

Phase 1: Necessary Features and Functionalities

1. Introduction

iGenuine Learning provides technical training in C, C++, Python, Java, and JavaScript, along with non-technical training in soft skills, communication, verbal, quants & reasoning training, to students across various colleges and universities.

To ensure effective learning outcomes and track progress in both technical and non-technical areas, we require an assessment portal that will facilitate assessment tasks related to our training programs.

2. Objective

The objective of the assessment portal is to:

- Provide a centralized platform for conducting assessments related to technical and non-technical training programs.
- Enable students to complete assessments remotely and conveniently.
- Allow trainers and administrators to create, manage, and evaluate assessment tasks in both technical and non-technical domains.
- Provide login access to colleges to view reports of their students' performance.

3. Functional Requirements

3.1. User Roles

- **Admin:** Can create and manage assessment tasks, create account for students, view reports, manage user accounts, and assign roles.
- **Sub-Admin:** Can create and manage assessment tasks, view reports.
- **Student:** Can access and complete assessments assigned to them and view reports.
- **College:** Can view reports of their students' performance.

3.2. Assessment Creation and Management

- Admin and sub-admin should be able to create various types of assessment tasks, including technical quizzes, coding assignments, and soft skills assessments, as well as quants and reasoning assessments.
- Ability to categorize assessments based on training college, training programs, type of assessment and period of assessment.
- Allow for the inclusion of code snippets and programming exercises in technical assessments.

3.3. Coding Assessment

- **Code Snippets:** Admin/Sub-admin should be able to insert short sections of code into assessment questions. These snippets may demonstrate a particular concept, require students to identify errors, or ask students to modify or complete the code.
- **Programming Exercises:** Admin/Sub-admin should also have the ability to include complete programming exercises as part of the assessment. These exercises may require students to write a program to solve a problem, implement an algorithm, or demonstrate understanding of a specific programming concept.
- **Interactive Coding Environment:** The assessment portal should provide an interactive coding environment where students can view the code provided in the assessment and write their solutions. This environment should support syntax highlighting, code execution, and feedback on the correctness of the code.
- **Evaluation of Code:** The portal should be able to automatically evaluate code submissions for correctness. This may involve running the code against test cases or comparing it to a predefined solution. Admin/Sub-admin should also have the option to manually review and grade code submissions if needed.
- **Support for Multiple Programming Languages:** Since the training covers various programming languages like C, C++, Python, Java, and JavaScript, the assessment portal should support these languages for both code snippets and programming exercises.

3.4. Assessment Delivery

- Students should be able to access assessments through a user-friendly interface, including coding environments for programming assessments, and quants and reasoning assessments.
- Provide options for timed assessments and auto-submission upon completion.
- Ensure compatibility with various devices and web browsers.

3.5. Assessment Evaluation

- Automated grading for objective-type questions & coding assignments in technical assessments, as well as quants and reasoning assessments.
- Manual grading by admin/sub-admin for coding assignments and subjective-type questions.

3.6. Reporting and Analytics

- Generate detailed reports on student performance in technical, soft skills, quants, and reasoning assessments, including scores, completion rates, and strengths/weaknesses.
- Provide insights into trends and areas of improvement for individual students, technical topics, and training programs.
- Allow colleges to access reports of their students' performance.

Admin Reports:

- Comprehensive overview of all assessment tasks created and managed.
- Access to detailed reports on student performance, including scores, completion rates, and strengths/weaknesses.
- Ability to view reports on user accounts and roles, including activity logs and access history.
- Detailed analytics and insights into trends and areas of improvement for individual students, technical & non-technical topics, and training programs.

- Advanced analytics on overall assessment effectiveness and training program performance.

Student Reports:

- Personalized reports showing individual performance in completed assessments.
- Scores and feedback on each assessment task completed.
- Overview of strengths and weaknesses based on assessment results.
- Progress tracking and comparison against previous assessments.

College Reports:

- Reports on their students' overall performance in assessments.
- Access to detailed reports on student performance, including scores, completion rates, and strengths/weaknesses.
- Aggregate data on student scores, completion rates, and progress.
- Insights into trends and areas of improvement for students from their institution.

4. Non-functional Requirements

- **Security:** Ensure data privacy and protection of sensitive information, including student records and assessment results.
- **Scalability:** The portal should be able to handle a large number of users, assessments, and colleges.
- **Performance:** Minimal downtime and fast response times, even during peak usage.
- **Customization:** Ability to customize the portal's appearance and branding for colleges.

Phase 2: Advanced Features

1. Advanced Reporting and Analytics

- Customized reports tailored to the college's specific requirements and objectives.
- Advanced data visualization tools for more in-depth analysis of student performance in assessments, such as interactive charts and graphs.
- Predictive analytics to forecast student outcomes and recommend personalized learning paths.

2. Coding Window/Box

- Integration of a coding window/box within the assessment platform for students to write and execute code directly.
- Support for various programming languages such as C, C++, Python, Java, and JavaScript.

3. Question Bank

- Creation of a question bank to store a repository of assessment questions categorized by topics and difficulty levels for all assessments.
- Ability for admin/sub-admin to select and use questions from the question bank when creating assessments.

4. Proctoring

- Implementation of proctoring features for real-time monitoring and detection of suspicious behavior during assessments to ensure the integrity of assessments, including webcam monitoring, screen recording, and keystroke analysis across all assessment types.

5. Predefined Practice Questions

- Provision of predefined practice questions for students to prepare for assessments.

- Practice questions covering various technical topics, soft skills, and reasoning abilities to cater to different skill levels.

Phase 3: Advanced Features

6. Personalized Learning Paths

- AI-driven recommendation engine to suggest specific training modules or resources based on individual student performance and goals across all assessment types.
- Adaptive assessments that adjust difficulty levels based on student proficiency, ensuring personalized learning experiences.

7. Gamification

- Implementation of gamified elements (e.g., badges, leader boards, rewards) to increase student engagement and motivation across all assessment types.
- Interactive simulations and challenges to reinforce learning concepts in a fun and engaging way.

8. Learning Resources Repository

- Centralized repository for training materials, including lecture slides, videos, code examples, practice exercises, and quants and reasoning study materials.
- Advanced search and filtering capabilities to help students find relevant resources quickly.
- Video Embedding Feature to integrate educational videos directly within the platform, providing students with easy access to visual learning content.

Conclusion

Phase 1 covers necessary functionalities required to establish a robust assessment portal for our training programs, while Phase 2 and 3 introduces advanced features aimed at enhancing

the learning experience, promoting collaboration, providing personalized learning opportunities, and ensuring the integrity of assessments across all assessment types.

By implementing the three phases, we aim to create a comprehensive and dynamic platform that maximizes the effectiveness of our training programs.