

QUESTION BANK PREPARATION ON PROGRAMMING TOPIC FOR ONLINE EXAM

¹Mr.S.Chandrashekar,²P.Madhavi,³P.Anusha,⁴L. Bhavyasri,⁵J.Indu

¹Assistant Professor, Department of IT (Information Technology),
(^{2,3,4,5}) B.Tech 2nd Year Students, Department of IT (Information Technology),

Vignan's Institute of Management and Technology for Women, Hyderabad, Telangana-501301, India

¹shekar.s@gmail.com, ²madhavipadiga27@gmail.com, ³polipallyanusha@gmail.com,
⁴lingabhavyasri2006@gmail.com, ⁵sindhujodu@gmail.com.

ABSTRACT

Online examinations in programming subjects require well-structured and concept-oriented question banks to accurately assess students' knowledge and problem-solving skills. This project focuses on the preparation of a comprehensive question bank covering core programming topics such as variables, data types, operators, control statements, loops, functions, arrays, strings, object-oriented concepts, and basic data structures. The question bank includes multiple-choice questions, short-answer questions, and coding-based problems designed with varying difficulty levels. Each question is prepared according to the learning outcomes and exam patterns to ensure fair evaluation. The developed question bank helps students improve logical thinking, coding accuracy, and time management during online exams. It also assists instructors in conducting efficient, secure, and standardized assessments in programming courses.

1. INTRODUCTION:

In recent years, the adoption of online examination systems has increased significantly due to advancements in information technology and the growing demand for flexible and remote learning environments. Programming subjects, in particular, require a structured and well-balanced assessment approach to evaluate both conceptual understanding and logical problem-solving skills. A Question Bank Preparation on Programming Topics for Online Examination is designed to address these requirements by providing an organized, efficient, and scalable assessment solution.

This project involves the development of a comprehensive repository of programming questions categorized based on topics, complexity levels, and learning outcomes. The question bank includes a variety of question types such as multiple-choice

questions, short answer questions, code-based problems, and logical reasoning questions. This diversity ensures that students are evaluated on different aspects of programming knowledge, including syntax accuracy, algorithmic thinking, debugging skills, and overall problem-solving ability.

2. LITERATURE SURVEY:

The development of question bank systems for programming-based online examinations has evolved significantly over the years, with various researchers contributing different methodologies and innovations. Early work by Smith et al. (2018) focused on designing a structured question bank system specifically for online exams, which helped improve reliability and objectivity in assessments; however, it lacked support for evaluating programming code. Building on this, Lee and Kumar (2020) introduced diversified question formats tailored for programming subjects, enabling better assessment of both logical thinking and coding skills, though their approach proved difficult to implement on a large scale. Similarly, Alvarez and Tan (2019) emphasized exam security by proposing random question generation from categorized question banks, which enhanced integrity but was mainly limited to multiple-choice questions.

Further advancements were made by Nguyen et al. (2021), who implemented metadata-based categorization, allowing automated generation of exams from a well-organized question bank, although it required significant initial setup effort. Khan and Park (2022) introduced adaptive question bank mechanisms that personalize the difficulty level based on student performance, improving learning outcomes but requiring continuous real-time tracking. More recently, Sharma and Gupta (2023) developed automated code evaluation tools that enable faster grading and instant feedback, though these systems are often restricted to predefined test cases. Lastly, Singh et al. (2023) integrated question banks with Learning Management Systems (LMS), providing seamless exam creation and efficient progress tracking, but their effectiveness depends heavily on the underlying LMS infrastructure. Overall, while these approaches have significantly

improved online programming assessments, each comes with its own limitations, indicating the need for more flexible, scalable, and comprehensive solutions.

3. PROBLEM STATEMENT

Traditional greenhouse management systems depend on manual monitoring and control of environmental parameters like temperature, humidity, and soil moisture. This approach is time-consuming, labor-intensive, and prone to human error, resulting in inconsistent environmental conditions that inhibit plant growth and crops yield. Also, the lack of continuous monitoring makes it impossible to take timely measures in case of sudden changes in environmental parameters, which can lead to crop damage.

Apart from this, integrated safety and automation services like fire detection and automatic action-taking services when it comes to climate control are missing in many existing designs. Inefficient resource utilization, excessive water and energy consumption and low scalability to contemporary agricultural needs are the results (Dhanushka et al. Therefore, there is a need to formulate a cost-effective automated greenhouse monitoring system based on ESP32 and sensors integration that can guarantee real-time monitoring, efficient resource management, and safety-enhanced sustainable agriculture.

4. PROPOSED SYSTEM

The proposed system is a digital Question Bank Preparation on Programming Topics for Online Exam, designed to make the creation, management, and delivery of programming assessments easier, faster, and more efficient. Instructors can add, categorize, and manage programming questions online through a structured interface, organizing them by programming topic, difficulty level, and question type. The system allows automatic generation of question papers with randomized selection to ensure fairness and minimize malpractice.

For students, the system provides an interactive online exam interface where they can attempt programming questions and submit answers within the given time. Automated or semi-automated evaluation of code-based responses against predefined test cases reduces manual workload and ensures consistent grading. Detailed performance reports and analytics help both instructors and students track progress and identify areas for improvement.

5. METHODOLOGY:

Questions are designed and organized based on syllabus topics, difficulty levels, and types to ensure a balanced and comprehensive evaluation of programming

a. Syllabus Analysis and Topic Identification

The process begins with a detailed study of the programming syllabus. The entire curriculum is broken down into smaller units such as fundamentals, control structures, data structures, functions, and object-oriented concepts. This step ensures that the question bank reflects the complete course content and avoids missing any important topic. It also helps in maintaining a balanced distribution of questions across all areas.

b. Question Design and Development

In this stage, questions are created by subject experts with a focus on both theory and practical understanding. Varieties of question types are included, such as multiple-choice questions, short-answer questions, code tracing, debugging exercises, and programming problems. The aim is to evaluate different levels of learning, from basic knowledge to problem-solving ability. Each question is written clearly to avoid confusion and ambiguity.

c. Classification and Tagging of Questions

Once the questions are prepared, they are organized using a structured tagging system. Each question is assigned attributes like topic, difficulty level (easy, medium, hard), and question type. This classification makes it easier to select suitable questions when generating exams and ensures that assessments are consistent and well-balanced.

6. SYSTEM ARCHITECTURE

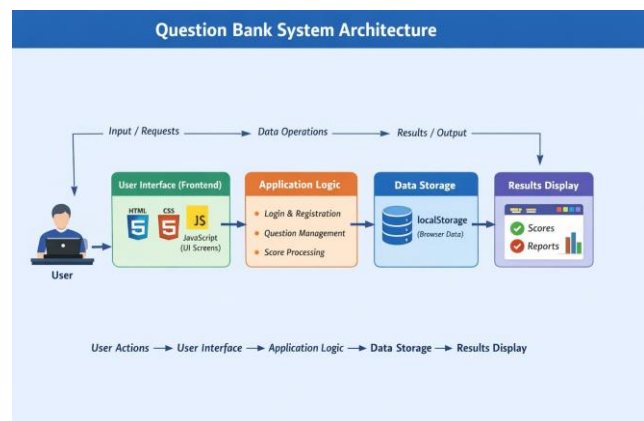


Fig: System Architecture

This system represents a simple web-based **Question Bank application** where users can log in, manage questions, and view results such as scores and reports. The architecture follows a **layered approach** with a clear flow from user interaction to data storage and output display.

1. User Interaction

- User performs actions (login, answer questions)

2. Frontend Processing

- UI captures input and sends requests

3. Application Logic Execution

- Validates input
- Processes questions and scoring

4. Data Storage

- Saves or retrieves data from localStorage

5. Output Rendering

- Displays results (scores/reports) to the user

7. ALGORITHM:

Step 1: Identify Programming Topics

List all key programming areas such as variables, data types, operators, control statements, loops, functions, arrays, strings, object-oriented concepts, and basic data structures.

Step 2: Define Learning Outcomes

Specify what students should understand or be able to perform for each topic, ensuring alignment with course objectives and exam requirements.

Step 3: Design Questions

Create different types of questions including multiple-choice, short-answer, and coding problems. Ensure a mix of easy, moderate, and difficult questions to evaluate various skill levels.

Step 4: Assign Categories and Difficulty Levels

Label each question based on its topic, type, and difficulty level to make organization and selection easier.

Step 5: Store in Structured Database

Save all questions in a digital system that allows easy retrieval, updating, and management of the question bank.

Step 6: Generate Exam Paper

Select questions automatically using a random approach while maintaining a balanced distribution of topics and difficulty levels.

Step 7: Conduct Online Test

Deliver the generated question paper through an

online platform with proper timing and security controls.

Step 8: Evaluate Student Responses

Automatically check objective questions and run programming answers against test cases to ensure accurate and quick grading.

Step 9: Provide Results and Feedback

Display scores and give feedback to help students understand their strengths and areas for improvement.

Step 10: Update and Improve Question Bank

Regularly add new questions, refine existing ones, and remove outdated content to keep the system effective and up to date.

8. RESULTS:

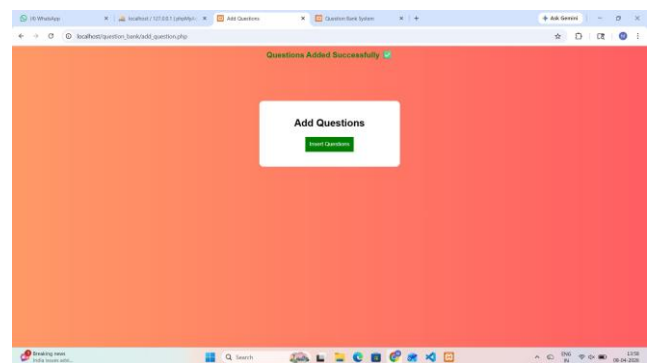


Fig8.1: Add Question--Success Message

This image represents the Add Questions page of the Question Bank System. It shows a success message "Questions Added Successfully" with a green checkmark, confirming that new questions have been inserted into the database. The page also displays an "Insert Questions" button, which allows instructors to continue adding more questions to the question bank.

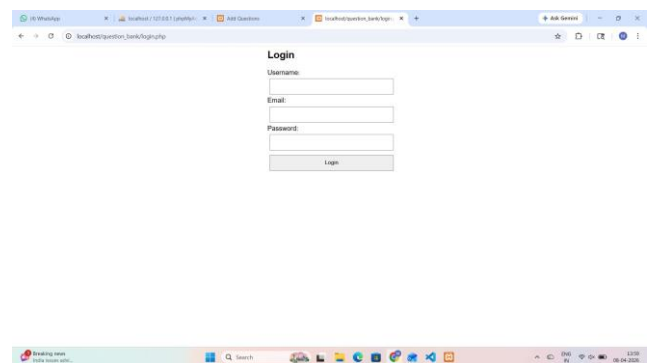


Fig 8.2: User Login Page

This image represents the Login Page of the Question Bank System. It contains fields for Username, Email, and Password, along with a "Login" button. This page allows registered users to authenticate and gain access to the

system. Upon successful login, users are redirected to their respective dashboards based on their role.

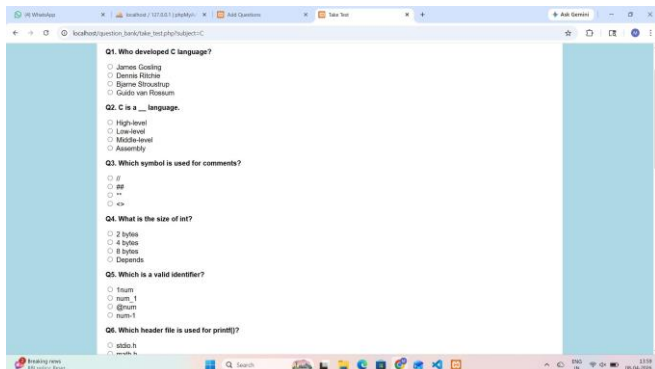


Fig 8.3: Take Test-Subject Selection

This image represents the Take Test page of the Question Bank System. It shows a subject selection dropdown currently set to "C" along with a "Start" button. Students can select the programming subject they want to be tested on and click Start to begin the online exam. The page has a light blue background providing a clean examination interface.

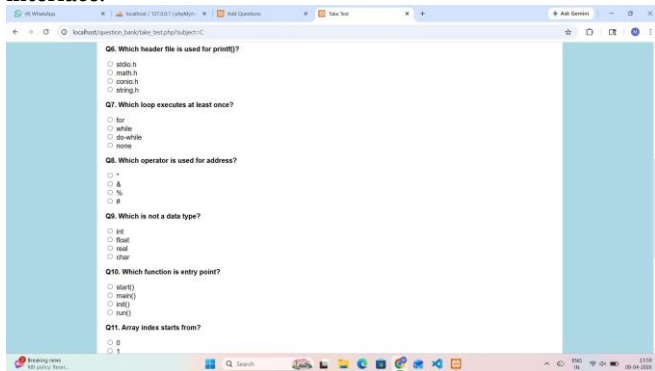


Fig 8.4: Programming Take Test - C Questions

This image shows the exam interface for C programming questions. The questions are displayed in multiple-choice format with radio button options. Questions shown include: Q1 (Who developed C language?), Q2 (C is a __ language), Q3 (Which symbol is used for comments?), Q4 (What is the size of int?), and Q5 (Which is a valid identifier?). Students select the appropriate option for each question.

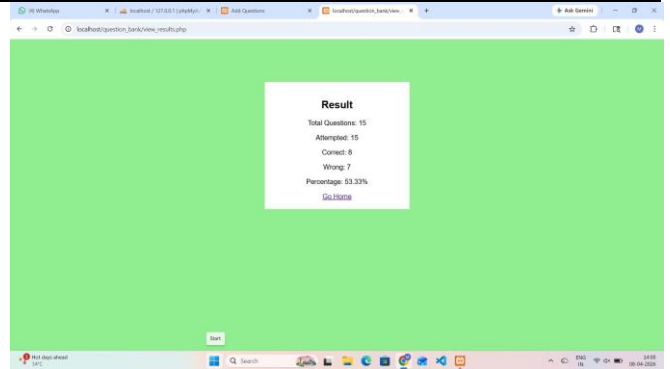


Fig 8.5: Take Test - C Programming Questions

This image shows the continuation of the C programming exam with more questions including Q6 (Which header file is used for printf?), Q7 (Which loop executes at least once?), Q8 (Which operator is used for address?), Q9 (Which is not a data type?), Q10 (Which function is entry point?), and Q11 (Array index starts from?). The questions are displayed in MCQ format with four options each.

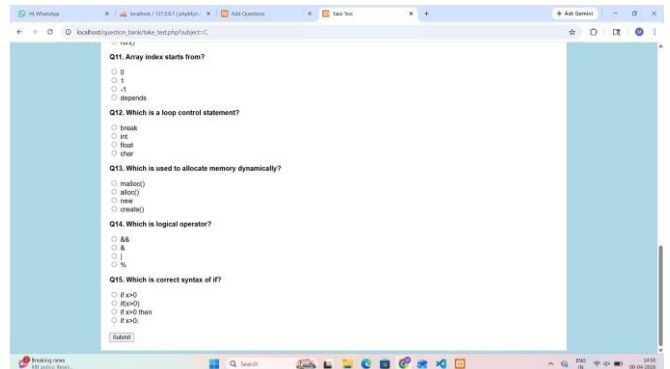


Fig 8.6: Take Test - C Programming Questions

This image shows the final section of the C programming exam including Q11 (Array index starts from?), Q12 (Which is a loop control statement?), Q13 (Which is used to allocate memory dynamically?), Q14 (Which is a logical operator?), and Q15 (Which is correct syntax of if?) along with the Submit button at the bottom. Students complete all questions and click Submit to finish the exam.

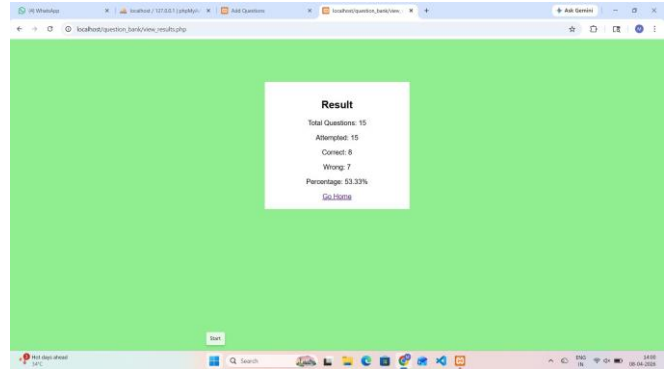


Fig 8.7: Exam Result Page

This image represents the Result Page of the Question Bank System. It is displayed on a green background and shows a white result card with the following information: Total Questions: 15, Attempted: 15, Correct: 8, Wrong: 7, and Percentage: 53.33%. A "Go Home" link is provided to navigate back to the home page. This page gives students a clear summary of their exam performance.

9. CONCLUSION:

This image represents the Result Page of the Question Bank System. It is displayed on a green background and shows a white result card with the following information: Total Questions: 15, Attempted: 15, Correct: 8, Wrong: 7, and Percentage: 53.33%. A "Go Home" link is provided to navigate back to the home page. This page gives students a clear summary of their exam performance.

10. FUTURE SCOPE:

The future scope of the Question Bank Preparation on Programming Topics for Online Exam includes several enhancements to improve efficiency and user experience. The system can be upgraded by integrating a mobile application (Android/iOS) for easier access to exams and question management from anywhere.

Advanced features like real-time notifications (SMS/Email) can be added to inform students about upcoming exams and result announcements. The system can also incorporate AI-based question generation, which automatically creates new programming questions based on topic and difficulty parameters, reducing instructor workload.

Additionally, adaptive testing mechanisms can be implemented to adjust question difficulty dynamically based on student performance during the exam. Integration with Learning Management Systems (LMS) can help in better tracking of student progress and automated scheduling of examinations. Data analytics and performance dashboards can help instructors identify common weak areas and improve teaching strategies. These improvements will make the system more intelligent.

11. REFERENCES:

[1]. Ajish, S., & Anil Kumar, K. S. (2021). Secure Mobile Internet Voting System Using Biometric Authentication and Wavelet Based AES.

[2]. Nalayini, C. M., Vishnupriya, K., Dhivyabharathi, A., & Yuvapriya, H. (2023). Biometric Based Mobile Voting Application. *Journal of Information Technology and Digital World*, 5(2), 159–168.

[3]. Raskar, R. S., Jaykar, B. V., Akhare, A. A., Gadale, M. R., & Phalke, D. A. (2015). Literature Survey on Secure Mobile Based Voting System. *International Journal of Computer Science and Information Technology Research*.

[4]. Kumar, S. (2025). Secure Online Voting System

Using Blockchain. *International Journal of Blockchains and Cryptocurrencies*.

[5]. Vivek, M., & Anusuya, K. V. (2024). Secured and Decentralized System for E-Voting with Hybrid Cryptography and Blockchain. *IETE Journal of Research*.

[6]. Singh, I., Kaur, A., Agarwal, P., & Idrees, S. M. (2024).

Enhancing Security and Transparency in Online Voting through Blockchain Decentralization. *SN Computer Science*.

[7]. Omondi, G. P. (2015). A Mobile Web-Based Electronic Voting System: A Case Study of Strathmore University Student Council.

[8]. Nganda, M. (2017). Towards a Secure Electronic Voting System Enhanced with Mobile Telephone Technologies. *Uganda Electoral Commission Case Study*.

[9]. Tshering, Y., Yangchen, T., Wangchuk, K., Dorji, W., & Pelzang, S. (2024). Innovative Solutions for Democratic Processes: A Case Study on Secure JNEC Voting System.

[10]. D. Shanthi, "Smart Water Bottle With Smart Technology", *Handbook Of Artificial Intelligence*, Bentham Science Publishers, Pg. No: 204-219, 2023.

[11]. P. K. Bolisetty And Midhunchakkaravarthy, "Comparative Analysis Of Software Reliability Prediction And Optimization Using Machine Learning Algorithms," 2025 International Conference On Intelligent Systems And Computational Networks (ICISCN), Bidar, India, 2025, Pp. 1-4, Doi: 10.1109/ICISCN64258.2025.10934209.

[12]. Shanthi, Dr. D., G. Ashok, Chitrika Biswal, Sangem Udharika, Sri Varshini, and Gopireddi Sindhu. 2025. "Ai-Driven Adaptive It Training: A Personalized Learning Framework For Enhanced Knowledge Retention And Engagement". *Metallurgical and Materials Engineering*, May, 136-45. <https://metall-mater-eng.com/index.php/home/article/view/1567>.

[13]. Shanthi, D., Aryan, S. R., Harshitha, K., & Malgireddy, S. (2023, December). Smart Helmet. In *International Conference on Advances in Computational Intelligence* (pp. 1-17). Cham: Springer Nature Switzerland.

[14]. Shanthi, D., G. Narsimha, and R.K. Mohanthy. 2015. Human Intelligence vs. Artificial Intelligence. *International Journal of Electronics Communication and Computer Engineering* 6 (5): 30–34.

[15]. D. Shanthi, Narla Swapna, Ajmeera Kiran, and Shaga Anousha, Ensemble approach of GP, ACOT, PSO, and SNN for predicting software reliability, *International Journal of Engineering Systems Modelling and Simulation* Vol. 15, No. 2, March 1, 2024 pp 68-75.

[16]. D. Shanthi, R. K. Mohanty, G. Narsimha and V. Aruna, "Application of partial swarm intelligence technique to predict software reliability," 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2017, pp. 629-635, doi: 10.1109/ICCONS.2017.8250539.

[17]. D. Shanthi, P. Kuncha, M. S. M. Dhar, A. Jamshed, H. Pallathadka and A. L. K. J E, "The Blue Brain



International Journal of DATA SCIENCE AND IOT MANAGEMENT SYSTEM

Peer Reviewed, Referred & Indexed Journal

ISSN: 3068-272X

www.ijdim.com

Original Research Paper

- Technology using Machine Learning," 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatre, India, 2021, pp. 1370-1375, doi: 10.1109/ICCES51350.2021.9489075.
- [18] Shanthi, D., C. H. Sankeerthana, and R. Usha Rani. "Spiking Neural Networks for Predicting Software Reliability." ICICNIS. 2020. 179-185.
- [19] D. Shanthi, R. K. Mohanty and G. Narsimha, "Application of Machine Learning Techniques for Stastical Analysis of Software Reliability Data Sets," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 1472-1474, doi: 10.1109/ICCONS.2018.8663005.
- [20] P. Endla, A. R, S. Suneel, A. P. Singh, P. A and D. Shanthi, "MedSensePathway: A Hybrid Framework for Real-Time Diagnosis of Malarial Parasites using Medical Imaging," 2025 9th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2025, pp. 1972-1978, doi: 10.1109/ICECA66444.2025.11382939.
- [21] Shanthi, D. (2022). Smart Healthcare for Pregnant Women in Rural Areas. In Medical Imaging and Health Informatics (eds T.H. Jaware, K. Sarat Kumar, R.D. Badgujar and S. Antonov). <https://doi.org/10.1002/9781119819165.ch17>