International Current Journal of Engineering and Science (ICJES)
Volume 4, Issue 5, pp 12-17, May-2025
ISSN (Online): 2583-7257
Article DOI: https://doi.org/10.47001/ICJES/2025.405003

Smart Educational Chatbot with Integrated Quiz Portal for Interactive Student Learning

[1]Prabhakaran R.N., [2]Anirudh Vikas P

[1,2]Department of Electronics and Telecommunication Engineering, RV College of Engineering, Bengaluru, India

***Abstract* - Interactive learning system was designed specifically for students and lecturers of a 1st year object-oriented programming (OOP) course. The aim of this system is to provide an engaging and effective learning experience by offering personalized assistance through a chatbot and reinforcing concepts through interactive quizzes. The objective of the lecturers is to be able to get an oversight of student performance and predict their likelihood of success in the course at an early stage based on their interactions with the system. The chatbot component of the system acts as a virtual tutor, capable of understanding and responding to student queries related to OOP concepts. It leverages natural language processing techniques to comprehend user inputs and generate informative and contextually relevant responses. The chatbot's responses are designed to offer explanations, examples, and clarifications, thereby promoting deeper understanding of OOP principles. The quiz portal component provides students with a diverse range of interactive quizzes that assess their knowledge of OOP concepts. The quizzes are dynamically generated, ensuring variability in questions and difficulty levels. Students can choose from multiple-choice to reinforce their theoretical understanding and practical application of OOP principles. Educational chatbots attempt to improve learning and lecturer insights, they struggle with issues such as data access and the lack of individualized attention. A solution to this suggests leveraging data mining and ontology to develop dynamic quiz chatbot knowledge.**

Keywords: Q&A Chatbot, Software development, Quiz system, Machine learning.

I. INTRODUCTION

A recent emphasis on interactive systems designed for certain courses is one example of how the quick development of technology has generated novel pedagogical techniques. A crucial turning point in an IT student's academic career is the core first-year object-oriented programming (OOP) curriculum. We explore the development of an interactive learning system that is designed to enhance how students interact with the subject matter while acknowledging the challenges involved in gaining an understanding of OOP principles. Understanding the crucial role that powerful learning tools play in the current educational context. A recent emphasis on interactive systems designed for certain courses is one example of how the quick development of technology has generated novel pedagogical techniques. A crucial turning point in a student's academic career is the core first-year object-oriented programming (OOP) curriculum. We explore the development of an interactive learning system that is set to revolutionize how students interact with the subject matter while acknowledging the challenges involved in grasping OOP principles.

A) Problem statement

Traditional OOP teaching strategies, which rely on lectures and static resources, frequently fall short of supporting the different learning demands of students like us. The lack of individualized attention during lectures and the restricted opportunities for students to actively reinforce their learning are two issues we experience. These obstacles prevent students from acquiring a thorough understanding of OOP, which may have an impact on their overall learning results and long-term success. Additionally, educators struggle with the challenge of assessing student progress and figuring out early signs of the tendency to excel in the course.

B) Significance

Our initiative to develop an interactive learning system tackles these difficulties head-on, advancing OOP learning into a time of personalized coaching and active participation. We aim to develop an environment where learning goes beyond the classroom by incorporating a sophisticated chatbot and a dynamic quiz portal, giving students a flexible platform to investigate, practice, and internalize OOP principles. Our effort to provide teachers with insights into student interactions is equally important in terms of maximizing student accomplishment through timely interventions and individualized support.

International Current Journal of Engineering and Science (ICJES)
Volume 4, Issue 5, pp 12-17, May-2025
ISSN (Online): 2583-7257
Article DOI: https://doi.org/10.47001/ICJES/2025.405003

C) Objectives

Through the use of cutting-edge technological tools, this research's main goal is to redefine OOP instruction by promoting more thorough understanding and grasp of its concepts. In particular, we want to accomplish the following: Create a customized interactive learning system for the first-year OOP course that uses a chatbot to offer individualized support and a dynamic quiz portal to reinforce topics. By giving teachers a thorough framework that enables them to forecast students' progress based on their interactions with the learning system, proactive assistance measures are made possible. Utilizing cutting-edge NLP approaches to produce insightful and contextually appropriate chatbot responses will improve student engagement and learning results.

Create a large collection of dynamically created tests that may be customized for different levels of difficulty and promote the use of OOP principles in real-world situations.

D) Research questions

How effectively can a chatbot act as a virtual tutor, offering explanations, examples, and clarifications for OOP concepts, thereby promoting a deeper understanding among students?

To what extent does the integration of a dynamic quiz portal enhance students' theoretical knowledge and practical application of OOP principles?

Can the interactions between students and the interactive learning system serve as early indicators of their potential success in the first-year OOP course, and how accurately can these interactions predict their likelihood of achievement?

II. LITERATURE REVIEW

In recent years, educational institutions have embraced technology to enhance the learning experience. Interactive platforms, such as chatbots and quiz portals, have become popular tools for engaging students. This literature review explores the benefits and challenges associated with implementing an Interactive Chatbot and Quiz Portal specifically tailored for Object-Oriented Programming students at SLIIT. Chatbots have proven to be effective tools for student engagement and support. Research by Anderson and Huttenocher (2018) highlights the positive impact of chatbots in providing instant help and personalized learning experiences [1]. Interactive quizzes contribute to active learning and knowledge retention. Martinez and Bannister (2019) emphasize the importance of well-designed quizzes in reinforcing programming concepts [2]. Customization is key when designing chatbots for programming students. Research by Li and Wang (2020) discusses the benefits of tailoring chatbot responses to the specific needs and challenges of OOP learners [3]. Gamifying quizzes can increase student motivation and participation. Johnson et al. (2017) explores the impact of gamification on student engagement in programming-related quizzes [4]. A positive user experience is crucial for the effectiveness of educational chatbots. Research by Chen et al. (2018) discusses the importance of designing chatbots with a user-centered approach [5]. Providing timely and constructive feedback is essential for student learning. Hattie and Timperley (2007) present a model for effective feedback, which can be applied to quiz portals [6]. The use of natural language processing (NLP) enhances the conversational abilities of chatbots. Jurafsky and Martin (2019) provide insights into the application of NLP in chatbot development [7]. As educational platforms collect sensitive data, addressing security and privacy concerns is crucial. Research by Acquisti (2009) discusses the challenges and potential solutions in securing educational data [8]. Evaluating the impact of educational technology is essential for continuous improvement. Tondeur et al. (2017) present a framework for assessing the effectiveness of technology integration in education [9]. The literature supports the potential benefits of an Interactive Chatbot and Quiz Portal for SLIIT OOP students. Customization, gamification, user experience, and security considerations are vital elements in the design and implementation of such educational technology.

III. RELATED WORK

An interactive chatbot and quiz portal for Object-Oriented Programming (OOP) students is an interesting project idea that can encompass various aspects of artificial intelligence, educational technology, and programming education. When looking for related work for such a project.

A) Educational Chatbots and Virtual Tutors

Look for research papers and projects that focus on developing educational chatbots or virtual tutors for programming education. These systems aim to assist students in learning programming concepts, providing explanations, and answering questions.

B) Programming Education Platforms

Investigate existing platforms that offer programming education, particularly those that include interactive components like quizzes, coding exercises, and chatbot-like interactions.

International Current Journal of Engineering and Science (ICJES)
Volume 4, Issue 5, pp 12-17, May-2025
ISSN (Online): 2583-7257
Article DOI: https://doi.org/10.47001/ICJES/2025.405003

C) Conversational AI and NLP

Dive into research on conversational AI, NLP, and dialogue systems. You can explore how these technologies are used to build interactive chatbots, how they handle context, intent recognition.

D) Personalized Learning

Research personalized learning systems that adapt to individual student needs. These systems use AI algorithms to analyze a student's progress and provide tailored learning materials, including quizzes and explanations.

E) User Experience (UX) Design

Explore UX design principles for creating intuitive and engaging educational platforms. Consider how user interface design, interaction flows, and visual elements contribute to the overall user experience.

IV. METHODOLOGY

The methodology encompasses the training of a subtopic classification model, the creation of a question classifier, and their integration into a chatbot application for generating relevant responses and categorizing questions in a database.

A) Subtopic Classification Model Training

A dataset containing user questions and their associated subtopics was sourced from an Excel file. The dataset was divided into training and testing sets using the train test split function, with a test size of 20% and a random state of 42 to ensure reproducibility. User questions were tokenized into individual sentences using sentence tokenization techniques. The tokenized sentences were preprocessed to remove irrelevant information, enhancing the model's learning process. A TF-IDF vectorizer was utilized to transform the tokenized sentences into numerical features, a format suitable for machine learning. A Support Vector Machine (SVM) model with a linear kernel was chosen for its efficacy in text classification tasks. The SVM model was trained using the vectorized training data and their corresponding subtopic labels. The trained SVM model was assessed using the testing data. Predictions were made on the test data, and the accuracy of the model's classifications was measured by comparing the predicted subtopics with the actual subtopic labels. Both the trained SVM model and the TF-IDF vectorizer were serialized using the job library, preserving their learned patterns and vectorization capabilities for future use.

B) Chatbot Application Integration and Response Generation

The serialized SVM model and vectorizer were loaded into the chatbot application. The model is responsible for classifying user queries into subtopics, while the vectorizer prepares input data for prediction. User input was processed to identify conversation starters or enders. If input matches any predefined starters or enders, appropriate responses were generated. For other user inputs, the SVM model predicted the subtopic label using the vectorized input. The associated confidence score was also calculated based on predicted probabilities. If user input contained a significant number of tokens, it was summarized using a pre-trained summarization pipeline. This ensured concise input for accurate classification. Based on the predicted subtopic, the chatbot generated responses specific to that subtopic.

Responses were tailored to the predicted subtopic, allowing the chatbot to provide informative answers. In cases where subtopic prediction confidence was low or when there was no relevant subtopic, the chatbot offered default messages to maintain user engagement.

C) Question Classifier for Database Labeling

A question classifier was designed to categorize user questions into labels for database storage. The dataset used for subtopic classification was extended to include question labels suitable for database organization. The extended dataset was split into training and testing sets or question classification.

A similar procedure was followed for tokenization, vectorization, and training using a Support Vector Machine (SVM) model. The trained model's accuracy in predicting question labels was evaluated on the testing data. For each user query, the trained question classifier predicted an appropriate label. The predicted label was associated with the corresponding answer and stored in the database for efficient categorization and retrieval.

D) Performance Evaluation

The chatbot application's performance was assessed by feeding it a range of user queries and analyzing the generated responses. Accuracy of subtopic prediction, coherence of generated responses, adherence to conversation rules, and question classification accuracy were evaluated.

This comprehensive methodology covers subtopic classification, chatbot response generation, and the additional process of question classification for database labeling. The training and integration of models, along with the evaluation

International Current Journal of Engineering and Science (ICJES)
Volume 4, Issue 5, pp 12-17, May-2025
ISSN (Online): 2583-7257
Article DOI: https://doi.org/10.47001/ICJES/2025.405003

of performance, ensure the chatbot's accuracy and efficiency in response generation and question categorization.

E) Data Collection and Preprocessing

To reduce noise and increase the effectiveness of the language model, a corpus of text documents was gathered and preprocessed. Various pre-processing methods were used on the papers, such as deleting stop words, punctuation, and other unneeded content in order to provide responses to user inquiries, a pre-trained language model was deployed. The language model was chosen because it proved effective for jobs requiring text creation. To extract semantic meaning from the text, instructor embeddings were employed. As a result, the language model was able to produce more pertinent and accurate results.

F) Document storage and embedding

Instructor embeddings were used to embed the documents in the corpus. This made it possible to index and retrieve the documents more quickly. In a database, the embedded documents were kept for further use. To link the system's many parts together, a question-answering chain was built. Through the retrieval of pertinent documents and the application of the language model to produce text, this chain enabled the system to provide replies to user inquiries. Answers to user inquiries were generated using the language model. The context of the query and the content of the obtained documents were used to produce the replies. The user could easily read the prepared replies that were provided.

A range of user inquiries were used to gauge the system's performance. The produced replies' accuracy, applicability, and coherence were assessed in comparison to anticipated results. Measurements were made of the system's effectiveness in terms of reaction time and resource use.

G) Web scraping for Additional Answers

To give thorough and varied responses to user inquiries, a further component was added to the system. This part required web scraping from the well-known programming Q&A website Stack Overflow. Obtaining pertinent information about the user's inquiry was the aim of web scraping.

A system was created to save the extracted answers and the labels that go with them. This made it possible for the system to quickly find and show users the solutions to their inquiries. The extracted response, the related labels, the original question, and any other metadata required for future reference were all saved with the answers in an organized format.

The stored answers and their associated labels were included into the question-answering chain's existing retrieval and presentation elements without any issues. This made sure that consumers could choose from a wide variety of high-quality responses.

Testing and validation were used to assess the precision and potency of the question classifier. In comparison to known categories, the classifier's capacity to forecast appropriate labels for various queries was evaluated. Based on aspects including response speed and usability, the database's effectiveness in storing and retrieving responses was assessed.

A comprehensive evaluation of the question-answering system's integration of extracted responses and labels was conducted. The system's overall efficacy was assessed using performance indicators such as response time, answer relevancy, and resource utilization.

H) Generating Quizzes from PDF Documents

The methodology outlines the process of extracting content from PDF documents, generating multiple-choice questions (MCQs) from the content, and creating a quiz dataset. This includes text extraction, filtering, question generation, and structuring the quiz data.

- PDF documents are loaded from a specified directory using the Directory Loader class and PyPDFLoader.
- Text content is extracted from each PDF document.
- The extracted content is split into manageable chunks using the Recursive Character Text Splitter. Chunks are created with an overlap to avoid information loss at chunk boundaries.
- Pre-processing of Extracted Text
- Special characters such as newline characters and bullet points are removed from the extracted text content.
- Cleaned content is prepared for further processing and analysis.

I) MCQ Generator Initialization

- An instance of the mcq_gen class is initialized, which encapsulates the MCQ generation process.
- The mcq_gen class uses pre-trained models, tokenizers, and libraries for generating MCQs.
- For each extracted text chunk, the cleaned content is used as input for MCQ generation.
- The cleaned content is provided to the predict_mcq function, which generates MCQs based on keywords and context.
- The generated MCQs are returned as output, along with other metadata.

International Current Journal of Engineering and Science (ICJES)
Volume 4, Issue 5, pp 12-17, May-2025
ISSN (Online): 2583-7257
Article DOI: https://doi.org/10.47001/ICJES/2025.405003

- The generated MCQs are filtered to remove empty entries or cases where no relevant questions could be generated.
- The filtered list of MCQs is organized for further processing.
- The filtered list of MCQs is used to create a structured quiz dataset.
- Each MCQ is converted into a Quiz object, containing the question statement, options, and correct answer index.
- The complete process is executed by running the run function, passing the directory containing the PDF documents.
- The extracted text chunks are cleaned and processed, and MCQs are generated for each chunk.
- The generated quiz data is returned as the mcq_quiz variable.

V. DISCUSSION

Designing an interactive chatbot and quiz portal for Object-Oriented Programming (OOP) students is a great idea to enhance their learning experience and engagement with the subject matter.

A) Platform Overview

The primary goal of this platform is to provide OOP students with an engaging and effective way to learn and practice the principles of Object-Oriented Programming. The platform will consist of two main components: an interactive chatbot and a quiz portal.

B) Interactive Chatbot

The interactive chatbot will serve as a virtual assistant that students can interact with to clarify concepts, ask questions, and seek guidance on OOP topics. The chatbot should be designed to provide clear and concise explanations, as well as answer common queries that students might have. Some key features of the chatbot include:

Natural Language Processing (NLP): The chatbot should use NLP techniques to understand and process the questions asked by students in a conversational manner.

Concept Explanations: The chatbot should be able to provide explanations for fundamental OOP concepts like classes, objects, inheritance, encapsulation, and abstraction.

Code Examples: The chatbot can generate and explain code examples to help students better understand coding principles.

Interactive Learning: The chatbot can engage students in interactive scenarios to reinforce their understanding of concepts through simulated coding challenges.

C) Quiz Portal

The quiz portal will provide a variety of quizzes related to OOP concepts. Quizzes can be designed to cater to different levels of difficulty, helping students gauge their understanding of the subject matter.

Question Bank: Develop a comprehensive question bank that covers a wide range of OOP topics and difficulty levels.

Multiple Choice and Coding Questions: Include both multiple-choice questions (MCQs) and coding-based questions to assess theoretical knowledge as well as practical coding skills.

Progress Tracking: Implement a progress tracking system that allows students to see their performance over time and identify areas that need improvement.

Immediate Feedback: Provide instant feedback after each quiz attempt, explaining the correct answers and providing references to relevant study materials.

VI. CONCLUSION

In conclusion, the development of an Interactive Chatbot and Quiz Portal for Object Oriented Programming (OOP) students presents a significant advancement in enhancing the learning experience for individuals studying this complex subject. This innovative platform offers a range of benefits that contribute to improved comprehension, engagement, and mastery of OOP principles. Through the Interactive Chatbot, students have access to personalized assistance and guidance, mirroring the experience of having an expert tutor available 24/7. This dynamic tool addresses students' queries in real-time, provides clarifications on challenging concepts, and reinforces their understanding through interactive discussions. As a result, students can overcome obstacles more effectively, fostering a deeper grasp of OOP concepts. Furthermore, the integration of a Quiz Portal complements the learning process by offering an interactive assessment mechanism. Quizzes enable students to apply theoretical knowledge to practical scenarios, enhancing critical thinking and problem-solving skills. The immediate feedback provided upon completing quizzes aids in identifying areas that require further attention, facilitating a targeted and efficient study approach.

International Current Journal of Engineering and Science (ICJES)
Volume 4, Issue 5, pp 12-17, May-2025
ISSN (Online): 2583-7257
Article DOI: https://doi.org/10.47001/ICJES/2025.405003

REFERENCES

[1] Anderson, C. A., & Huttenlocher, D. (2018). Chatbot Applications in Education. International Journal of Artificial Intelligence in Education, 28(1), 16-27.

[2] Martinez, G., & Bannister, E. (2019). The Role of Interactive Quizzes in Programming Education. Journal of Educational Technology & Society, 22(2), 127-138.

[3] Li, Y., & Wang, J. (2020). Customizing Chatbots for Programming Education. Computers in Human Behavior, 102, 164-173.

[4] Johnson, D., Deterding, S., Kuhn, K. A., Staneva, A., Stoyanov, D., & Hides, L. (2017). Gamification for Health and Wellbeing: A Systematic Review of the Literature. Internet Interventions, 9, 1-15.

[5] Chen, J., Song, L., Weng, M., Zhang, Y., & Yang, C. (2018). Designing Chatbots for Educational Purposes: A User-Centered Approach. International Journal of Human-Computer Interaction, 34(6), 511-520.

[6] Hattie, J., & Timperley, H. (2007). The Power of Feedback. Review of Educational Research, 77(1), 81-112.

[7] Jurafsky, D., & Martin, J. H. (2019). Speech and Language Processing (3rd ed.). Pearson.

[8] Acquisti, A. (2009). Nudging Privacy: The Behavioral Economics of Personal Information. IEEE Security & Privacy, 7(6), 82-85.

[9] Tondeur, J., et al. (2017). Measuring TPACK in educational practice: A systematic review. Educational Technology & Society, 20(1), 58-67.

Citation of this Article:

Prabhakaran R.N., & Anirudh Vikas P. (2025). Smart Educational Chatbot with Integrated Quiz Portal for Interactive Student Learning. *International Current Journal of Engineering and Science - ICJES*, 4(5), 12-17. Article DOI: https://doi.org/10.47001/ICJES/2025.405003

****

