of subject or project relevant data are all available. QAs are required for online or offline cooperation between team members or users. To that purpose, this paper presents a web application called CodeUP with features like QA system, Question similarity testing, and user profile rating for boosting communication and cooperation efficiency in CLKS for academic groups and small development teams. CodeUP is intended to be quickly established and step for academic or development groups to collaborate. As the CodeUP application supports the CLKS, it is also an ideal tool for academia and development teams to perform computer supported QA system and knowledge sharing in the sphere of work or study.

Index Terms: BERT, Collaborative Learning, Deep Learning Models, Electra, Knowledge sharing, Natural Language Processing, Universal Sentence Encoder, Software module, Question Answer.

1. Introduction

Collaborative learning and knowledge sharing (CLKS) has been identified as a critical life skill for the twenty-first century in academia and development [1]. It has been used to increase educational standards and assist students in better understanding and resolving their problems. The collaboration and exchange of information among teachers in technical and programming courses result in the development of high-quality education that fosters creativity and innovation. Collaborative learning involves a group of individuals working together to acquire new skills, solve problems, and even generate novel services or products. Meanwhile, a knowledge-sharing platform takes this information and presents it in a way that allows users to easily comprehend and engage with it. Apart from preserving knowledge, users of such a platform can communicate, ask questions, provide feedback, and make basic edits without needing to switch to different software. By integrating learning and knowledge processes, CLKS enhances organizational learning. Effective communication, sharing of ideas, and transmission of information all contribute to the advancement of knowledge through various means, including in-person meetings, discussions, faculty development initiatives, and collaborations between the industry and educational institutions [2, 3, 4, 5, 6, 7].

Technically, collaborative knowledge sharing is the use of tools or technology that allows teams of participants to communicate, share ideas, collaborate, record the interactions, and save the interactions for future use. Web-based tools and technologies are regarded as the most practical and cost-effective method of group learning and information exchange. These platforms are now widely available on the internet. Web 2.0 tools are classified into three categories: 1) Instant messaging and chatting apps such as Yahoo Messenger, Google Hangouts, and others. 2) Web-based and mobile video conferencing apps like Zoom, Microsoft Teams, Google Meet, and others. 3) Internet barriers such as blogs, Wikipedia, and question-and-answer websites such as Quora and Stack Overflow. Online tools for teaching, learning, and assessment are available in academia, such as Google Classroom, Piazza, ELAP, and others [8, 9]. The programming and development teams in software development use Question-Answering (QA) platforms such as Quora and Stack Overflow, as well as social networking-based web forms such as Facebook. Google Classroom, for example, includes team management, task management, and assessment features, but no domain-specific QA system. The same as with the Piazza and the Facebook web forum [10].

Student team's collaboration task can be made easy by choosing efficient knowledge sharing platform (KSP). There are five essential features to look for in any KSP, they are 1) Search engine or powerful search facility – user should be able to find correct information via search facility, 2) Collaboration and communication – user of the system able to ask questions, provide answers and feedback to boost the productivity of the development team, 3) Integration – easy integration of the KSP to the development environment to find the information, 4) Reporting and Analytics – availability of reporting and analytics provide the team to easily analyze the weak areas and provides the scope for initiating steps for improving the training and communication process, 5) Inter platform operations – user should be able to access the information irrespective of the operating platform such as web or mobile. Some examples of KSPs are Nuclino, Notion, Confluence, Microsoft Share point, Google Workplace, and Papyrs. Out of these the first three platforms are free to use with limited features and users option. The rest three platforms consist of powerful features but they are proprietary which requires license purchase [2, 11, 12].

1.1. Contributions

To this end, this paper presents a web application, called CodeUP, with the features such as QA system, user profile rating and question similarity. CodeUp is a web application for collaborative knowledge sharing platform, where users can build up their community, by searching for other users within the same domain. Users also get a detailed view about the other users profile like the technical skills, experiences in programming and the profile rating which will give an idea about knowledge of that user. CodeUP is a web-based collaboration tool created specifically for users who are looking for high-quality programming-related content and who want to interact with other developers and who share their interests. The primary driving force behind this work is the fact that it is challenging for students and developers to expand their networks on conventional social networking sites because there is more distraction than knowledge-based content on these platforms [13]. Additionally, it is a platform where developers can discuss their issues, assist others in resolving their issues, or simply see what topics are being discussed in order to possibly discover a new module they are unfamiliar with but find intriguing. To save developers time in researching new technical advancements and connecting with the proper individual to help them or collaborate with them on something that has the potential to address the programming issue. This web application is designed to setup easily and to make it as an ideal tool for academia and development teams to perform computer supported learning and knowledge sharing in the sphere of work or study.

1.2. Existing Approaches vs. CodeUP

StackOverflow is quite slow in identifying question similarity; it takes a few days to detect the similarity between questions, but CodeUP offers the similarity in real time with the highest accuracy. CodeUP can get embeddings in real time using BERT, Electra, and USE (Universal Sentence Encoder), and then use cosine similarity to get the top 'N' comparable questions with similarity scores. Profile rating is vital in determining the user's degree of skill. Existing Q&A sites, such as Stackoverflow, assign a profile grade based on conditions such as a user's consistency, for example, how many days the user accesses the site constantly, and likes on answers. Rating on that basis is not a characteristic that is relevant. Because rating a person based on the number of days he visits the site does not guarantee that the response provided by that user is relevant. In addition, Stackoverflow determines user rating based on user details if the user has finished his profile. However, we must evaluate elements such as the number of responses provided by the user and the amount of likes he has earned on them. In CodeUP, we find profile ratings based on the number of replies as well as the amount of likes the user has earned on them. In this way, all of these issues are being managed in a balanced manner.

The remaining sections of the paper are structured as follows. Following an overview of relevant studies in Section 2, Section 3 introduces the design and implementation process of the CodeUP web application. It demonstrates how the application incorporates essential elements such as a QA system, user ratings, and question similarity checking to enhance collaboration within academic and development teams. Section 4 presents the software implementation and testing. Finally, Section 5 concludes the paper with future work.

2. Related Works

This section provides an overview of the relevant research conducted on the key modules of CodeUP, namely the Question Answer System, User rating, and Question similarity checking.

In one study by Tomas et al. [8], an approach was described that utilized convolution neural networks to train a wrapper capable of extracting data from previously unexplored templates. This wrapper could effectively gather data from individual web pages without requiring site-specific configuration. The authors also introduced a technique for spatial text encoding, combining visual and textual elements into a single neural network. Promising outcomes were observed in early attempts to extract product information, indicating the potential for a site-independent web wrapper.

Faisal et al. [14] proposed expert-ranking methods based on the g-index and tested them using data from the Stack Overflow forum. Their approach assessed a user's reputation and level of expertise based on the quality and consistency of their responses. Experimental results confirmed that the suggested expert-ranking algorithms, particularly Weighted Exp-PC and Exp-PC, more accurately identified genuine experts.

Diyanati et al. [15] proposed a method for assessing individuals' knowledge levels using statistical information from questions and answers. Two approaches were tested: the scoring method, which focused on the scores assigned to questions and responses, and the comment-mining method, which analyzed positive and negative comments. The scoring method did not show a significant correlation between question and response scores, making it unsuitable for determining users' knowledge levels. However, the comment-mining method effectively assessed users' competence.

These techniques were applied and evaluated using real data from the Stack Overflow website. Overall, the research demonstrated the potential of convolution neural networks for data extraction, the effectiveness of expert-ranking algorithms, and the limitations of certain scoring methods in assessing knowledge levels.

Yang et al. [16] proposed an innovative approach for learning representations that capture semantic similarity at the sentence level using conversational data. They employed an unsupervised model to predict conversational input-response pairings, resulting in well-performing sentence embeddings. These embeddings demonstrated strong performance in both the SemEval 2017 Community Question Answering (CQA) question similarity sub-task and the semantic textual similarity (STS) benchmark. The authors achieved this by combining the conversational input-response prediction task with a natural language inference task and employing multitask training. Similarly, Daneial et al. [17] developed models that specifically focused on transferring learning to different Natural Language Processing (NLP) tasks to create sentence embeddings. These models exhibited effective and accurate performance across a range of transfer tasks.

Sentence embeddings-based transfer learning has been found to outperform word-level transfer, as noted by Yang et al. [18]. The authors observed significant performance improvements in transfer learning with sentence embeddings, even when limited supervised training data is available for the transfer task. In a related study, Zhang et al. [19] proposed a rule-based and semantic automated Information Extraction (IE) method using construction regulatory papers. This method utilizes a set of IE rules based on pattern matching and conflict resolution (CR) rules.

Stack Overflow, a popular Community-based Question Answer (CQA) website focusing on software programming, has experienced a surge in popularity. However, the occurrence of duplicate queries on Stack Overflow poses a challenge, requiring manual identification by high-reputation users. Wang et al. [20] addressed this issue by investigating the application of powerful deep learning techniques, including Convolution Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM), along with the Word2Vec framework. To identify duplicate questions, the authors developed three deep learning models, namely WV-CNN, WV-RNN, and WV-LSTM, based on Word2Vec, CNN, RNN, and LSTM. These models were shown to outperform four traditional machine learning techniques, including Support Vector Machine, Logic Regression, Random Forest, and eXtreme Gradient Boosting, in terms of recall-rate Boosting [21, 22, 23, 24, 25]. The utilization of Word2Vec allowed for comprehensive semantic representation at both the word and document levels.

3. The Design and Development of CodeUP

3.1. Methodology of Research

To develop the CodeUP application, the Software Development Life Cycle (SDLC) methodology is employed, consisting of four main steps: 1) Requirements analysis, 2) Software design, 3) Design implementation (coding), and 4) Testing. In line with the objectives and contributions outlined in section 1, the requirements for CodeUP development are thoroughly examined. These requirements encompass functional aspects, the Q&A system, the database, and the web application itself. The analysis of these requirements is then incorporated into the design phase to gain a deeper understanding of the proposed application's functionality. The design phase involves the implementation of the Q&A system, the development of the web application using ReactJS, and the design of the database tables using MongoDB. Finally, the SDLC concludes with the testing phase, during which relevant test cases are prepared and executed to ensure the functionality and reliability of the completed CodeUP application.

3.2. *Brief Description of the Solution Approach*

CodeUP is a web-based collaboration tool created specifically for users who are looking for high-quality programming-related content and who want to interact with other developers who share their interests. It consists of three main modules such as 1) the Question-Answer Module, 2) Finding Similar Question Module, and 3) Profile Rating Module. The Question-Answer Module has features such as asking questions, editing questions, deleting questions, listing all questions, searching questions based on category, adding answers to the question, such as the answer that will help in the profile rating module, adding comments to that answer, such as whether it was helpful or what other ways of implementing that solution, editing answer, and deleting answer. The Locating Similar Questions module assists users in finding questions that are similar to what they want to ask because there is a good likelihood that someone else has had a similar difficulty and someone has helped him with a solution.

3.3. *Process Flow of CodeUP*

Fig. 1 and Fig. 2 respectively show the activity diagram and control flow diagram of the CodeUP. Fig. 2 depicts in detail various modules and sub-modules and flow of CodeUP. Like the other social networking sites, a user needs to sign up for accessing the system. An authentic user can work on the modules mentioned above. After successful login, the home page of the CodeUP will be displayed to user. In this, My Questions shows the list of questions posted for answers by the user. Question Feed shows the list of questions posted by the users under collaboration. Profile section maintains the profile information of the user such as Experience and the number of questions answered. Search Question will help the user to search for questions as per the topic category. Over a period of time most of the questions posted by users will be answered by other users so that the Search similar question module will find similar questions and present on to the screen as and when the user starts typing the question. Any user may start posting the question anytime; the posted questions are checked with the input question entered. During the asking question, Search similar question module will be activated to check does the answer to the question is available in the database. If the question is not available then the entered question will be saved in to the database.

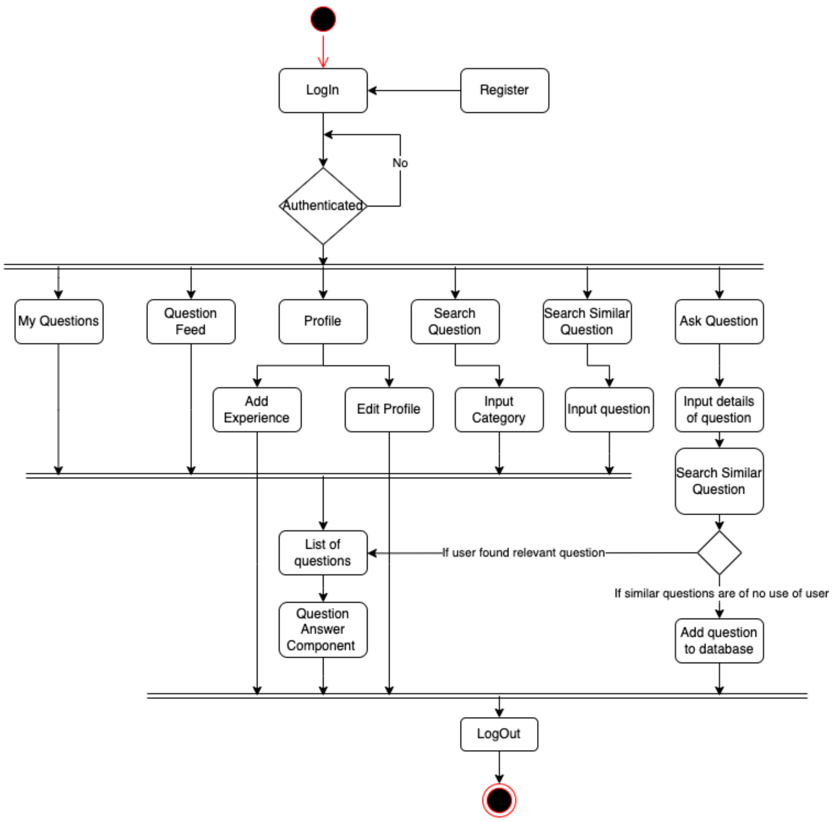


Fig.1. Activity diagram of CodeUP

3.4. *Requirement Analysis*

CodeUp is developed by considering the following functional and non-functional requirements into consideration.

A. *Functional Requirements of Each Module*

The QA module of CodeUP is designed to fulfill the following requirements:

- Enable users to add questions.
- Allow users to delete questions.
- Provide users with the ability to edit questions.
- Enable users to search for similar questions.
- Allow users to browse questions based on categories.
- Provide users with access to view all questions.
- Allow users to select a category through a dropdown menu.
- Enable users to add answers to questions.
- Allow users to edit their own answers.
- Provide the capability for users to delete their own answers.
- Enable users to like answers.
- Allow users to comment on answers.

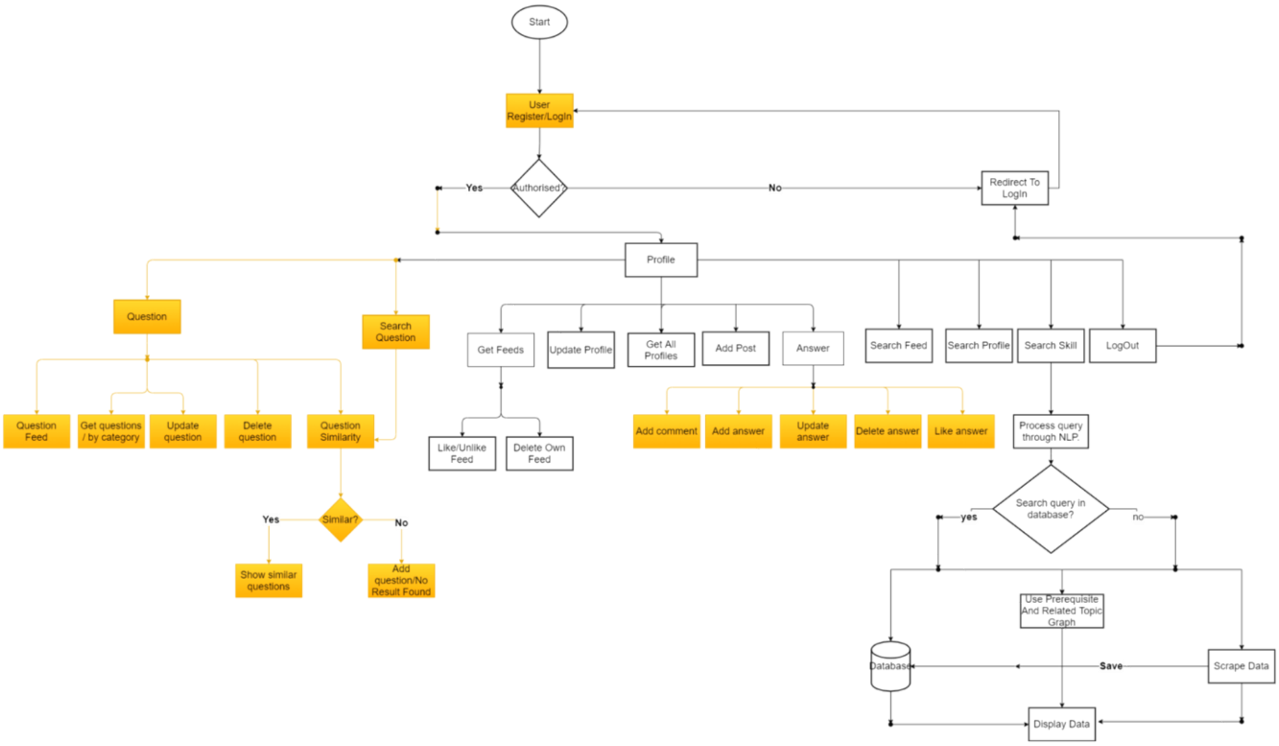


Fig.2. Control flow diagram of the CodeUP

The functional requirements have been identified and implemented within the profile module to ensure that users can effectively find and manage profiles, add relevant information, update their own profiles, include their experience, and view profile ratings.

B. Non Functional Requirements

The following non-functional requirements are addressed in the CodeUP.

- Security: It indicates whether the product stores or transmits sensitive information. Is it necessary for the IT department to conform to specified standards? To address this, we implemented authorization checks in each route.
- Capacity: It specifies the system's storage requirements, both now and in the future. How will the system handle increased volume demands? This we addressed using AWS support.
- Compatibility: It specifies the minimal hardware requirements. What operating systems and versions are required to be supported? We utilized Google colab for it. RAM for the system should be minimum 2 GB.
- Reliability and availability: It specifies the critical failure time under typical conditions. CodeUP is highly available as most of the functions are asynchronous.
- Maintainability + Manageability: This metric indicates how long it takes to repair components and how readily an administrator can administer the system. We only need management once when we start our system; currently, we are utilizing Google colab, so we have the model to manage it each time when started.
- Scalability: It indicates the maximum workloads under which the system will continue to perform as intended.
- Usability: This metric indicates how simple the product is to use. What defines the product's user experience?

And our user experience is unrivalled.

- Performance: It refers to how quickly the system responds to user actions or how long a user must wait for a certain process to occur. CodeUP loading time is 5 minutes for merely loading the model and thereafter, all other procedures take milliseconds to complete.

3.5. *Solution Approach*

This section describes the working of the three core modules of the CodeUP in detail.

A. *Question Answer (QA) Module*

This module is where people work together to ask and answer each other's questions. This module's operation is straightforward. Users can ask inquiries using a tab as shown in the Fig. 3. In order to ask a question, users are required to provide specific information, including the title or main question, a concise summary of the topic, relevant tags, and the appropriate category. To facilitate this process, we have incorporated the React Quill package for inputting the description. This feature enables users to share links, include code snippets, and utilize basic text editing functions such as different font sizes, bold, italic, and underline formatting options. By employing these capabilities, users can convey their questions more effectively and dynamically. Additionally, it is mandatory to include a minimum of two tags and a maximum of thirty tags. Tags serve the purpose of enabling other users to identify questions relevant to specific topics. Furthermore, the category selection determines the broader module or technology associated with the inquiry. Users can easily search for questions by category by utilizing the "Search Question" tab. Then the user can proceed to click the Ask button by entering the title of the question such that system finds questions similar to the entering question are shown. The user can go through those questions entered and user may select anyone of them if found suitable. Figs 4, 5, and 6 respectively show the interfaces for searching the questions, viewing questions posted by self, and finding similar questions. We built all of the checks in the back-end for the question module such that even if the user tries to circumvent the front-end, the back-end will not allow it. For example, only authorized users can add questions; to do this, we used JWT (JSON Web Token) tokens for authorization. If a user uploads media to a question or answer, it is saved to the AWS (Amazon Web Services) S3 bucket. Also likes and comments in the answer module were saved for user rating analysis. The model was constructed by using referencing instead of embedding comments in such a way that the likes are normalized for user rating.

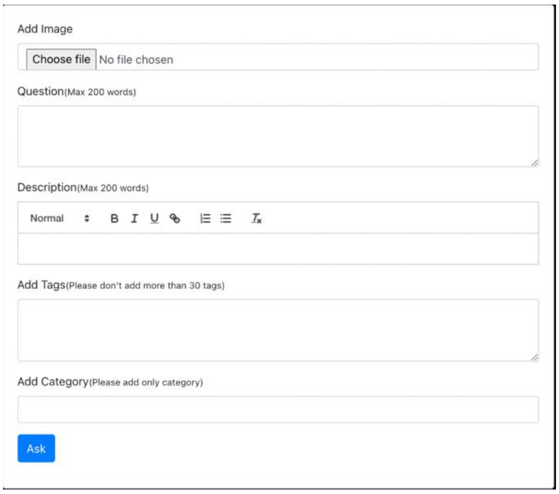


Fig.3. Snapshot of ask question page in CodeUP

The APIs are implemented for each module. The description of APIs implemented for each module and the issues addressed are as follows.

B. *Question Module*

Within the question module, we have integrated several APIs at the backend, each serving a specific purpose:

- An API designed for adding new questions, which functions only when the user is logged in.
- An API developed for updating existing questions, provided that the user attempting the update is the owner of the question.
- An API created for deleting existing questions, with the condition that the user initiating the deletion is the owner of the question.
- An API implemented for retrieving all questions, accessible exclusively to logged-in users.
- An API designed to fetch questions based on their respective categories.
- An API developed specifically for retrieving a question based on its unique ID.

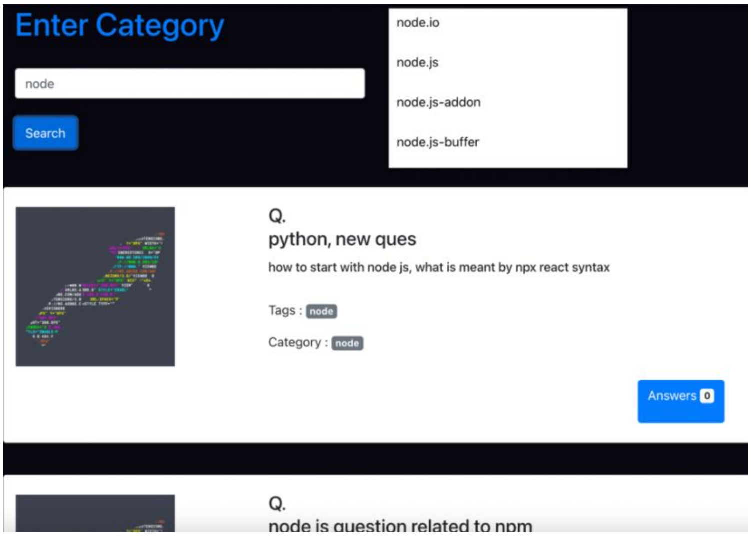


Fig.4. Snapshot of search by category page in CodeUP

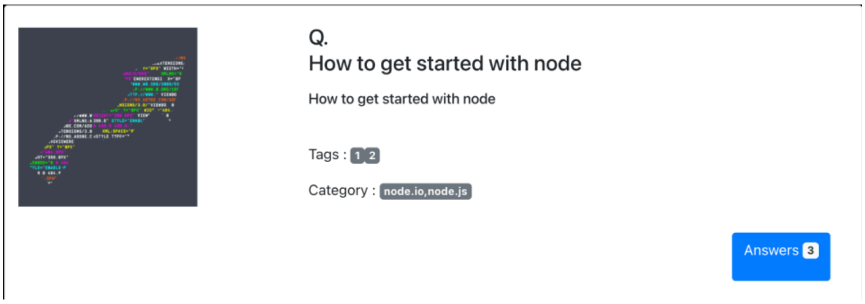


Fig.5. Snapshot of My Question page in CodeUP

During implementation of APIs, for edge case conditions such as changing and removing questions by ID, system first check for the presence of Question, then it checks if that question belongs to that user or not. A user can enter the category of a question while adding it, then system provides a dropdown with all available domains for user convenience. Issues were in deciding which model to use for recommending users with the categories. Finding Fuzzy search will work for dictionary implementation was a great challenge. Thus, we used Fuzzy search for it, which is a procedure that finds Web pages that are likely to be related to a search argument even if the argument does not exactly match the requested information [26].

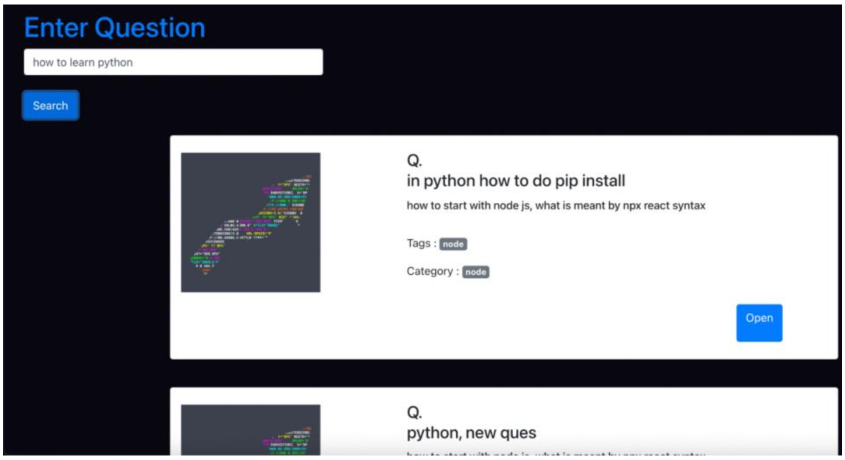


Fig.6. Snapshot of similar question search page in CodeUP

C. Answer Module

In the answer module, we have integrated several APIs at the backend to enhance its functionality. These APIs include:

- An API dedicated to adding new answers, which can only be utilized by logged-in users.
- An API designed for updating existing answers, with the condition that the user attempting the update is the owner of the corresponding question.
- An API developed specifically for deleting existing answers, provided that the user initiating the deletion is the owner of the associated question.
- An API implemented for commenting on answers, allowing users to provide additional feedback or insights.
- An API created for liking answers, enabling users to express their appreciation or agreement with specific responses.

While designing the answer module, load has been taken into consideration. In general, embedding provides better performance for read operations, as well as the ability to request and retrieve related data in a single database operation. Embedded data models make it possible to update related data in a single atomic write operation. To handle multiple users we have used referencing in likes and comments, i.e., normalized mode to 1) represent more complex many-to-many relationships, and 2) model large hierarchical data sets.

D. Profile Rating Module

With the growing popularity of Question-and-Answer platforms like StackOverflow, Ask, and Yahoo! Answers, it is crucial to address the issue of maintaining high-quality questions and answers. As these platforms gain traction, ensuring the quality of the content becomes a paramount concern. In Quora and Stack Overflow users can ask questions, and can even provide their answers [20, 27, 28]. The credibility and expertise of the questioner and the respondents in the field of the question is one of the solutions to get around this problem. In other words, individuals with a high level of expertise can answer these questions appropriately. So Profile rating plays an important role. Providing similar questions, whenever a user searches for a question, is one of the most important features. Generally, the users with high reputation manually analyze and mark duplicate questions. Stack Overflow provides profile rating based on user profile details such as consistency of user, for how long the user is a visitor to the site, number of answers and likes. But visiting the site each day or for 100 consecutive days, and one answer getting 10 likes and ten answers getting 1 like do not show real expertise of the user. The rationale for this requirement is that there may be numerous answers to each query, and some of these solutions may be of poor quality. One answer to this challenge is the credibility and knowledge of the questioner and respondents in the topic of the question. In other words, people with a high degree of skill ask more challenging and high-quality questions in their field of competence, and those with a high level of experience can appropriately answer these questions. This Rating Module assigns ratings to users depending on the quality of their responses (relevancy). It takes into account a variety of parameters, such as how many likes a user has received and how many answers a person has provided. What is the user's consistency? The difficulties were in determining the formula and the way in which rating functionality is added. Maintaining a balance between these aspects was also difficult. However, we developed a simple formula to compute the user profile rating based on the likes and amount of responses. Let $U = \{U_1, U_2, \dots, U_n\}$ are the users of the system, and $\{A_{i1}, A_{i2}, \dots, A_{in}\}$ are the answers provided by the user U_i, then based on the likes and responses the users profile rating is computed in two stages such as 1) computing mean scores, and 2) computing profile rating as follows

1) Computing All Users Mean Scores

Iterate over all users U in the system
    For each user U_i,
        Iterate over all answers U_i provided and arrange the likes to answers in ascending order
        Consider the mid value of sorted array as **score**, number of answers as **length**
        mean_score_U_i = score * length
        allUserScores[U_i] = mean_score_U_i

2) Computing User Rating

max_rating = maximum(allUserScores[U_i])
Iterate over all users U in the system
    For each user U_i,
        profile_rating_U_i = (mean_score_U_i * 5) / max_score

We estimated mean likes and provided ratings ranging from 1 to 5, for this the user's consistency were taken into account [14, 15]. Fig. 7 shows the user profile interface in CodeUP. It consists of the details such as social platforms profile links, technologies and skills the user processes, and experience details. A user can update the skill set and the profile information any time. However the rating will be given by the CodeUP based on the number of answers provided and likes to the answer as described above.

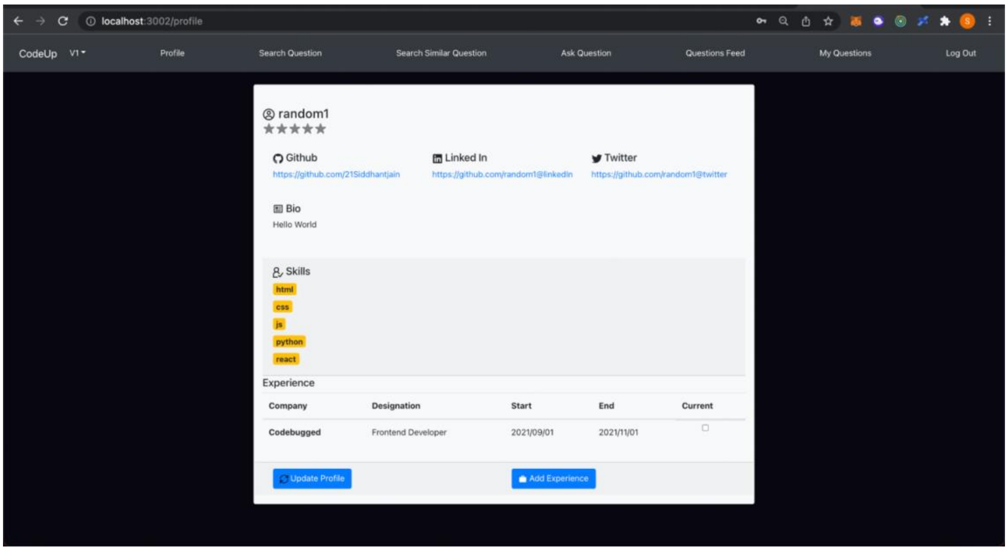


Fig.7. Snapshot of user profile page in CodeUP

E. Question Similarity Module

In this module, we utilized BERT, Universal sentence encoder (USE), and Electra sentence embedding utilizing BERT to determine whether or not a question exists in our database [18, 17, 29, 30]. The following presents the description of these models.

- Semantic Textual Similarity: In the study "Learning Semantic Textual Similarity from Conversations," a novel approach for learning sentence representations for semantic textual similarity is introduced. The idea behind this approach is that sentences exhibit semantic similarity when they elicit similar responses. For instance, questions like "How old are you?" and "What is your age?" are both related to age and can be answered with similar responses such as "I am 20 years old." On the other hand, although "How are you?" and "How old are you?" share almost identical words, they have distinct meanings and lead to different responses [31].
- Universal Sentence Encoder: The Universal Sentence Encoder is a model that enhances the multitask training method mentioned earlier by incorporating additional tasks. These tasks are jointly trained with a model similar to skip-thought, where the objective is to predict the sentences surrounding a given text snippet. Unlike the original encoder-decoder architecture of skip-thought, this model employs an encode-only architecture using a shared encoder for driving the prediction tasks. This design reduces training time significantly while maintaining performance across various transfer tasks, including sentiment analysis and semantic similarity classification.
- BERT: BERT (Bidirectional Encoder Representations from Transformers) is presented in the recent article by Google AI Language researchers [4, 32]. It has sparked interest in the Machine Learning community by presenting cutting-edge findings in a wide range of NLP tasks, such as Question Answering (SQuAD v1.1), Natural Language Inference (NLI), and others. The use of bidirectional training of transformer, a popular attention model, to language modeling is BERT's key technical breakthrough. Previously, researchers focused at a text sequence from left to right or a combination of left-to-right and right-to-left training. The findings reveal that bidirectional trained language models have a better grasp of language context and flow than single-direction language models. Authors described a unique technique called Masked LM (MLM) in the publication, which allows bidirectional training in models that were previously unachievable.
- ELECTRA: It is a self-supervised language representation learning system. It can be used to pre-train transformer networks with a small amount of compute. ELECTRA models, like GAN discriminators, are trained to identify "genuine" input tokens from "false" input tokens created by another neural network. ELECTRA performs well on a small scale, even when trained on a single GPU. On the SQuAD 2.0 dataset, ELECTRA achieves cutting-edge outcomes at big scale [33]. This repository contains code for pre-training ELECTRA, including tiny ELECTRA models that may be trained on a single GPU. It also allows user to fine-tune ELECTRA on downstream tasks such as classification (e.g., GLUE), QA (e.g., SQuAD), and sequence tagging (e.g., text chunking). This repository also includes code for Electric, an ELECTRA variant inspired by energy-based models. Electric offers a more fundamental interpretation of ELECTRA as a "negative sampling" cloze model. It can also efficiently generate pseudo-likelihood scores for text, which can be used to re-rank speech recognition or machine translation system outputs.

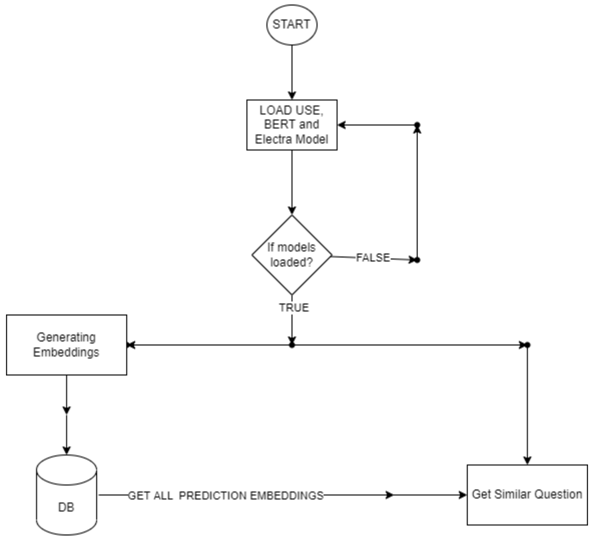


Fig.8. Control flow diagram of Sentence Similarity in CodeUP

Fig. 8 shows the control flow of the Question Similarity module. This module consists of three endpoints:

- Loading NLU Model(USE, BERT and Electra)
- Generating embeddings for the given question title.
- Obtain Top 3 Similar Questions are obtained using Average of Cosine Similarity of three predictions generated by the respective model [34].

4. Software Implementation and Testing

4.1. Content Delivery

The following technologies has been used for implementation of the CodeUP

- Axios - Axios is a JavaScript library that allows making HTTP requests from both Node.js and web browsers. It supports the Promise API and is commonly used for fetching data from APIs and sending data from users to APIs in CodeUP.
- Bootstrap - Bootstrap, a responsive CSS framework, is employed in CodeUP for web development. It offers design templates and components for various interface elements such as typography, forms, buttons, and navigation. Bootstrap is utilized for styling the web pages, making them visually appealing. Components like containers, buttons, badges, and grids are used extensively, with the grid component providing precise control over the layout.
- Fuse.js - Fuse.js is a lightweight and powerful fuzzy-search library that enables approximate string matching. It is utilized in CodeUP to provide users with a list of categories for selecting the appropriate category related to their question. It is also used for searching and finding similar questions.
- React Js - React is a JavaScript library utilized in CodeUP to develop dynamic and interactive web pages through the use of UI components. The integration of Redux with React enabled effective state management within the application. Multiple components were created, including a Spinner for loading content and a Private Route for authentication. CodeUP also includes an alert component to notify users of actions such as adding questions or answers, and finding similar questions. React hooks, such as useState and useEffect, were employed for state management and post-rendering actions. React Router's Link component facilitated page navigation, while the useHistory component handled page redirection. Overall, React, Redux, and various components and hooks were utilized to enhance the functionality and user experience of CodeUP.
- React-Icons - The react-icons library is integrated into CodeUP to easily include popular icons in React projects. It supports importing only the icons needed for the project, enhancing performance. These high-quality vector icons are utilized to enhance the visual appeal of the website, including icons for GitHub, LinkedIn, Twitter, likes, comments, expansions, profiles, and more.
- Redux - It is an open-source JavaScript library used for centralized state management in applications. It is frequently used in conjunction with frameworks like React or Angular to build user interfaces. In CodeUP, Redux plays a vital role in managing the application's state. Different reducers are utilized for modules such as answers, questions, and alerts, ensuring efficient handling of data displayed on the frontend without the need for repeated backend requests.

4.2. NLU Application Programming Interface

Issue observed during this module was selecting the best operating system. Since loading the model requires Linux Environment and better performance in terms of space and time we had to deploy our API's using Colab and NGrok. The following libraries are used to address this:

- Cosine Similarity - The similarity of two vectors in an inner product space is measured by cosine similarity. It is calculated by taking the cosine of the angle between two vectors and determining whether two vectors are pointing in the same general direction. It is frequently used in text analysis to assess document similarity [33].
- NLU - NLU stands for natural language understanding; it assists data scientists in comprehending text data produced in human languages by reducing the NLP component to no more than one line of code. Instead of focusing on data processing and feature engineering, a data scientist might use NLU to focus on understanding the natural language data [35, 36].

4.3. CodeUP Application Programming Interface

- Flask ngrok: Flask ngrok is a simple solution for showcasing Flask applications from your local machine. It leverages the powerful ngrok utility to expose your Flask apps running on localhost to the internet, making them accessible remotely.
- Flask: Flask is a micro web framework based on Python. It is categorized as a micro framework because it does not impose any specific tools or libraries for usage. While Flask lacks certain components like a database abstraction layer and form validation, it allows for the integration of extensions that can enhance the application with additional functionalities seamlessly. These extensions cover a wide range of features, including object-relational mappers, form validation, file upload handling, support for various open authentication protocols, and other useful tools related to the framework.
- PySpark: PySpark serves as a Python interface for Apache Spark, offering a convenient way to develop Spark applications using Python APIs. It includes the PySpark shell, which allows for interactive data exploration within a distributed context. PySpark provides compatibility with various features of Spark, such as Spark SQL, DataFrame, Streaming, MLlib (Machine Learning), and Spark Core.

4.4. CodeUP Testing

Software testing is the process of evaluating the functioning of a software programme in order to determine whether the generated software fits the defined criteria and to discover defects in order to deliver a quality product. It is a critical stage of the software development life cycle since it evaluates and validates the system under test. Testing is critical for a recommendation system to assure system quality and functionality before delivery. To evaluate system output, proper quality assessment procedures must be used to identify system performance in comparison to the benchmark level or similar products. During the testing process we encountered several risks in running the applications. These risks were mitigated by incorporating latest technical solutions as shown in the Table 1. Table 2 shows the list of test cases prepared and tested for each module.

Table 1. Risks observed vs. mitigation followed in CodeUP

Sno	Risks observed	Mitigation followed
1	If too many uses come simultaneously to the site, then they might suffer delay in adding answers and questions.	To avoid delay, we have used asynchronous functions in the models, so that the tasks which consume time get executed in the backend.
2	Fetching and uploading media might take a long time.	To avoid delay in fetching and uploading media we have used a cloud service AWS, S3 bucket.
3	Updating profile rating each time a user likes the answer, might lead to a lot of load on the system.	Have created a separate script to update the profile rating, which could be run on a daily or weekly basis.
4	Unauthorized users try to access functions which are only for authorized users using URL manipulation.	To avoid unauthorized access of data we have created private routes which will check if the user is authorized on every webpage.
5	If user tries to bypass the frontend through any tool.	We have added authentication to the backend such that if any user bypass the frontend, backend will stop it.

Fig. 9 shows the summary of test result after running the test cases developed for question module for 3 times. Fig. 10 shows the summary of test result after running the test cases developed for question module for 5 times. Fig. 11 shows that summary of test results after running the test cases for profile module. It can be observed from the test result that the all the test cases has been executed successfully.

Table 2. Test cases developed for the CodeUP

Sno	Module	Test cases
1	Question module	• Only users with authorization can add new questions • Questions must have description, category, and title. • Before adding a question, similar questions should be predicted • Answers can be written in a question • Fetching a question with large number of answers • Load balancing mechanisms are implemented to manage the retrieval of multiple questions simultaneously.
2	Answer module	• Answer must have description • While fetching answer, comments should also come • Checking for condition with too many comments • Checking for condition with too many likes
3	Profile rating	• Profile rating is based on the number of likes a user has got. • It is also based on the number of answers a user has given. • If the number of users increases, and the number of answers given by each is huge, testing for it.
4	Question similarity	• Few questions are prepared to check against the testing data.

	executed	failed
iterations	3	0
requests	27	0
test-scripts	24	0
prerequest-scripts	0	0
assertions	99	0
total run duration: 37.4s		
total data received: 1.42MB (approx)		
average response time: 1289ms [min: 11ms, max: 6.1s, s.d.: 1796ms]		

Fig.9. Summary of test results for question module

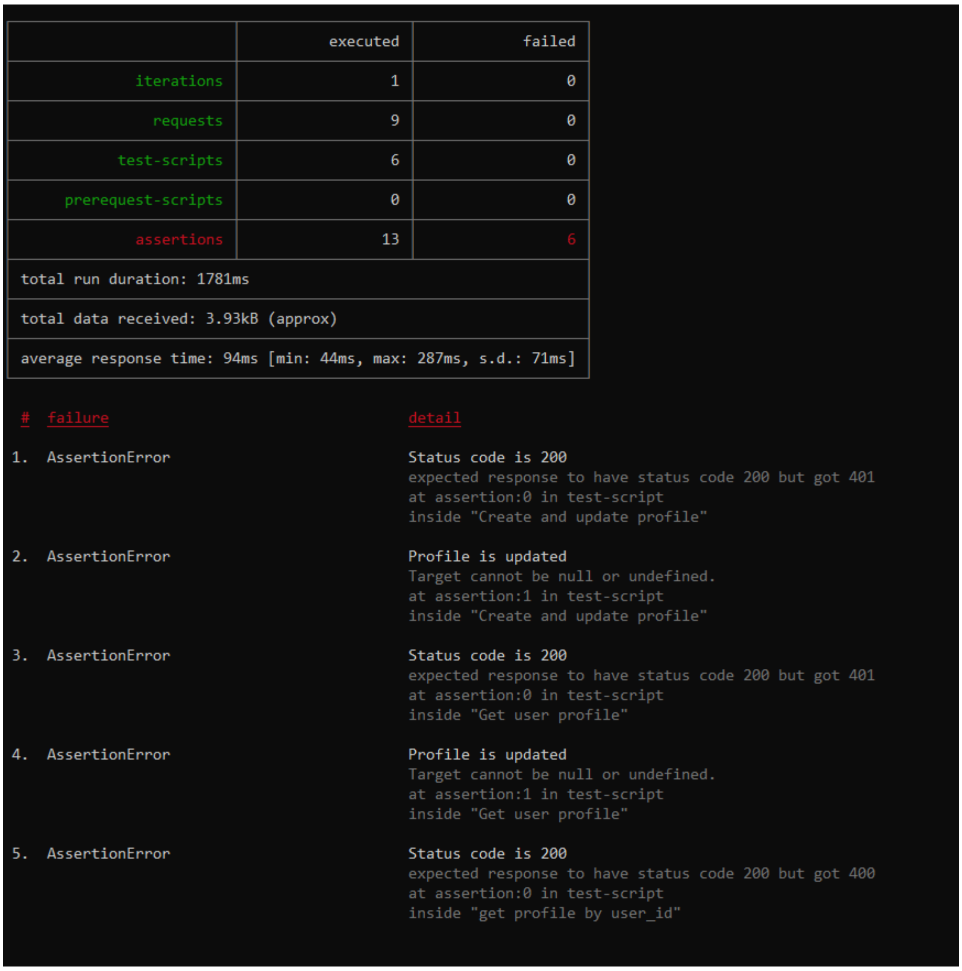
	executed	failed
iterations	5	0
requests	30	0
test-scripts	30	0
prerequest-scripts	0	0
assertions	110	0
total run duration: 9.8s		
total data received: 11.21kB (approx)		
average response time: 232ms [min: 93ms, max: 549ms, s.d.: 136ms]		

Fig.10. Summary of test results for answer module

	executed	failed
iterations	1	0
requests	9	0
test-scripts	6	0
prerequest-scripts	0	0
assertions	13	0
total run duration: 2s		
total data received: 6.93kB (approx)		
average response time: 135ms [min: 44ms, max: 290ms, s.d.: 68ms]		

Fig.11. Summary of test results for Profile module

Fig. 12 shows the test result report when the request is invalid. It can be observed from the report that when the request is invalid the test conditions failed to execute and related errors were displayed.



	executed	failed
iterations	1	0
requests	9	0
test-scripts	6	0
prerequest-scripts	0	0
assertions	13	6
total run duration: 1781ms		
total data received: 3.93kB (approx)		
average response time: 94ms [min: 44ms, max: 287ms, s.d.: 71ms]		

#	failure	detail
1.	AssertionError	Status code is 200 expected response to have status code 200 but got 401 at assertion:0 in test-script inside "Create and update profile"
2.	AssertionError	Profile is updated Target cannot be null or undefined. at assertion:1 in test-script inside "Create and update profile"
3.	AssertionError	Status code is 200 expected response to have status code 200 but got 401 at assertion:0 in test-script inside "Get user profile"
4.	AssertionError	Profile is updated Target cannot be null or undefined. at assertion:1 in test-script inside "Get user profile"
5.	AssertionError	Status code is 200 expected response to have status code 200 but got 400 at assertion:0 in test-script inside "get profile by user_id"

Fig.12. Summary of test results when all the requests are invalid

During testing the combination of NLU models (Electra, BERT and USE) used in CodeUP, it was found that when a question entered which is getting compared with our testing data and getting top similar questions with maximum similarity. Fig. 13, 14 and 15 shows the similarity for the questions “how to install linux?”, “How to find minimum value in numpy array?” and “How to sort an array in Java?”, It can be observed from the Fig.s that the questions similarity identified are almost similar to the question asked.

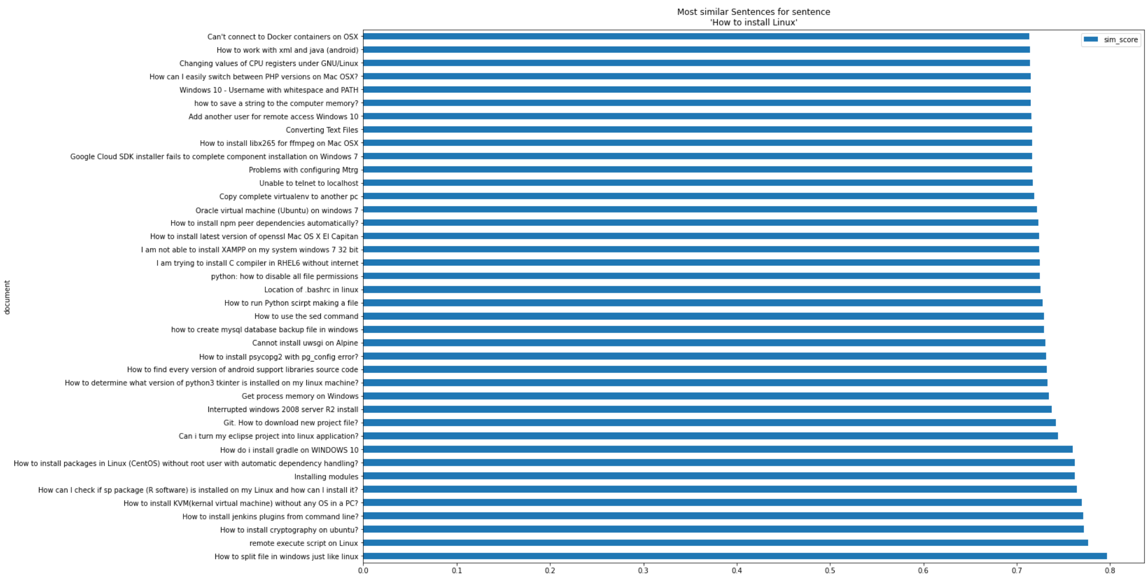


Fig.13. Sentence similarity for the question "How to install Linux?"

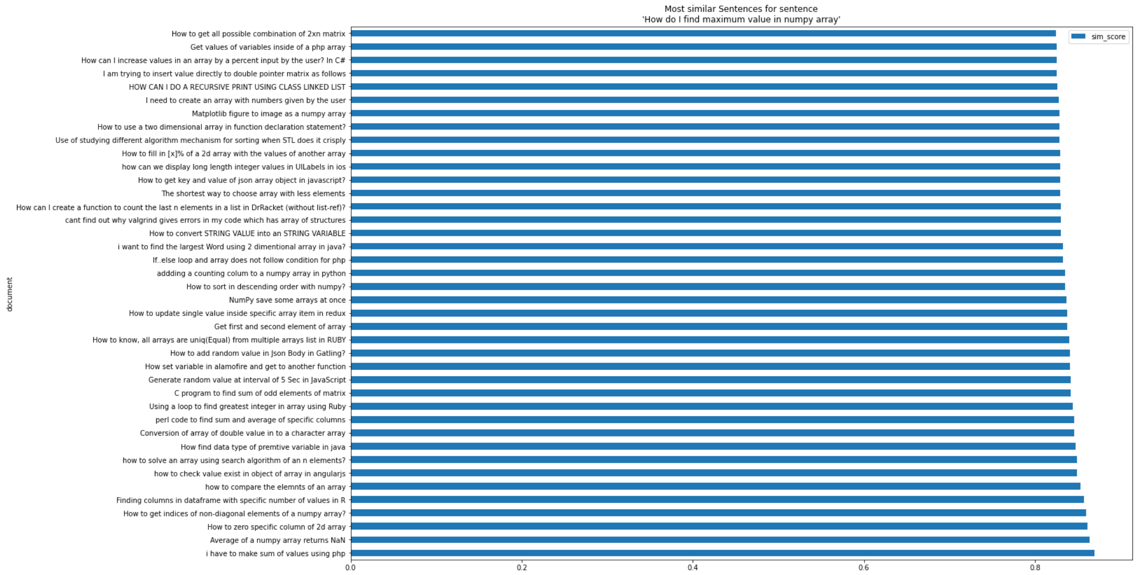


Fig.14. Sentence similarity for the question "How to find minimum value in numpy array?"

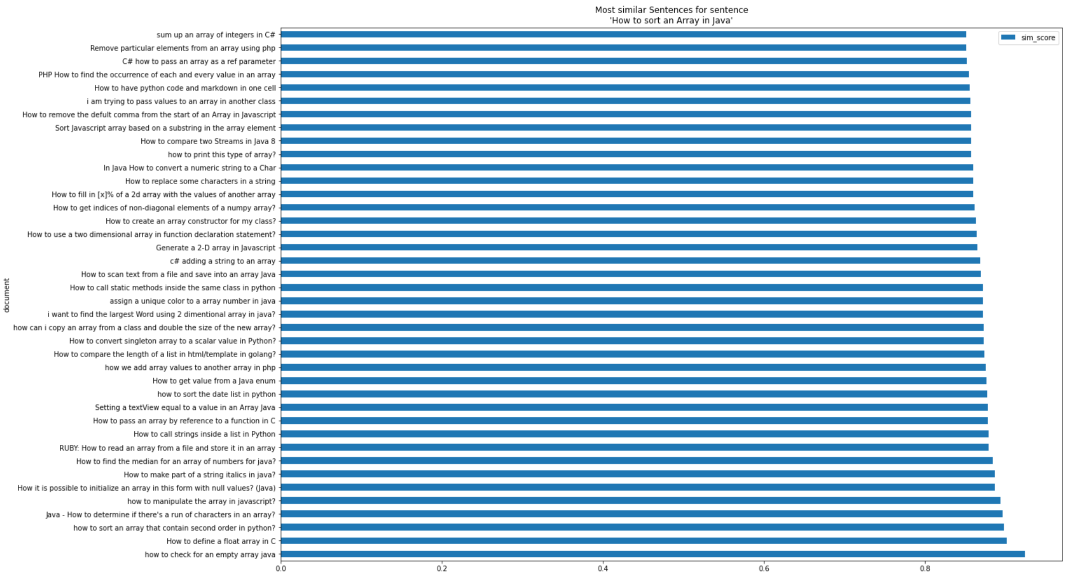


Fig.15. Sentence similarity of question "How to sort an array in Java?"

Fig. 16 shows the heatmap of the sentence similarity of the questions entered in the CodeUP. It is observed from the figure that the system is able to identify the similar questions entered by the user. To assess the performance of the CodeUP system, we conducted an experiment by deploying it on the Local Area Network (LAN) of the University. Total 30 final year graduate students were selected to utilize CodeUP as a collaborative platform for their minor project work. The feedback received from the users indicated a high level of satisfaction with the system, particularly highlighting its user-friendly interface and the effectiveness of the Q&A recommendations in facilitating information retrieval.

5. Conclusions and Future Work

This paper presents a web application, called CodeUP, with the features such as QA system, user profile rating and question similarity. CodeUp is a web application for collaborative knowledge sharing platform, where users can build up their community, by searching for other users within the same domain. Users also get a detailed view about the other users profile like the technical skills, experiences in programming and the profile rating which will give an idea about knowledge of that user. CodeUP is a web-based collaboration tool created specifically for users who are looking for high-quality programming-related content and who want to interact with other developers and who share their interests. As a future work, we attempt to extend the CodeUP to accomodate recommending questions to the user based on the profile and interaction with questions, integrating github data to improve profile rating, integrating chat bots, and chat systems to allow users to have a better experience.

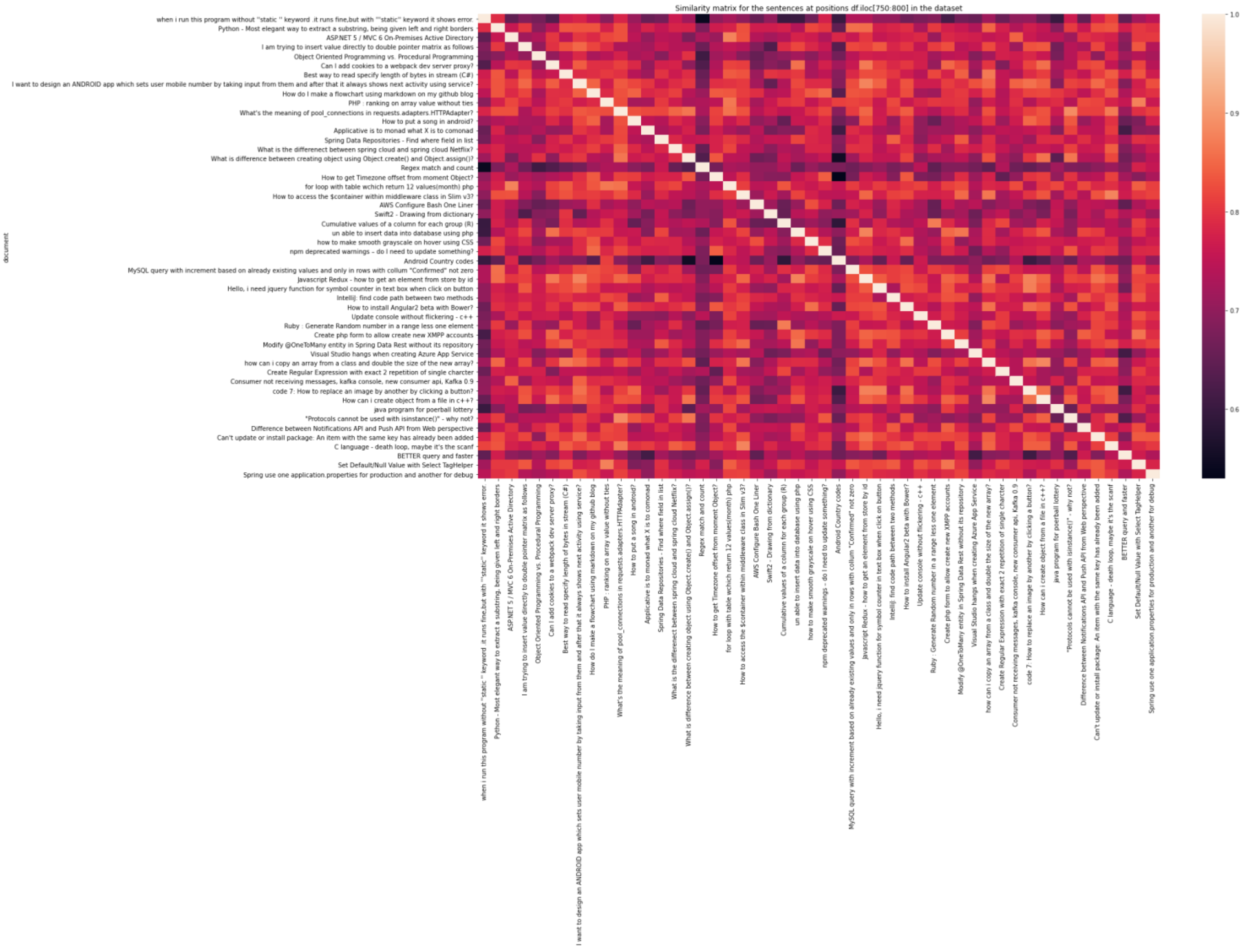


Fig.16. Heatmap of sentence similarity

References

[1] D. I. Castaneda and S. Cuellar, "Knowledge sharing and innovation: A systematic review," Knowledge and Process Management, vol. 27, p. 159–173, 2020.

[2] T.-M. Nguyen, "Four-dimensional model: a literature review in online organisational knowledge sharing," VINE Journal of Information and Knowledge Management Systems, 2020.

[3] J. Yao, A. Crupi, A. Di Minin and X. Zhang, "Knowledge sharing and technological innovation capabilities of Chinese software SMEs," Journal of Knowledge Management, 2020.

[4] Y. Sun, X. Zhou, A. Jeyaraj, R.-A. Shang and F. Hu, "The impact of enterprise social media platforms on knowledge sharing: An affordance lens perspective," Journal of Enterprise Information Management, 2019.

[5] P. Setialana, A. Fitria, R. Atika, R. N. Fadilla and others, "Development of WeShare As a Knowledge Sharing Platform to Realize the Freedom in Learning," in Journal of Physics: Conference Series, 2021.

[6] P. V. Rao and A. P. S. Kumar, "The societal communication of the Q&A community on topic modeling," The Journal of Supercomputing, vol. 78, p. 1117–1143, June 2021.

[7] D. A. Prabowo and G. B. Herwanto, "Duplicate Question Detection in Question Answer Website using Convolutional Neural Network," in 2019 5th International Conference on Science and Technology (ICST), 2019.

[8] Ø. Tønnessen, A. Dhir and B.-T. Flåten, "Digital knowledge sharing and creative performance: Work from home during the COVID-19 pandemic," Technological Forecasting and Social Change, vol. 170, p. 120866, 2021.

[9] J. Hao, L. Liu, A. A. von Davier, N. Lederer, D. Zapata-Rivera, P. Jakl and M. Bakkenson, "EPCAL: ETS platform for collaborative assessment and learning," ETS Research Report Series, vol. 2017, p. 1–14, 2017.

[10] S. Chatterjee, N. P. Rana and Y. K. Dwivedi, "Social media as a tool of knowledge sharing in academia: an empirical study using valance, instrumentality and expectancy (VIE) approach," Journal of Knowledge Management, 2020.

[11] V. S. Rekha and S. Venkatapathy, "Understanding the Usage of Online Forums as Learning Platforms," Procedia Computer Science, vol. 46, p. 499–506, 2015.

[12] R. O. de Castro, C. Sanin, A. Levula and E. Szczerbicki, "The Development of a Conceptual Framework for Knowledge Sharing in Agile IT Projects," Cybernetics and Systems, vol. 53, p. 529–540, 2022.

[13] Y. Feng, Z. Yi, C. Yang, R. Chen and Y. Feng, "How do gamification mechanics drive solvers' Knowledge contribution? A study of collaborative knowledge crowdsourcing," Technological Forecasting and Social Change, vol. 177, p. 121520, April 2022.

[14] M. S. Faisal, A. Daud, A. U. Akram, R. A. Abbasi, N. R. Aljohani and I. Mehmood, "Expert ranking techniques for online rated forums," Computers in Human Behavior, vol. 100, p. 168–176, November 2019.

[15] A. Diyanati, B. S. Sheykhahmadloo, S. M. Fakhrahmad, M. H. Sadredini and M. H. Diyanati, "A proposed approach to determining expertise level of StackOverflow programmers based on mining of user comments," Journal of Computer Languages, vol. 61, p. 101000, December 2020.

[16] Y. Yang, S. Yuan, D. Cer, S.-y. Kong, N. Constant, P. Pilar, H. Ge, Y.-H. Sung, B. Strope and R. Kurzweil, "Learning Semantic Textual Similarity from Conversations," in Proceedings of The Third Workshop on Representation Learning for NLP, 2018.

[17] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope and R. Kurzweil, "Universal Sentence Encoder for English," in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2018.

[18] Y. Yang and C. Tar, "Advances in semantic textual similarity," Google AI Blog, 2018.

[19] J. Zhang and N. M. El-Gohary, "Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking," Journal of Computing in Civil Engineering, vol. 30, p. 04015014, 2016.

[20] L. Wang, L. Zhang and J. Jiang, "Duplicate question detection with deep learning in stack overflow," IEEE Access, vol. 8, p. 25964–25975, 2020.

[21] T. Gogar, O. Hubacek and J. Sedivy, "Deep neural networks for web page information extraction," in IFIP International Conference on Artificial Intelligence Applications and Innovations, 2016.

[22] S. Pang, P. Bao and W. Gu, "Research into Consumers' Continued Use Intention of Knowledge Sharing Platforms," in Education and Awareness of Sustainability: Proceedings of the 3rd Eurasian Conference on Educational Innovation 2020 (ECEI 2020), 2020.

[23] L. Nie, Y. Li, F. Feng, X. Song, M. Wang and Y. Wang, "Large-Scale Question Tagging via Joint Question-Topic Embedding Learning," ACM Transactions on Information Systems, vol. 38, p. 1–23, March 2020.

[24] S. K. Maity, A. Panigrahi, S. Ghosh, A. Banerjee, P. Goyal and A. Mukherjee, "DeepTagRec: A Content-cum-User Based Tag Recommendation Framework for Stack Overflow," in Lecture Notes in Computer Science, Springer International Publishing, 2019, p. 125–131.

[25] M. S. M. Jabbar, L. Kumar, H. W. Samuel, M.-Y. Kim, S. Prabharkar, R. Goebel and O. Zaiane, "DeepDup: Duplicate Question Detection in Community Question Answering," in 2021 5th International Conference on Deep Learning Technologies (ICDLT), 2021.

[26] M. M. Tambakhe and D. P. Wagh, "Duplicate Question Pair Detection with Machine Learning," BULLETIN MONUMENTAL, vol. 22, 2021.

[27] V. K. R. Anishaa, P. Sathvika and S. Rawat, "Identifying Similar Question Pairs Using Machine Learning Techniques," Indian Journal of Science and Technology, vol. 14, p. 1635–1641, May 2021.

[28] G. Elangovan, J. Umamageswaran, G. Indumathi and A. V. Kalpana, "Medical Quora Tagging using MATAR and LDA Algorithm," Journal of Physics: Conference Series, vol. 1964, p. 042029, July 2021.

[29] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou and P. Jiang, "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer," in Proceedings of the 28th ACM international conference on information and knowledge management, 2019.

[30] K. Clark, M.-T. Luong, Q. V. Le and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," arXiv preprint arXiv:2003.10555, 2020.

[31] M. K. Ho, "Question classification via machine learning techniques," Nanyang Technological University.

[32] Annamoradnejad, M. Fazli and J. Habibi, "Predicting Subjective Features from Questions on QA Websites using BERT," in 2020 6th International Conference on Web Research (ICWR), 2020.

[33] A. Dhakal, A. Poudel, S. Pandey, S. Gaire and H. P. Baral, "Exploring Deep Learning in Semantic Question Matching," in 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), 2018.

[34] R. Lahitani, A. E. Permanasari and N. A. Setiawan, "Cosine similarity to determine similarity measure: Study case in online essay assessment," in 2016 4th International Conference on Cyber and IT Service Management, 2016.

[35] D. Meenakshi and A. R. M. Shanavas, "Transformer induced enhanced feature engineering for contextual similarity detection in text," Bulletin of Electrical Engineering and Informatics, vol. 11, p. 2124–2130, August 2022.

[36] O. Habeh, F. Thekrallah, S. A. Salloum and K. Shaalan, "Knowledge sharing challenges and solutions within software development team: a systematic review," Recent Advances in Intelligent Systems and Smart Applications, p. 121–141, 2021.

Authors' Profiles

Yashi Aggarwal currently pursuing final year of Integrate B.Tech M.Tech in Computer Science and Engineering at Jaypee Institute of Information Technology, Noida, India. Her programming interests include Backend Development, Cloud Computing, AWS and Competitive Programming.

P. Raghu Vamsi is currently working as Assistant Professor (Senior Grade) in the Department of Computer Science and Engineering (CSE), Jaypee Institute of Information Technology (JIIT), NOIDA, UP, India from July 2016. He received B.E in CSE from University of Madras in 2003, M.Tech (Software Engineering) from Kakatiya University in 2007, M.B.A in HRM from IGNOU in 2010, and M.A in Astrology from PSR Telugu University, Hyderabad in 2014, and PhD in CSE from JIIT, NOIDA in 2016. He has cleared technical certifications such as Core Java certification by Brain Bench, IBM DB2 Application and Fundamentals, and IBM Rational Application Developer for Web sphere. He has more than 15 years of teaching experience in various engineering institutions and 4 year of research experience. So far, he guided several undergraduate and post graduate projects and dissertations. His research interests include security in networks and communication, Wireless Adhoc Networks, IoT and Blockchain. He is a life member of Indian Society for Technical Education (ISTE) India.

Siddhant Jain currently pursuing final year of Integrate B.Tech M.Tech in Computer Science and Engineering at Jaypee Institute of Information Technology, Noida, India. His programming interests include Ethereum Smart Contract development, and React application development.

Jayanth Goel received the Bachelor of Technology in Computer Science and Engineering from Jaypee Institute of Information Technology, Noida in 2022. He is currently working at Publicis Sapient, India. His research interest includes applications of machine learning, natural language process, web application development, and Blockchain.

How to cite this paper: Yashi Agarwal, P. Raghu Vamsi, Siddhant Jain, Jayant Goel, "CodeUP: A Web Application for Collaborative Question-answering System", International Journal of Information Technology and Computer Science(IJITCS), Vol.15, No.4, pp.33-49, 2023. DOI:10.5815/ijitcs.2023.04.04