



C. V. Raman
Global University
ODISHA BHUBANESWAR INDIA

**A Report Submitted as a Part Of Experiential
Learning On :**

C++

Topic:

"JobPrep++ "

Presented By :

| NAME | REGD. NUMBER |
|----------------------------|---------------------|
| RAJNEESH RANJAN | CL20250106018920102 |
| SMRUTI PRATISRUTI MAHARANA | CL2025010601899755 |
| ROHIT KUMAR RAM | CL2025010601873139 |
| SUBHAMITA DASH | CL2025010601911955 |
| ARYAZEET DEBEPRIYA | CL2025010601924677 |

Department of Computer Science Engineering

TABLE OF CONTENTS

| SI. No. | TOPIC | PAGE NO. |
|----------------|--------------------|-----------------|
| 1. | ABSTRACT | 1 |
| 2. | INTRODUCTION | 2 |
| 3. | OBJECTIVE | 3-4 |
| 4. | SIGNIFICANCE | 5 |
| 5. | LIBRARIES IMPORTED | 6 |
| 7. | SOURCE CODE | 7-14 |
| 9. | OUTPUT | 15-17 |
| 10. | ADVANTAGES | 18 |
| 11. | DISADVANTAGES | 19 |
| 12. | FUTURE WORK | 20-21 |
| 13. | CONCLUSION | 22 |
| 14. | REFERENCE | 23 |

ABSTRACT

JobPrep++ is a quiz-based application developed in C++ to assist students in their placement preparation. It covers five key subjects—**C, C++, Java, Python, and MERN (MongoDB, Express.js, React, Node.js)**—which are essential for technical interviews. Each subject includes three quizzes, and every quiz consists of **20 carefully selected questions, totaling 300 questions**. These questions are designed to test **fundamental concepts, problem-solving skills, and coding knowledge, helping users strengthen their technical proficiency**.

The application features a secure user authentication system, requiring users to register with a username and password. Once logged in, users can attempt quizzes and track their progress. The system records scores, enabling users to analyze their past performances upon re-login. This allows for continuous learning and improvement by identifying strong and weak areas in different subjects.

Developed entirely in C++, JobPrep++ ensures a fast, efficient, and lightweight experience, making it accessible across different platforms with minimal system requirements. The structured and interactive approach helps students systematically prepare for placement exams and technical interviews.

With its **vast question bank, secure login system, and performance-tracking capabilities**, JobPrep++ serves as a valuable tool for aspirants looking to enhance their technical knowledge in both programming languages and full-stack development. It provides a structured learning path to boost confidence and improve problem-solving abilities for placement success.

INTRODUCTION

JobPrep++ is a quiz-based application designed to help students enhance their technical skills for placement preparation. Developed entirely in C++, it covers five essential subjects: C, C++, Java, Python, and MERN (MongoDB, Express.js, React, Node.js). These subjects are fundamental for technical interviews and competitive coding, making JobPrep++ a valuable tool for students aiming to strengthen their knowledge. The application includes a total of 300 questions, distributed across three quizzes per subject, with each quiz containing 20 questions. These questions are carefully curated to test problem-solving abilities and fundamental concepts.

One of the standout features of JobPrep++ is its secure user authentication system, which provides a personalized learning experience. Users must create an account with a unique username and password, allowing them to log in securely, access quizzes, and save their progress. The system records quiz scores, enabling users to track their performance over time. This feature helps learners analyze their strengths and weaknesses, allowing them to focus on areas that need improvement and refine their skills accordingly.

Built using C++, JobPrep++ ensures a fast, efficient, and lightweight user experience. The application is designed to run smoothly on different platforms without requiring high system resources. Its structured approach encourages continuous learning, while the user-friendly interface makes navigation simple and intuitive.

With its extensive question bank, secure login functionality, and performance-tracking features, JobPrep++ serves as an effective tool for students preparing for technical interviews and placement exams. It offers a structured and interactive learning path to enhance problem-solving skills, boost confidence, and improve overall technical competency, ensuring better results in technical assessments.

OBJECTIVE

The primary objective of JobPrep++ is to provide students with a structured and efficient platform for technical placement preparation. By offering subject-specific quizzes in C, C++, Java, Python, and MERN (MongoDB, Express.js, React, Node.js), the application aims to strengthen fundamental concepts and problem-solving skills required for technical interviews and competitive exams.

Key objectives of JobPrep++ include:

- **Enhancing Technical Knowledge :** Providing 300 carefully curated questions across five subjects to help students reinforce their understanding of core programming concepts and frameworks.
- **Structured Learning Approach :** Organizing quizzes systematically, with three quizzes per subject, each containing 20 questions, ensuring comprehensive coverage of key topics.
- **User Authentication & Personalization :** Implementing a secure login system with username and password authentication to store user progress and provide a personalized experience.
- **Performance Tracking :** Recording and saving quiz scores to help users analyze their progress over time, identify strengths and weaknesses, and focus on areas that need improvement.
- **Lightweight and Efficient Platform :** Utilizing C++ to develop a fast, optimized, and resource-efficient application that runs smoothly on different platforms.
- **Boosting Placement Readiness :** Offering a structured way for students to practice technical questions, improve accuracy and speed, and gain confidence before appearing for placement exams.
- **Encouraging Self-Assessment :** Enabling students to repeatedly attempt quizzes and compare their scores over time to track progress and gauge their level of preparation.

- **Developing Time Management Skills** : Helping users practice solving questions under a time constraint to simulate real placement test conditions and improve efficiency.
- **Providing a Practical Learning Experience** : Ensuring hands-on exposure to technical subjects by focusing on real-world questions and coding-related problems.
- **Reducing Dependency on External Study Materials** : Offering a comprehensive in-app resource that eliminates the need for multiple preparation sources.
- **Ensuring Secure Data Storage** : Safeguarding user credentials and progress records to maintain privacy and prevent unauthorized access.
- **Scalability for Future Expansion** : Allowing easy expansion by adding new subjects, quizzes, and advanced features to keep up with evolving placement trends.

Through these objectives, JobPrep++ aims to serve as a reliable and effective learning tool, helping students enhance their technical skills and succeed in their career aspirations.

SIGNIFICANCE

The significance of JobPrep++ lies in its ability to provide a structured, efficient, and engaging platform for students preparing for technical placements. By focusing on key programming languages and frameworks such as C, C++, Java, Python, and MERN (MongoDB, Express.js, React, Node.js), the application ensures comprehensive coverage of essential topics required for technical interviews.

- **Enhanced Learning Experience:** The application offers a systematic approach to learning by dividing each subject into three quizzes, ensuring students get well-rounded practice on fundamental concepts. The structured format helps in step-by-step learning, making technical subjects easier to grasp.
- **Personalized User Progress Tracking:** With a secure user authentication system, JobPrep++ allows users to track their quiz performance over time. By storing quiz scores, users can analyze their progress, identify weak areas, and focus on improving their skills. This feature ensures a continuous and effective learning process.
- **Placement-Oriented Preparation:** The curated set of 300 questions is designed to simulate real placement tests. By attempting quizzes under time constraints, users develop time management skills, improving their speed and accuracy in problem-solving—an essential requirement for competitive exams and technical interviews.
- **Lightweight and Efficient:** Built entirely in C++, JobPrep++ provides a fast, efficient, and resource-friendly environment that runs smoothly on different systems without requiring high-end hardware. This makes it accessible to a wide range of users.
- **Self-Assessment and Confidence Building:** By enabling users to attempt quizzes multiple times and compare previous scores, the application promotes self-assessment. It encourages students to track their growth, gain confidence, and enter placement exams better prepared.

LIBRARIES IMPORTED

- **#include <iostream>:** Used for standard input and output operations like cin, cout, cerr, and clog. Essential for basic console interaction in C++.
- **#include <fstream>:** Provides file handling capabilities using ifstream for input and ofstream for output. Supports reading, writing, and checking file states
- **#include <string>:** Enables working with std::string for efficient text handling and manipulation. Includes various functions like substr(), find(), append(), and erase()
- **#include <vector>:** Provides a dynamic array with std::vector, allowing flexible resizing and element access. Supports useful methods like push_back(), size(), and at() for safe indexing
- **#include <cstdlib>:** Contains general-purpose functions including random number generation and memory management. Includes conversion functions like atoi(), atof(), and system-related operations
- **#include <ctime>:** Provides time-related functions such as time(), clock(), and difftime() for timing operations. Commonly used to seed random number generation using srand(time(0))
- **#include <limits>:** Defines numeric limits for data types through std::numeric_limits<T>. Useful for handling input validation, buffer clearing, and checking data type ranges

SOURCE CODE

```
1 #include <iostream>
2 #include <fstream>
3 #include <vector>
4 #include <sstream>
5 #include <chrono>
6 #include <ctime>
7 #include <map>
8
9 using namespace std;
10
11 struct Question
12 {
13     string question;
14     vector<string> options;
15     char correctAnswer;
16 };
17
18 struct Quiz
19 {
20     string quizName;
21     vector<Question> questions;
22 };
23
24 map<string, string> users; // Stores username-password pairs
25 vector<Quiz> quizzes;
26
27 void loadUsers()
28 {
29     ifstream file("users.txt");
30     string username, password;
31     while (file >> username >> password)
32     {
33         users[username] = password;
34     }
35     file.close();
36 }
37
38 void saveUser(const string &username, const string &password)
39 {
40     ofstream file("users.txt", ios::app);
41     file << username << " " << password << "\n";
42     users[username] = password;
43     file.close();
44 }
```

SOURCE CODE

```
47 vector<Quiz> loadQuizzes() {
48     vector<Quiz> loadedQuizzes;
49
50     map<string, vector<vector<Question>>> questionBank = {
51         {"C++", {
52             { // Quiz 1
53                 {"1. What is C++?", {"A programming language", "A database", "An OS", "A hardware"}, 'A'},
54                 {"2. Which keyword defines a class in C++?", {"function", "class", "struct", "namespace"}, 'B'},
55                 {"3. What is the extension of a C++ file?", {"txt", "cpp", "java", "py"}, 'B'},
56                 {"4. Who developed C++?", {"Dennis Ritchie", "Bjarne Stroustrup", "James Gosling", "Guido van Rossum"}, 'B'},
57                 {"5. Which of these is not a C++ keyword?", {"new", "delete", "malloc", "class"}, 'C'},
58                 {"6. What is the output of 'cout << 5 + 10;', {"5", "10", "15", "Error"}, 'C'},
59                 {"7. Which operator is used to allocate memory dynamically?", {"malloc", "alloc", "new", "create"}, 'C'},
60                 {"8. What is the extension of a C++ source file?", {"c", "cpp", "java", "py"}, 'B'},
61                 {"9. Which of these is an access specifier in C++?", {"private", "public", "protected", "All of the above"}, 'D'},
62                 {"10. What does 'OOP' stand for?", {"Object Oriented Programming", "Online Object Protocol", "Ordered Object Processing", "None of the above"}, 'A'},
63                 {"11. Which symbol is used for comments in C++?", {"//", "--", "##", "***"}, 'A'},
64                 {"12. Which loop runs at least once in C++?", {"for", "while", "do-while", "None"}, 'C'},
65                 {"13. Which header file is required for input-output operations?", {"<iostream>", "<stdio.h>", "<string>", "<fstream>"}, 'A'},
66                 {"14. Which of these is used for exception handling?", {"throw", "catch", "try", "All of the above"}, 'D'},
67                 {"15. Which function is the entry point in a C++ program?", {"start()", "init()", "main()", "execute()"}, 'C'},
68                 {"16. What is the default return type of main()?", {"void", "int", "float", "char"}, 'B'},
69                 {"17. Which operator is used to access members of a class?", {"->", ".(dot)", "::", "/"}, 'B'},
70                 {"18. Which keyword is used to define a constant in C++?", {"final", "const", "static", "define"}, 'B'},
71                 {"19. Which of these is a feature of OOP?", {"Encapsulation", "Inheritance", "Polymorphism", "All of the above"}, 'D'},
72                 {"20. Which function deallocates memory allocated by 'new'?", {"delete", "free", "dealloc", "remove"}, 'A'}
73             },
74             { // Quiz 2
75                 {"1. What is a pointer?", {"A reference", "A variable", "A function", "An integer"}, 'A'},
76                 {"2. Which operator is used for dynamic memory allocation?", {"malloc", "new", "alloc", "dynamic"}, 'B'},
77                 {"3. Which header file is needed for input-output operations?", {"<stdio.h>", "<iostream>", "<stdlib.h>", "<conio.h>"}, 'B'},
78                 {"4. What is the size of 'int' on most 64-bit systems?", {"2 bytes", "4 bytes", "8 bytes", "16 bytes"}, 'B'},
79                 {"5. What is the purpose of the 'this' pointer?", {"Refers to the current object", "Refers to a global object", "Refers to the previous object", "None"}, 'A'},
80                 {"6. Which of these is not a type of constructor in C++?", {"Default", "Parameterized", "Copy", "Friend"}, 'D'},
81                 {"7. What is function overloading?", {"Multiple functions with the same name but different parameters", "Defining a function inside another function"}, 'A'},
82                 {"8. Which operator cannot be overloaded?", {"+", "=", ".(dot)", "[]"}, 'C'},
83                 {"9. Which of these keywords is used for templates?", {"class", "template", "typename", "Both B and C"}, 'D'},
84                 {"10. What is a virtual function?", {"A function that is dynamically bound at runtime", "A function that cannot be used", "A function that runs in a separate thread"}, 'A'},
85                 {"11. Which class can access private members of another class?", {"Inherited class", "Friend class", "Abstract class", "None"}, 'B'},
86                 {"12. What is the default access specifier for a class member?", {"public", "private", "protected", "static"}, 'B'},
87                 {"13. What is the difference between struct and class?", {"No difference", "struct has private members by default", "class has private members by default"}, 'A'},
88                 {"14. What is an abstract class?", {"A class with at least one pure virtual function", "A class without a constructor", "A class that cannot be instantiated"}, 'A'},
89                 {"15. What is the advantage of using references over pointers?", {"References cannot be NULL", "References require less memory", "References are faster"}, 'A'},
90                 {"16. How do you declare a pure virtual function?", {"virtual void f() {}", "virtual void f() = 0;", "void f() = 0;", "void f(){}"}, 'B'},
91                 {"17. What does the keyword 'mutable' do?", {"Allows modification of a member even in a const object", "Prevents modification", "Converts a variable to a pointer"}, 'A'},
92                 {"18. Which STL container does not allow duplicate elements?", {"vector", "map", "multiset", "deque"}, 'B'},
93                 {"19. Which operator is used to declare a smart pointer?", {"->", ".(dot)", "&", "*"}, 'D'},
94                 {"20. Which feature allows multiple functions to have the same name but different implementations?", {"Inheritance", "Encapsulation", "Polymorphism"}, 'A'}
95             },
96             { // Quiz 3
97                 {"1. Which concept allows function overloading?", {"Polymorphism", "Encapsulation", "Inheritance", "Abstraction"}, 'A'},
98                 {"2. What is 'cin' used for?", {"Output", "Input", "File handling", "Memory allocation"}, 'B'},
99                 {"3. Which keyword is used for a constant variable?", {"final", "static", "const", "define"}, 'C'},
100                {"4. Which of the following statements about move semantics is true?", {"Move constructors prevent unnecessary copies", "Move constructors always copy objects", "Move constructors always move objects", "Move constructors are slower than copy constructors"}, 'A'},
101                {"5. What is the main purpose of the 'explicit' keyword in C++?", {"Prevents implicit type conversion", "Forces inline expansion", "Declares a variable as const", "None"}, 'A'},
102                {"6. Which C++11 feature is used to ensure a class cannot be copied?", {"delete keyword", "static keyword", "constexpr keyword", "volatile keyword"}, 'A'},
103                {"7. What is the output of the following code? \n\n"
104                    "int x = 5;\n"
105                    "auto lambda = [x]() mutable { x++; return x; };\n"
106                    "cout << lambda() << endl;"}, {"5", "6", "Compilation error", "Undefined behavior"}, 'B'},
107                {"8. Which type of inheritance can cause the [Diamond problem] in C++?", {"Single", "Multiple", "Multilevel", "Hybrid"}, 'B'},
108                {"9. What is the advantage of emplace_back() over push_back() in vectors?", {"Constructs objects directly in-place", "Faster than push_back()", "Uses deep copy", "None"}, 'A'},
109                {"10. Which of these is NOT a valid C++11 memory management feature?", {"std::unique_ptr", "std::shared_ptr", "std::weak_ptr", "std::gabage_collector"}, 'A'},
110                {"11. Which exception will be thrown if [new] fails to allocate memory?", {"std::bad_alloc", "std::overflow_error", "std::runtime_error", "std::invalid_argument"}, 'A'},
111                {"12. Which of the following is the correct syntax to prevent object slicing?", {"Declare a virtual destructor", "Use the final keyword", "Declare a pure virtual function", "None"}, 'A'},
112                {"13. Which of these operators cannot be overloaded?", {"[], >, ?: (ternary), ()"}, 'C'},
113                {"14. Which of the following is true about [volatile] keyword?", {"Prevents compiler optimizations", "Ensures atomic operations", "Used to declare static variables", "None"}, 'A'},
114                {"15. Which of the following is NOT a valid way to prevent memory leaks?", {"Use smart pointers", "Use manual delete calls", "Implement RAI", "Use garbage collection"}, 'A'},
115                {"16. Which standard library feature allows iteration over a container with range-based for loops?", {"begin() and end()", "rbegin() and rend()", "cbegin() and cend()"}, 'A'},
116                {"17. What will the following C++ code output? \n\n"
117                    "struct Base { virtual void show() { cout << 'Base'; } };\n"
118                    "struct Derived : Base { void show() override { cout << 'Derived'; } };\n"
119                    "Base* obj = new Derived();\n"
120                    "obj->show();"}, {"Base", "Derived", "Compilation Error", "Runtime Error"}, 'B'},
121                {"18. Which type of polymorphism allows method calls to be determined at runtime?", {"Static", "compile-time", "Runtime (dynamic)", "Function overloading"}, 'A'},
122                {"19. Which of these statements is false regarding RAI (Resource Acquisition Is Initialization)?", {"It ensures proper resource management", "It eliminates memory leaks", "None"}, 'A'},
123                {"20. Which of the following is an example of perfect forwarding in C++?", {"Using std::forward()", "Using copy constructors", "Using move constructors", "Using swap()"}, 'A'}
124            }
125        }
126    }
127 }
```

SOURCE CODE

```
126 {"Java", {
127     { // Quiz 1 - Basics of Java
128         {"1. What is Java?", {"A database", "A programming language", "An operating system", "A web browser"}, 'B'},
129         {"2. Who developed Java?", {"Dennis Ritchie", "Bjarne Stroustrup", "James Gosling", "Guido van Rossum"}, 'C'},
130         {"3. What is the extension of a Java source file?", {"java", ".class", ".cpp", ".py"}, 'A'},
131         {"4. Which keyword is used to define a class in Java?", {"struct", "class", "function", "define"}, 'B'},
132         {"5. Which of these is not a Java keyword?", {"new", "delete", "final", "static"}, 'B'},
133         {"6. What is the default value of an uninitialized int variable in Java?", {"0", "null", "garbage value", "None"}, 'A'},
134         {"7. What is the output of 'System.out.println(5 + 10);'?", {"5", "10", "15", "Error"}, 'C'},
135         {"8. Which keyword is used to create an object in Java?", {"new", "malloc", "alloc", "create"}, 'A'},
136         {"9. Which of these is a valid method declaration in Java?", {"int method();", "method int();", "void method[]();", "declare method();"}, 'A'},
137         {"10. What does JVM stand for?", {"Java Virtual Machine", "Java Variable Manager", "Java Version Method", "None of the above"}, 'A'},
138         {"11. What is the correct way to start a single-line comment in Java?", {"//", "--", "#", "*"}, 'A'},
139         {"12. Which of these loops runs at least once?", {"for", "while", "do-while", "None"}, 'C'},
140         {"13. Which access specifier allows access within the same package?", {"public", "private", "protected", "default"}, 'D'},
141         {"14. What is the base class of all Java classes?", {"Object", "Main", "Base", "Parent"}, 'A'},
142         {"15. What is the output of 'System.out.println(10/3);'?", {"3.33", "3", "3.0", "Error"}, 'B'},
143         {"16. Which of these is not a primitive data type in Java?", {"int", "float", "string", "char"}, 'C'},
144         {"17. What does 'final' keyword do in Java?", {"Prevents method overriding", "Prevents variable modification", "Prevents class inheritance", "All of the above"}, 'A'},
145         {"18. How do you take user input in Java?", {"Scanner", "InputReader", "System.in.read()", "cin>>"}, 'A'},
146         {"19. Which method is called when an object is destroyed?", {"finalize()", "destructor()", "end()", "delete()"}, 'A'},
147         {"20. Which package is imported by default in every Java program?", {"java.lang", "java.util", "java.io", "java.net"}, 'A'}
148     },
149     { // Quiz 2 - Intermediate Java
150         {"1. What is method overloading?", {"Multiple methods with same name but different parameters", "Defining a method inside another method", "Using a method inside another method", "Overriding a method"}, 'A'},
151         {"2. What is the default access specifier of a class?", {"private", "protected", "public", "default"}, 'D'},
152         {"3. Which data structure does a HashMap use internally?", {"Array", "LinkedList", "Hashtable", "Tree"}, 'C'},
153         {"4. What is the purpose of the 'this' keyword?", {"Refers to the current object", "Refers to a global object", "Refers to the previous object", "None of the above"}, 'A'},
154         {"5. Which class is the superclass for all exceptions?", {"Throwable", "Exception", "RuntimeException", "Error"}, 'A'},
155         {"6. What is the output of 'Math.round(2.6)'?", {"2", "3", "2.6", "Error"}, 'B'},
156         {"7. What is a constructor in Java?", {"A method that initializes an object", "A method that destroys an object", "A static method", "A method that returns a value"}, 'A'},
157         {"8. Which keyword is used to inherit a class?", {"implements", "extends", "inherits", "super"}, 'B'},
158         {"9. Which Java collection allows storing unique elements in sorted order?", {"ArrayList", "HashSet", "TreeSet", "LinkedList"}, 'C'},
159         {"10. Which of these statements is true about abstract classes?", {"They cannot have constructors", "They cannot have abstract methods", "They can have both constructors and abstract methods", "None of the above"}, 'A'},
160         {"11. What does the 'super' keyword do?", {"Calls the parent class constructor", "Calls the child class constructor", "Creates a new object", "None of the above"}, 'A'},
161         {"12. Which of these methods is used to check if a string contains another string?", {"contains()", "find()", "includes()", "match()"}, 'A'},
162         {"13. What will happen if we call 'main()' method inside itself?", {"Infinite recursion", "Compilation error", "Runtime error", "Nothing happens"}, 'A'},
163         {"14. Which interface is used for handling threads in Java?", {"Threadable", "Runnable", "Multithread", "Process"}, 'B'},
164         {"15. What is the default value of a boolean variable?", {"true", "false", "null", "0"}, 'B'},
165         {"16. What is the use of the 'instanceof' keyword?", {"Checks if an object belongs to a class", "Declares an instance variable", "Creates a new object", "None of the above"}, 'A'},
166         {"17. What is the return type of the hashCode() method?", {"int", "float", "String", "Object"}, 'A'},
167         {"18. Which of the following is not an OOP principle?", {"Encapsulation", "Inheritance", "Polymorphism", "Compilation"}, 'D'},
168         {"19. What does 'static' keyword do?", {"Makes a variable constant", "Makes a method belong to the class instead of an object", "Prevents method overriding", "None of the above"}, 'A'},
169         {"20. What is the purpose of 'finally' block?", {"It executes after 'try' and 'catch' blocks", "It is used to catch exceptions", "It is used to throw exceptions", "None of the above"}, 'A'}
170     },
171     { // Quiz 3 - Advanced Java
172         {"1. Which Java memory area stores objects created using 'new'?", {"Stack", "Heap", "Method Area", "Registers"}, 'B'},
173         {"2. What does the 'volatile' keyword ensure?", {"Method synchronization", "Variable visibility across threads", "Prevents deadlocks", "Makes a variable thread-safe"}, 'A'},
174         {"3. Which thread state occurs when a thread is waiting for a monitor lock?", {"BLOCKED", "WAITING", "TIMED_WAITING", "RUNNABLE"}, 'A'},
175         {"4. What is the primary difference between 'wait()' and 'sleep()'?", {"'wait()' releases the lock, 'sleep()' doesn't", "sleep() releases the lock, wait() releases the lock", "Both release the lock", "None of the above"}, 'A'},
176         {"5. What is the function of the 'ForkJoinPool' in Java?", {"Manages multithreading efficiently", "Divides tasks recursively for parallel processing", "Creates multiple threads", "None of the above"}, 'A'},
177         {"6. Which of these is an immutable class in Java?", {"String", "StringBuilder", "HashMap", "ArrayList"}, 'A'},
178         {"7. What does 'CompletableFuture' provide in Java?", {"Asynchronous computation", "Synchronous execution", "Reflection capabilities", "Garbage collection"}, 'A'},
179         {"8. Which feature was introduced in Java 8 for functional programming?", {"Streams", "Generics", "Synchronization", "Reflection"}, 'A'},
180         {"9. What is the output of 'System.out.println(10 == 10.0);'?", {"true", "false", "Compilation error", "Runtime error"}, 'A'},
181         {"10. How do you prevent object creation from a class?", {"Declare the class final", "Make the constructor private", "Use synchronized keyword", "Implement the Comparable interface"}, 'A'},
182         {"11. Which design pattern is used in 'java.util.Observer'?", {"Singleton", "Factory", "Observer", "Adapter"}, 'C'},
183         {"12. Which functional interface does 'forEach()' use in Java Streams?", {"Runnable", "Consumer", "Supplier", "Function"}, 'B'},
184         {"13. How can we create a thread in Java?", {"Implement Runnable", "Extend Thread", "Use ExecutorService", "All of the above"}, 'D'},
185         {"14. What is the return type of 'map()' in Java Streams?", {"Stream<T>", "List<T>", "void", "Collection<T>"}, 'A'},
186         {"15. What does the 'Optional' class in Java help with?", {"Avoiding NullPointerException", "Performance optimization", "Memory management", "Type safety"}, 'A'},
187         {"16. What is the difference between HashMap and ConcurrentHashMap?", {"ConcurrentHashMap allows thread-safe operations", "HashMap is faster", "HashMap is thread-safe", "Both are the same"}, 'B'},
188         {"17. What does 'javac' command do?", {"Runs Java programs", "Compiles Java source code", "Starts JVM", "Debugs code"}, 'B'},
189         {"18. What is the difference between deep copy and shallow copy?", {"Deep copy clones all objects", "Shallow copy copies references", "Both A & B", "None of the above"}, 'A'},
190         {"19. What is a memory leak in Java?", {"Unused objects are not garbage collected", "Excessive recursion", "Too many synchronized threads", "Use of final variables"}, 'A'},
191         {"20. Which of these is a thread-safe collection?", {"ArrayList", "HashMap", "CopyOnWriteArrayList", "LinkedList"}, 'C'}
192     }
193 }
```

SOURCE CODE

```
194 {"Python": {  
195     // *Python Quiz 1 (Basic)*  
196     ["1. What is Python?", {"A snake", " A programming language", " A database", " A hardware"}, 'B'],  
197     ["2. Which keyword is used to define a function in Python?", {"function", " def", " class", " lambda"}, 'B'],  
198     ["3. What is the extension of a Python file?", {"txt", "cpp", "java", "py"}, 'D'],  
199     ["4. Who developed Python?", {"Dennis Ritchie", " Bjarne Stroustrup", " James Gosling", " Guido van Rossum"}, 'D'],  
200     ["5. Which of these is a valid Python variable name?", {"var", " _var", " var-name", " class"}, 'B'],  
201     ["6. What is the output of 'print(5 + 10)'?", {"5", " 10", " 15", " Error"}, 'C'],  
202     ["7. Which function is used to get user input in Python?", {"scanf()", " gets()", " input()", " cin"}, 'C'],  
203     ["8. Which of these is a valid way to comment in Python?", {"//", "--", "#", "***"}, 'C'],  
204     ["9. Which data type is used to store multiple values in one variable?", {"list", " int", " float", " bool"}, 'A'],  
205     ["10. What will 'print(type(3.14))' output?", {"int", " float", " double", " string"}, 'B'],  
206     ["11. What is the result of '10 // 3'??", {"3.33", " 3", " 3.0", " 4"}, 'B'],  
207     ["12. What is the default return type of a function that has no return statement?", {"void", " None", " int", " null"}, 'B'],  
208     ["13. Which keyword is used for loops in Python?", {"repeat", " for", " loop", " iterate"}, 'B'],  
209     ["14. What is the output of 'bool([])'?", {"True", " False", " None", " Error"}, 'B'],  
210     ["15. What is the correct syntax to declare a dictionary?", {"{}", "[ ]", " dict"}, 'D'],  
211     ["16. How do you open a file in read mode?", {"open('file.txt', 'w')", "open('file.txt', 'r')", "open('file.txt', 'a')", "open('file.txt', 'x')"}, 'B'],  
212     ["17. Which of the following is immutable?", {"list", " dictionary", " set", " tuple"}, 'D'],  
213     ["18. What is the output of 'print(2 ** 3)'?", {"5", " 6", " 8", " 9"}, 'C'],  
214     ["19. Which of these functions converts a string to an integer?", {"str()", " float()", " int()", " chr()"}, 'C'],  
215     ["20. What is the default value of an uninitialized variable in Python?", {"None", " 0", " Null", " Undefined"}, 'A']  
},  
1. // *Python Quiz 2 (Intermediate)*  
2. ["1. Which of the following is a valid way to define a function?", {"function myFunc():", " def myFunc():", " define myFunc():", " func myFunc():"}, 'B'],  
3. ["2. How do you write a lambda function?", {"lambda x: x+2", "lambda(x): x+2", "lambda x { return x+2 }", "def lambda x: x+2"}, 'A'],  
4. ["3. What is the difference between a list and a tuple?", {"Lists are mutable, tuples are immutable", "Lists store numbers, tuples store strings", "Tuples are immutable, lists are mutable"}, 'B'],  
5. ["4. What does the 'finally' block do in Python exceptions?", {"Runs if an error occurs", "Runs after the 'try' block", "Skips the 'except' block", "Is part of the finally block"}, 'C'],  
6. ["5. What will 'print(type({}))' output?", {"list", " dict", " set", " tuple"}, 'B'],  
7. ["6. What is a shallow copy?", {"A reference to the original object", "A deep copy of an object", "Copies only mutable objects", "Copies all nested objects"}, 'B'],  
8. ["7. How do you open a file in append mode?", {"open('file.txt', 'r')", "open('file.txt', 'a')", "open('file.txt', 'w')", "open('file.txt', 'x')"}, 'B'],  
9. ["8. What does 'zip()' do in Python?", {"Merges two iterables", "Compresses a file", "Converts a list to a tuple", "None of the above"}, 'A'],  
10. ["9. Which module is used for regular expressions?", {"re", " regex", " reg", " search"}, 'A'],  
11. ["10. What does the 'super()' function do?", {"Calls a parent class method", "Calls a child class method", "Declares a superclass", "None of the above"}, 'B'],  
12. ["11. Which function is used to remove an element from a set?", {"remove()", "pop()", "discard()"}, 'D'],  
13. ["12. What is a generator in Python?", {"A function that returns an iterator", "A function that executes instantly", "A function that runs only once", "A function that generates values"}, 'B'],  
14. ["13. How do you define a class in Python?", {"class MyClass():", "define MyClass():", "struct MyClass()", "object MyClass():"}, 'A'],  
15. ["14. What is the purpose of '_init_' in a class?", {"It is a destructor", "It initializes objects", "It is a static method", "It deletes objects"}, 'B'],  
16. ["15. Which keyword is used to create an abstract class?", {"interface", "abstract", "class"}, "None of the above"], 'D'],  
17. ["16. What does 'enumerate()' do?", {"Returns a tuple", "Adds a counter to an iterable", "Sorts a list", "None of the above"}, 'B'],  
18. ["17. What is the purpose of 'is' in Python?", {"Compares values", "Compares memory locations", "Assigns values", "Creates a list"}, 'B'],  
19. ["18. Which of these sorts a list in descending order?", {"sorted(list, reverse=True)", "list.sort(reverse=True)", "list[::-1]", "All of the above"}, 'B'],  
20. ["19. What is the difference between 'is' and '=='?", {"No difference", "'is' compares values, '==' compares memory", "'is' compares memory, '==' compares objects", "None of the above"}, 'B'],  
21. ["20. Which function removes duplicates from a list?", {"set(list)", "list.remove_duplicates()", "unique(list)", "list.clear_duplicates()"}, 'A']  
},  
1. // *Python Quiz 3 (Advanced)*  
2. ["1. Which decorator is used to define a static method?", {"@staticmethod", " @classmethod", " @property", " @private"}, 'A'],  
3. ["2. What does 'deepcopy()' do?", {"Copies an object deeply", "Copies an object shallowly", "Copies only mutable parts", "None of the above"}, 'A'],  
4. ["3. What does 'yield' do in a function?", {"Returns a value", "Stops execution", "Creates a generator", "None of the above"}, 'C'],  
5. ["4. Which built-in function is used to check if an object is iterable?", {"iter()", "next()", "isinstance()", "hasattr()"}, 'A'],  
6. ["5. How do you declare a metaclass?", {"class Meta(type):", "def MetaClass():", "struct Meta():", "class Meta()"}, 'A'],  
7. ["6. What does '_call_' do?", {"Calls an object", "Creates an instance", "Calls a function", "None of the above"}, 'A'],  
8. ["7. What is monkey patching?", {"Modifying code at runtime", "Overwriting a class", "Debugging", "None of the above"}, 'A'],  
9. ["8. Which function in NumPy creates an array?", {"np.array()", "np.create()", "np.make()", "np.new()"}, 'A'],  
10. ["9. How do you serialize an object?", {"pickle.dump()", "json.dump()", "Both A and B", "None of the above"}, 'C'],  
11. ["10. What will be the output of the following code? \n\n    def func(lst=()):\n        lst.append(1)\n        return lst\n    print(func())\n\n    print(func()[1], [1, 1], [1, 1][1], [1, 1][1, 1]), 'B'],  
12. ["11. What does the 'nonlocal' keyword do in Python?", {"Declares a global variable", "Declares a local variable", "Allows modification of a variable in the outer scope", "None of the above"}, 'C'],  
13. ["12. Which of the following statements about Python generators is true?", {"Generators store all values in memory", "Generators use 'yield' instead of 'return'", "Generators are faster than loops", "Generators are slower than loops"}, 'B'],  
14. ["13. What will be the output of the following code? \n\n    def f():\n        return lambda x: x * 2\n    g = f()\n    print(g(5)), '10", "5", "None", "Error"}, 'A'],  
15. ["14. What will be the output of the following code? \n\n    class A:\n        def __init__(self):\n            print('A')\n    class B(A):\n        def __init__(self):\n            print('B')\n        super().__init__()\n    B(), 'A B", "B A", "A", "B"}, 'B'],  
16. ["15. Which of the following is true about Python's memory management?", {"Uses manual garbage collection", "Uses reference counting and a garbage collector", "None of the above"}, 'B'],  
17. ["16. What is the correct syntax for defining a metaclass in Python?", {"class Meta(type): pass", "class Meta(MetaClass=type): pass", "class Meta(): pass"}, 'B'],  
18. ["17. What will be the output of the following code? \n\n    def test():\n        try:\n            return 1\n        finally:\n            return 2\n    print(test()), '1", "2", "None", "Error"}, 'B'],  
19. ["18. What does 'GIL' stand for in Python?", {"Global Integer Lock", "Global Interpreter Lock", "General Iteration Loop", "General Inheritance Library"}, 'B'],  
20. ["19. What does the 'inspect' module in Python allow you to do?", {"Modify bytecode", "Analyze code structure at runtime", "Manage virtual environments", "None of the above"}, 'B'],  
21. ["20. What will be the output of the following code? \n\n    from collections import Counter\n    c = Counter('aabbc')\n    print(c.most_common(1)), '[(a, 3)]", "[('b', 2)]", "[('a', 3), ('b', 2)]", "[('c', 1)]", 'A']  
},  
}
```

SOURCE CODE

```
288 // MERN, {
289   // Quiz 1 - MERN Stack Basics
290   ("1. What does MERN stand for?", {" MongoDB, Express, React, Node", " MySQL, Express, Redux, Node", " MongoDB, Ember, React, Node", " MongoDB, Express, React", 'A'},
291   ("2. What type of database is MongoDB?", {" Relational", " NoSQL", " Graph-based", " Key-Value Store"}, 'B'),
292   ("3. Which command is used to install Express.js?", {" npm install express", " node install express", " install express.js", " npm add express"}, 'A'),
293   ("4. Which of these is used to handle routes in Express.js?", {" Router", " Path", " Redirect", " Handler"}, 'A'),
294   ("5. What is the purpose of React in the MERN stack?", {" Backend framework", " Frontend UI library", " Database management", " Server-side scripting"}, 'C'),
295   ("6. Which hook is used for state management in React?", {" useState", " useEffect", " useContext", " useRef"}, 'A'),
296   ("7. Which function in Express is used to handle HTTP requests?", {" app.listen()", " app.use()", " app.get()", " app.route()"}, 'C'),
297   ("8. Which package is used to connect MongoDB with Node.js?", {" mongoose", " mongo-client", " mongodb", " mongoDB-connect"}, 'A'),
298   ("9. What is the purpose of the virtual DOM in React?", {" Improves performance", " Provides backend logic", " Replaces Redux", " Stores real-time data"}, 'B'),
299   ("10. Which of these is a valid state management solution in React?", {" Redux", " useState", " context API", " All of the above"}, 'D'),
300   ("11. How do you start a Node.js server?", {" node server.js", " start node", " node -server"}, 'A'),
301   ("12. Which method is used to send a JSON response in Express?", {" res.send()", " res.json()", " res.write()", " res.end()"}, 'B'),
302   ("13. What does JSX stand for in React?", {" JavaScript XML", " JSON Express", " JavaScript Express", " Java Syntax Extension"}, 'A'),
303   ("14. Which of the following is a correct way to define a functional component in React?", {" function MyComponent() {}", " const MyComponent = () => {}"}, 'B'),
304   ("15. How do you pass data from a parent component to a child component in React?", {" Props", " Redux", " State", " Context API"}, 'A'),
305   ("16. Which command is used to install dependencies listed in package.json?", {" npm install", " npm run install", " node install", " npm init"}, 'A'),
306   ("17. What is the default port for a React development server?", {" 5000", " 3000", " 8000", " 8080"}, 'B'),
307   ("18. How do you define a route parameter in Express?", {" /user/:id", " /user{id}", " /user&{id}"}, 'A'),
308   ("19. Which HTTP method is used to update an existing resource?", {" GET", " POST", " PUT", " DELETE"}, 'C'),
309   ("20. Which of these middleware functions is used for handling JSON requests in Express?", {" body-parser", " json-parser", " express.json()"}, " All of the above")
310 },
311 // Quiz 2 - Intermediate MERN
312   ("1. What is the primary purpose of Mongoose in a MERN stack application?", {" To manage React state", " To simplify MongoDB interactions", " To handle Express routing", " To validate user input"}, 'B'),
313   ("2. How do you define a schema in Mongoose?", {" new Schema()", " new Model()", " new Document()", " new Collection()"}, 'A'),
314   ("3. What is the purpose of useEffect in React?", {" Handle component rendering", " Handle side effects", " Manage global state", " Control event listeners"}, 'B'),
315   ("4. Which method is used to fetch data from an API in React?", {" axios.get()", " fetch()", " userFetch()", " Both A & B"}, 'D'),
316   ("5. What is the difference between useState and useReducer?", {" useState is for complex state logic", " useState is for local state only", " useReducer is for complex state logic", " both are identical"}, 'A'),
317   ("6. Which middleware function is used for handling JSON requests in Express?", {" express.json()", " body-parser.json()", " json-parser()", " Both A & B"}, 'B'),
318   ("7. What is the purpose of useMemo in React?", {" To optimize performance", " To update the state", " To replace useState", " To define component props"}, 'A'),
319   ("8. How do you create a protected route in React?", {" Using Redux", " Using React Router and authentication logic", " Using useState", " Using Express"}, 'B'),
320   ("9. Which of the following is NOT a valid lifecycle method in a React class component?", {" componentDidMount", " componentWillUnmount", " useEffect", " componentDidUpdate"}, 'C'),
321   ("10. What is JWT used for in MERN applications?", {" Frontend state management", " Backend logging", " User authentication", " API caching"}, 'C'),
322   ("11. Which method is used to hash passwords in Node.js?", {" bcrypt.hash()", " hashPassword()", " encryptPassword()", " crypto.hash()"}, 'A'),
323   ("12. What is the correct way to handle errors in Express middleware?", {" app.use((err, req, res, next) => {...})", " app.get(err, req, res, next) {...}"}, 'B'),
324   ("13. What is the purpose of Redux in a React application?", {" Manages component lifecycle", " Handles backend API calls", " Manages global state", " Optimizes component rendering"}, 'A'),
325   ("14. How do you create a virtual field in Mongoose?", {" Using virtual in schema", " Using pre-save hook", " Using a separate collection"}, " Not possible"),
326   ("15. What is the role of next() in Express middleware?", {" Terminates the request", " Passes control to the next middleware", " Redirects to a new route"}, 'B'),
327   ("16. What is the difference between sessionStorage and localStorage?", {" sessionStorage persists even after closing browser", " localStorage is only available in the browser"}, 'A'),
328   ("17. What does ObjectId represent in MongoDB?", {" A Unique ID for each document", " The ID of a collection", " A schema type", " A primary key in SQL"}, 'B'),
329   ("18. How do you optimize performance in a React application?", {" Using React.memo", " Avoiding unnecessary re-renders", " Using useCallback and useMemo"}, 'A'),
330   ("19. What is the purpose of CORS in Express?", {" Blocks frontend requests", " Allows cross-origin requests", " Prevents database access", " Encrypts API responses"}, 'B'),
331   ("20. Which tool is commonly used for debugging Express applications?", {" React DevTools", " Mongoose Debugger", " Morgan", " Redux DevTools"}, 'C')
332 },
333 // Quiz 3 - Advanced MERN
334   ("1. What is server-side rendering (SSR) in React?", {" Rendering components in the browser", " Rendering components on the server before sending them to the client"}, 'B'),
335   ("2. Which React framework is best for SSR?", {" Create React App", " Next.js", " Redux", " Express.js"}, 'B'),
336   ("3. How can you improve MongoDB query performance?", {" Using indexes", " Using embedded documents", " Avoiding unnecessary fields", " All of the above"}, 'D'),
337   ("4. What is the purpose of lazy() in React?", {" Code splitting and lazy loading", " Fetching data from an API", " Managing state in Redux", " Optimizing component rendering"}, 'B'),
338   ("5. How do you handle race conditions in async operations?", {" Using async/await", " Using Promises", " Using mutex locks", " All of the above"}, 'D'),
339   ("6. What is the difference between cluster and worker_threads in Node.js?", {" Cluster runs multiple instances, worker_threads runs threads inside a process"}, 'B'),
340   ("7. What is the purpose of debouncing in React?", {" Improves API call efficiency", " Reduces unnecessary renders", " Optimizes input handling", " All of the above"}, 'D'),
341   ("8. How does WebSocket differ from HTTP?", {" WebSocket is persistent, HTTP is stateless", " WebSocket only works on localhost", " HTTP is faster than WebSocket", " Both are the same"}, 'A'),
342   ("9. How can you prevent SQL injection in MongoDB?", {" Using parameterized queries", " Using Mongoose validation", " Sanitizing user input", " All of the above"}, 'D'),
343   ("10. What is helmet used for in Express.js?", {" Secure HTTP headers", " Authentication", " Database security", " Logging"}, 'A'),
344   ("11. What is the purpose of next.config.js in Next.js?", {" Customizing the Next.js build process", " Managing React state", " Configuring MongoDB", " Defining environment variables"}, 'B'),
345   ("12. What is the purpose of getServerSideProps in Next.js?", {" Fetch data at build time", " Fetch data on each request for SSR", " Fetch data only on the client"}, 'B'),
346   ("13. What is the purpose of React Profiler?", {" Debugging Express API", " Measuring component rendering performance", " Managing Redux state", " Handling component hydration"}, 'B'),
347   ("14. How can you handle authentication in a MERN application?", {" JWT tokens", " OAuth", " Sessions & Cookies", " All of the above"}, 'D'),
348   ("15. Which database scaling technique improves MongoDB read performance?", {" Replication", " Sharding", " Indexing", " All of the above"}, 'D'),
349   ("16. What is the advantage of Service Workers in a MERN app?", {" Improves offline capabilities", " Enhances security", " Reduces API calls", " All of the above"}, 'B'),
350   ("17. Which package is used for GraphQL API development in a MERN stack?", {" express-graphql", " mongoose", " next-auth", " redux-graphql"}, 'A'),
351   ("18. How do you prevent memory leaks in a Node.js server?", {" Using event listeners correctly", " Avoiding global variables", " Managing database connections", " Using clearInterval"}, 'B'),
352   ("19. What is the primary advantage of using Redis in a MERN application?", {" Faster data retrieval", " Secure authentication", " Improved component reuse", " Both A and C"}, 'D'),
353   ("20. What is the purpose of useTransition in React 18?", {" Improves performance for UI state updates", " Handles routing in Next.js", " Fetches data from a remote source"}, 'B')
354 },
355 }
356 },
357 for (const auto &subject : questionBank) {
358   for (int i = 0; i < 3; i++) { // Creating Quiz 1, 2, 3 for each subject
359     Quiz quiz;
360     quiz.quizName = subject.first + " Quiz " + to_string(i + 1);
361     quiz.questions = subject.second[i]; // Assign different sets of questions
362     loadedQuizzes.push_back(quiz);
363   }
364 }
365 }
366 return loadedQuizzes;
367 }
```

SOURCE CODE

```
370 // Get today's date as a string
371 string getTodayDate() {
372     auto now = chrono::system_clock::to_time_t(chrono::system_clock::now());
373     tm *ltm = localtime(&now);
374     stringstream dateStream;
375     dateStream << 1900 + ltm->tm_year << "-" << 1 + ltm->tm_mon << "-" << ltm->tm_mday;
376     return dateStream.str();
377 }
378
379 // Get current time as a string (HH:MM:SS)
380 string getCurrentTime() {
381     auto now = chrono::system_clock::to_time_t(chrono::system_clock::now());
382     tm *ltm = localtime(&now);
383     stringstream timeStream;
384     timeStream << ltm->tm_hour << ":" << ltm->tm_min << ":" << ltm->tm_sec;
385     return timeStream.str();
386 }
387
388 // Count previous attempts for a specific quiz by a user
389 int getAttemptCount(const string &username, const string &quizName) {
390     ifstream file("Leaderboard.txt");
391     string line, user, quiz, date;
392     int attempt, score, count = 0;
393
394     while (file >> user >> quiz >> date >> attempt >> score) {
395         if (user == username && quiz == quizName) {
396             count = attempt; // Keep track of last attempt number
397         }
398     }
399     file.close();
400     return count + 1; // Next attempt number
401 }
402
403 void playQuiz(const Quiz &quiz, const string &username) {
404     int score = 0;
405     cout << "\nStarting quiz: " << quiz.quizName << "\n";
406     for (const auto &q : quiz.questions) {
407         cout << "\n" << q.question << "\n";
408         for (int i = 0; i < 4; i++) {
409             cout << char('A' + i) << ". " << q.options[i] << "\n";
410         }
411
412         cout << "Your answer (A/B/C/D): ";
413         char answer;
414         auto start = chrono::steady_clock::now();
415         cin >> answer;
416         auto end = chrono::steady_clock::now();
417
418         chrono::duration<double> elapsed = end - start;
419         if (elapsed.count() > 30.0) {
420             cout << "Time's up! Moving to the next question.\n";
421             continue;
422         }
423
424         if (toupper(answer) == q.correctAnswer) {
425             cout << "Correct!\n";
426             score++;
427         } else {
428             cout << "Wrong! The correct answer was " << q.correctAnswer << "\n";
429         }
430     }
431 }
```

SOURCE CODE

```
432     string date = getTodayDate();
433     int attemptNumber = getAttemptCount(username, quiz.quizName);
434
435     ofstream leaderboard("leaderboard.txt", ios::app);
436     leaderboard << username << " " << quiz.quizName << " " << date << " " << attemptNumber << " " << score << "\n";
437     leaderboard.close();
438
439     cout << "\nQuiz Over! Your score: " << score << "\n";
440 }
441
442 // Function to display past scores with date & time
443 void showLeaderboard(const string &username) {
444     ifstream file("leaderboard.txt");
445     string name, quizName, date;
446     int attempt, score;
447     bool found = false;
448
449     cout << "\n===== Your Past Quiz Scores =====\n";
450     while (file >> name >> quizName >> date >> attempt >> score) {
451         if (name == username) {
452             cout << quizName << " | Attempt: " << attempt
453             << " | Date: " << date
454             << " | Score: " << score << "\n";
455             found = true;
456         }
457     }
458     file.close();
459
460     if (!found) {
461         cout << "No past quiz attempts found.\n";
462     }
463 }
464
465 void mainMenu();
466
467 void userMenu(const string &username) {
468     while (true) {
469         cout << "\n1. Select Programming Language\n2. View Past Scores\n3. Logout\nchoose an option: ";
470         int choice;
471         cin >> choice;
472         if (choice == 1) {
473             vector<string> subjects = {"C++", "Java", "Python", "MERN", "C"};
474             cout << "Choose a programming language:\n";
475             for (size_t i = 0; i < subjects.size(); i++) {
476                 cout << i + 1 << ". " << subjects[i] << "\n";
477             }
478             int subjectChoice;
479             cin >> subjectChoice;
480
481             if (subjectChoice > 0 && subjectChoice <= subjects.size()) {
482                 string selectedSubject = subjects[subjectChoice - 1];
483                 cout << "Choose a quiz:\n";
484                 for (int i = 1; i <= 3; i++) {
485                     cout << i << ". " << selectedSubject << " Quiz " << i << "\n";
486                 }
487                 int quizChoice;
488                 cin >> quizChoice;
489
490                 if (quizChoice > 0 && quizChoice <= 3) {
491                     playQuiz(quizzes[(subjectChoice - 1) * 3 + (quizChoice - 1)], username);
492                 }
493             }
494         } else if (choice == 2) {
495             showLeaderboard(username);
496         } else if (choice == 3) {
497             cout << "Logging out...\n";
498             mainMenu();
499             return;
500         }
501     }
502 }
```

SOURCE CODE

```
501     |     else {
502     |         cout << "Invalid choice! Try again.\n";
503     |
504 }
505 }
506
507 void mainMenu() {
508     loadUsers();
509     quizzes = loadQuizzes();
510
511     cout << "\n1. Login\n2. Signup\n3. Exit\nChoose an option: ";
512     int choice;
513     cin >> choice;
514
515     string username, password;
516     if (choice == 1) {
517         cout << "Enter Username: ";
518         cin >> username;
519         cout << "Enter Password: ";
520         cin >> password;
521         if (users.find(username) != users.end() && users[username] == password) {
522             cout << "Login Successful!\n";
523             userMenu(username);
524         } else {
525             cout << "Invalid Credentials! Try Again.\n";
526             mainMenu();
527         }
528     }
529     else if (choice == 2) {
530         cout << "Choose a Username: ";
531         cin >> username;
532         cout << "Choose a Password: ";
533         cin >> password;
534         saveUser(username, password);
535         cout << "Signup Successful! You can now log in.\n";
536         mainMenu();
537     } else if (choice == 3) {
538         cout << "Exiting...\n";
539         exit(0);
540     } else {
541         cout << "Invalid choice! Try again.\n";
542         mainMenu();
543     }
544 }
545
546 int main() {
547     mainMenu();
548     return 0;
549 }
```

OUTPUT

Main Menu

- 1. Login
- 2. Signup
- 3. Exit

Choose an option: 2

Signup

Choose a Username: Rohit 9101@rohit
Choose a Password: 123456
Signup Successful! You can now log in.

Main Menu (After Signup)

- 1. Login
- 2. Signup
- 3. Exit

Choose an option: 1

Login

Enter Username: Rohit 9101@rohit
Enter Password: 123456
Login Successful!

User Menu

- 1. Select Programming Language
- 2. View Past Scores
- 3. Logout

Choose an option: 1

OUTPUT

Select Programming Language

Choose a programming language:

- 1. C++
- 2. Java
- 3. Python
- 4. MERN
- 5. C

Choose an option: 1

Select Quiz

Choose a quiz:

- 1. C++ Quiz 1
- 2. C++ Quiz 2
- 3. C++ Quiz 3

Choose an option: 1

Quiz Questions

Starting quiz: C++ Quiz 1

1. What is C++?

- A. A programming language
- B. A database
- C. An OS
- D. A hardware

Your answer (A/B/C/D): A

Correct!

2. Which keyword defines a class in C++?

- A. function
- B. class
- C. struct
- D. namespace

Your answer (A/B/C/D): B

Correct!

...

OUTPUT

User Menu (After Quiz)

- 1. Select Programming Language
- 2. View Past Scores
- 3. Logout

Choose an option: 2

Name of Users :

Rajneesh neesh_1111
Smruti smr_1234
Subhasmita Subha@1111
Rohit 9101@rohit
Aryajeet arya8909

Marks of each quiz for all users in Quiz 1 with date and time :

Subhasmita C++ Quiz 1 2025-3-13 1 19
Smruti C++ Quiz 1 2025-3-13 1 19
Rajneesh C++ Quiz 1 2025-3-13 1 18
Rohit C++ Quiz 1 2025-3-13 1 20
Aryajeet C++ Quiz 1 2025-3-13 1 18

ADVANTAGES

1. **Structured Preparation:** JobPrep++ provides a well-organized curriculum that includes technical and non-technical resources, ensuring a comprehensive approach to job readiness. It helps users systematically cover key topics such as coding, aptitude, and soft skills.
2. **Enhanced Problem-Solving Skills:** The platform integrates coding challenges, logical reasoning problems, and case studies, allowing users to develop strong problem-solving abilities. Regular practice on real-world problems improves analytical thinking and decision-making skills.
3. **AI-Driven Personalized Learning:** JobPrep++ adapts to individual performance using AI-based assessments, offering customized study plans. This feature helps users focus on weak areas while optimizing preparation time for maximum efficiency.
4. **Mock Interviews & Resume Guidance:** Users can participate in mock interviews conducted by industry experts to improve their confidence and communication skills. Additionally, the platform provides resume-building tips and feedback to help candidates create strong job applications.
5. **Real-Time Feedback & Progress Tracking:** The system continuously tracks a user's performance, offering instant feedback on coding tests, quizzes, and interview simulations. This allows users to measure their progress and work on areas that need improvement.
6. **Company-Specific Preparation:** JobPrep++ offers insights into the hiring patterns and frequently asked questions of top recruiters. By practicing company-specific problems, candidates can gain an edge in technical and HR interviews.
7. **Access to Industry-Relevant Insights:** The platform provides access to real-world interview experiences, industry trends, and expert advice. This helps users stay updated on the latest hiring processes and expectations of recruiters.

DISADVANTAGES

- 1. Dependency on Internet Connectivity:** Since JobPrep++ is an online platform, users require a stable internet connection to access learning materials, practice tests, and live sessions. Poor connectivity can disrupt the learning experience.
- 2. Lack of Personalized Human Interaction:** While AI-driven learning offers personalized recommendations, it cannot fully replace human mentorship. Some users may prefer direct interaction with mentors for better understanding and guidance.
- 3. Overwhelming Content for Beginners:** The vast amount of study materials, coding problems, and mock tests can be overwhelming for beginners. Without proper guidance, users might struggle to prioritize their learning path.
- 4. Limited Hands-on Project Experience:** While JobPrep++ focuses on coding and interview preparation, it may not provide extensive hands-on project work, which is crucial for real-world job readiness in software development.
- 5. Not Suitable for All Job Roles:** The platform primarily focuses on technical job preparation. Those seeking non-technical roles may find limited resources, making it less beneficial for candidates from other domains.
- 6. Competitive Pressure & Stress:** The leaderboard, timed assessments, and competitive nature of the platform can sometimes lead to unnecessary stress and anxiety, affecting the confidence of some users.

FUTURE WORK

To enhance JobPrep++ and make it a more powerful career preparation platform, several key future improvements can be introduced:

- ❖ **Enhanced AI-Powered Personalized Learning:** Implementing advanced AI models to provide highly customized study plans based on individual strengths, weaknesses, and career aspirations.
- ❖ **Live Mentorship & Career Guidance:** Introducing one-on-one mentorship sessions with industry experts, HR professionals, and recruiters to offer resume feedback, career advice, and interview preparation strategies.
- ❖ **Hands-on Project Integration:** Incorporating real-world projects and case studies into the platform, allowing users to gain practical experience and build a strong portfolio.
- ❖ **Support for Non-Technical Job Roles:** Expanding content to include preparation resources for non-technical roles such as business analysis, UI/UX design, product management, and finance-related careers.
- ❖ **Gamification & Reward System:** Enhancing user engagement by introducing a structured reward system with badges, leaderboards, and incentives for consistent learning and participation.
- ❖ **AI-Powered Resume Builder:** Implementing an AI-driven tool that provides instant resume analysis, suggests improvements, and tailors resumes based on job descriptions.
- ❖ **Mock Interview & Voice Analysis:** Developing AI-powered mock interview sessions with real-time voice analysis to assess confidence, clarity, and communication skills.

- ❖ **Industry-Specific Preparation Tracks:** Creating specialized learning paths for different industries, such as IT, healthcare, finance, and marketing, to provide job-specific preparation.
- ❖ **Integration with Job Portals & Networking Platforms:** Connecting users with job opportunities by integrating JobPrep++ with platforms like LinkedIn, GitHub, and Indeed.
- ❖ **Multilingual Support:** Expanding the platform to support multiple languages, making job preparation accessible to users from diverse linguistic backgrounds.
- ❖ **Soft Skills & Personality Development Modules:** Adding courses on leadership, teamwork, problem-solving, and emotional intelligence to improve overall professional readiness.
- ❖ **Advanced Data Analytics for Performance Tracking:** Providing users with detailed insights on their learning progress, strengths, and areas needing improvement through data-driven analytics.
- ❖ **Cybersecurity Awareness & Digital Literacy Training:** Including courses on online security, data privacy, and ethical hacking to prepare users for modern workplace challenges.

CONCLUSION

JobPrep++ serves as a dynamic and innovative job preparation platform designed to equip users with the necessary skills and confidence for career success. By integrating AI-driven personalized learning, real-world projects, and interactive assessments, it provides a structured approach to mastering job-relevant skills. The platform tailors study plans based on individual strengths and weaknesses, ensuring a focused and effective learning experience.

Beyond technical skills, JobPrep++ offers mentorship programs, mock interviews, and resume-building tools to help users navigate the job market with confidence. These features provide real-world exposure, bridging the gap between theoretical knowledge and practical application. Industry-specific learning tracks further enhance its effectiveness, making it relevant for diverse career paths.

As technology evolves, JobPrep++ continues to innovate by incorporating AI-driven analytics, multilingual support, and cybersecurity training. These enhancements aim to make learning more adaptive and accessible to a global audience. Additionally, expanding networking opportunities and employer partnerships will strengthen the platform's role in career development.

With its holistic approach, JobPrep++ is set to revolutionize job preparation, making it more efficient, personalized, and impactful. By continuously improving its features and adapting to industry trends, it ensures that job seekers remain competitive and well-prepared. As the job market becomes more demanding, JobPrep++ stands as a reliable companion for career success.

REFERENCE

- [1] Brown, P., & Hesketh, A. (2004). **The Mismanagement of Talent: Employability and Jobs in the Knowledge Economy.** Oxford University Press.
- [2] McKinsey & Company (2020). **Future of Work: The Impact of AI on Job Readiness and Skill Development**
- [3] World Economic Forum (2023). **The Future of Jobs Report 2023: Reskilling and Upskilling in a Digital World**
- [4] LinkedIn Learning (2022). **The Rise of AI in Career Development and Job Preparation**
- [5] Harvard Business Review (2021). **The Role of AI and Data-Driven Learning in Job Market Preparation**
- [6] National Skills Coalition (2023). **Closing the Skills Gap: The Need for Personalized Learning in Workforce Development**