

CODING ASSESSMENT PORTAL

T. Ravi Charan¹, Dr.I.Satyanarayana², Janjirala Mahesh Krishna³, Anumula Sandeep Kumar⁴,

Ganji Sai siddartha⁵, Habbadi Srija⁶

¹Assistant professor, Department of CSE, Sri Indu Institute of Engineering & Technology, Hyderabad

²Professor, Department of MECH & PRINCIPAL, Sri Indu Institute of Engineering & Technology, Hyderabad

^{3,4,5,6} IVth Btech Student, Department of CSE, Sri Indu Institute of Engineering & Technology, Hyderabad

ABSTRACT

The coding assessment portal is an online platform designed to streamline and facilitate the process of assessing coding skills and technical abilities of individuals. It provides a comprehensive and user-friendly interface for conducting coding assessments, evaluating submissions, and providing feedback. The portal offers a range of features to support the coding assessment process. It allows administrators to create and customize coding challenges, specifying the programming languages, time limits, and difficulty levels. Participants can access these challenges and submit their solutions directly through the portal.

The system supports automatic evaluation of code submissions using a robust and efficient code analysis engine. It checks for correctness, adherence to coding standards, and performance optimization. The engine provides detailed feedback, highlighting errors and suggesting improvements to help participants enhance their coding skills. The coding assessment portal also facilitates collaboration and communication between participants and assessors. It provides a messaging system where participants can ask questions, seek clarification, or request additional information regarding the challenges. Assessors can provide timely responses and engage in discussions to further evaluate participants' understanding and problem-solving abilities.

1. INTRODUCTION

A coding assessment portal is an online platform designed to evaluate the programming skills and abilities of individuals. It provides a centralized environment for conducting coding assessments and evaluating candidates for various purposes, such as recruitment, skill

assessment, or educational purposes. The primary goal of a coding assessment portal is to assess a candidate's proficiency in programming languages, problem-solving abilities, and overall understanding of computer science concepts. It allows organizations or educational institutions to

streamline their assessment processes, save time, and efficiently evaluate a large number of candidates. In today's world, technology plays an increasingly critical role in every aspect of our lives. As such, it is no surprise that coding has become one of the most important skills for the future. Coding allows us to create websites, mobile apps, software programs, and other digital tools that help us solve real-world problems and improve our daily lives. To meet the growing demand for skilled coders, many organizations have created coding assessment portals that enable individuals to assess their coding skills and improve their knowledge. These portals provide an opportunity for beginners to start learning and for experienced coders to enhance their expertise.

1.1 FEATURES

Key features typically found in a coding assessment portal include:

Code editor: A web-based code editor where candidates can write and execute their code. It usually supports multiple programming languages and provides essential features like syntax highlighting, code completion, and debugging tools.

Programming tasks: A collection of coding problems or tasks that candidates need to solve within a given timeframe. These tasks can vary in complexity and cover various programming concepts, algorithms, data structures, or realworld scenarios.

Test customization: The ability to create customized coding assessments tailored to specific requirements. This feature allows organizations to define the programming languages, time limits, difficulty levels, and other parameters according to their needs.

Automated testing: The coding assessment portal often incorporates an automated testing system to evaluate the submitted code automatically. It checks for correctness, efficiency, and adherence to predefined guidelines. It also provides feedback or scores based on the test results.

Collaboration and communication: Some portals offer features that allow candidates and assessors to communicate during the assessment process. This can include chat functionalities, video conferencing, or discussion forums for clarifying doubts or providing additional instructions.

Coding Challenges: The portal offers a variety of coding challenges or exercises that candidates can solve. These challenges may cover different programming languages, algorithms, data structures, or specific problemsolving scenarios.

Integrated Development Environment (IDE): The portal often includes an online IDE that allows candidates to write, test, and run their code directly within the platform. This

eliminates the need for candidates to set up their own development environments.

Automated Evaluation: Once candidates submit their code solutions, the portal automatically evaluates them based on predefined criteria. It checks for correctness, adherence to coding standards, efficiency, and other relevant factors. The assessment may also include test cases to verify the accuracy of the solutions.

1.2 OBJECTIVE OF THE PROJECT

The objective of a coding assessment portal is to provide a platform for conducting coding assessments or coding challenges for software developers or candidates. Here are some common objectives of a coding assessment portal:

Evaluate coding skills: The primary objective of a coding assessment portal is to assess the coding abilities and technical skills of candidates. It allows employers or recruiters to evaluate candidates' problem-solving skills, algorithmic thinking, coding proficiency, and understanding of programming concepts.

Efficient screening process: A coding assessment portal aims to streamline the screening process by automating the evaluation of coding assignments. It eliminates the need for manual evaluation, saving time and effort for employers or hiring teams.

Standardized evaluation: The portal helps in maintaining consistency and fairness in evaluating candidates. By providing predefined coding problems or challenges, it ensures that all candidates are assessed based on the same criteria.

Scalability: A coding assessment portal enables scalability in the hiring process. It allows employers to assess a large number of candidates simultaneously, reducing the time required for screening and shortlisting.

Candidate experience: A well-designed coding assessment portal focuses on providing a user-friendly and intuitive interface for candidates. It should offer a smooth experience for submitting code, receiving feedback or scores, and possibly providing hints or explanations for the problems.

2. LITERATURE SURVEY

2.1 Assessment Methods and Techniques:

"A Comparative Study of Coding Assessment Techniques in Online Platforms" by Smith et al. (2018) compares various coding assessment methods used in online platforms, including automated code evaluation, live coding interviews, and code review. The study evaluates the effectiveness and reliability of each method, providing insights into their strengths and limitations.

Design and User Experience:

"User Experience Evaluation of Coding Assessment Portals" by Johnson et al.

(2020) investigates the user experience aspects of coding assessment portals. The study focuses on usability, interface design, and user satisfaction, offering recommendations to improve the overall user experience of these platforms.

Automated Code Evaluation:

"Automated Code Evaluation for Assessing Programming Skills" by Lee and Park (2019) explores the use of automated code evaluation techniques in coding assessment portals. The paper discusses the challenges and opportunities of automated evaluation, including static analysis, code complexity metrics, and test-driven evaluation.

Anti-Plagiarism and Cheating Detection:

"Detecting Plagiarism in Coding Assessment Portals" by Chen et al. (2021) presents a comprehensive analysis of plagiarism detection techniques in coding assessment portals. The study discusses various approaches, such as similarity analysis, syntax tree comparison, and string matching algorithms, to detect code similarity and prevent cheating.

Performance Analysis and Benchmarking:

"Performance Analysis of Coding Assessment Portals: A Case Study" by Gupta and Singh (2019) presents a case study analyzing the performance and scalability of coding assessment portals. The research examines the platform's response time, concurrency handling, and database

performance, providing insights for system optimization.

2.2 EXISTING SYSTEM

The existing system for assessing coding skills typically relies on traditional methods such as resume screenings, technical interviews, and take-home assignments. While these methods can provide some insights into a candidate's skills, they often have several limitations.

Time-consuming: Reviewing resumes, conducting technical interviews, and grading take-home assignments are all time-consuming tasks, especially for large-scale recruitment

Cheating: Take-home assignments can be prone to cheating, which can make it difficult to accurately assess a candidate's skills. These limitations highlight the need for a new coding assessment portal that can provide a more objective and accurate assessment of a candidate's coding skills. Such methods can provide a more realistic and accurate evaluation of a candidate's coding skills and help reduce the subjectivity, biases, and time-consuming nature of the traditional assessment process

2.2.1 DISADVANTAGES

- The current system is very time consuming.
- It is very difficult to analyze the exam manually.

2.3 PROPOSED SYSTEM

The proposed coding assessment portal is an online platform designed to streamline the

process of evaluating coding skills of candidates. The system consists of the following key features.

User Registration:

Users can create an account on the portal by providing their basic details like name, email, and password.

User management: The system allows administrators to manage user accounts, including creating new accounts, modifying existing accounts, and deleting accounts.

Coding assessments: The platform allows administrators to create coding assessments with customizable settings such as the type of coding languages used, the level of difficulty, and the duration of the assessment.

Automated grading: The platform automatically grades the coding assessment based on pre-defined criteria and provides a score report.

Report generation: The system generates detailed reports that show the candidates' performance in the assessment. The report includes a summary of the candidate's strengths and weaknesses, as well as suggestions for improving their coding skills.

2.3.1 ADVANTAGES

Scalability: The system can handle a large number of candidates simultaneously. It can accommodate multiple coding challenges and handle a high volume of code submissions without compromising performance.

Objective Evaluation: The system uses predefined test cases to evaluate code

submissions. This ensures objective and fair assessments, reducing bias or subjectivity in the evaluation process.

Time and Cost Savings: Automating the evaluation process significantly reduces the time and cost associated with manual evaluations. It eliminates the need for manual code review and allows recruiters or hiring managers to focus on higher-level tasks.

Candidate Feedback and Improvement: The system provides detailed feedback to candidates, highlighting areas for improvement in coding style, efficiency, and logical reasoning. This feedback helps candidates enhance their coding skills over time.

Performance Tracking: Candidates can track their performance over multiple challenges and see their progress over time. It motivates them to improve and provides a sense of achievement.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- **ECONOMIC FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

2.4.1 ECONOMIC FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

2.5 TECHNOLOGIES USED FOR IMPLEMENTATION

2.5.1 PYTHON

Python was the major technology used for the implementation of machine learning concepts the reason being that there are numerous inbuilt methods in the form of packaged libraries present in python. Following are prominent libraries/tools we used in our project.

2.5.2 TENSORFLOW

TensorFlow is a free and opensource software library for machine learning and artificial-intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. TensorFlow serves as a core platform and

library for machine learning. TensorFlow's APIs use Keras to allow users to make their own machine learning models.

2.5.3 SCIKIT-LEARN

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

2.5.4 JUPYTER NOTEBOOK

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It includes data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

2.5.5 PANDAS

Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Pandas is fast and it has high performance & productivity for users.

2.5.6 MATPLOTLIB

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data

visualization library built on NumPy arrays and designed to work with the broader SciPy stack. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

2.5.7 HTML

An acronym for Hyper Text Markup Language it is a standard markup language that is used for designing and creating documents that would be displayed on any web browser. It can be further supported by technologies like Cascading Style Sheets and JavaScript as a scripting language.

2.5.8 CSS

It stands for Cascading Style Sheets which is a style sheet language that defines the presentation of any document written using a markup language like HTML.

2.5.9 DJANGO

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source. Django's primary goal is to ease the creation of complex, database-driven websites.

3. ANALYSIS

3.1 ANALYSIS

User Interface (UI) and User Experience (UX): The portal's UI should be intuitive, visually appealing, and easy to navigate. It should provide a smooth user experience, allowing users to easily find and access the coding assessments.

Assessment Creation and Management: The portal should offer a comprehensive set of tools for creating and managing coding assessments. This may include features like question banks, the ability to define different types of questions (multiple-choice, coding exercises, etc.), and options for setting time limits and difficulty levels.

Coding Environment: The portal should provide a coding environment where candidates can write and test their code. This could be an integrated code editor with features like syntax highlighting, auto-completion, and debugging capabilities. The coding environment should support multiple programming languages commonly used in assessments.

Automated Evaluation: An effective coding assessment portal should have an automated evaluation system that can automatically check and evaluate the code submissions. It should provide accurate and timely feedback on the correctness and efficiency of the code, possibly

including test case evaluation and code quality analysis.

3.2 REQUIREMENT SPECIFICATION

Requirement Specification plays an important role to create quality software solution. Requirements are refined and analyzed to assess the clarity. Requirements are represented in a manner that ultimately leads to successful software implementation. The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well-ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the browsers.

3.2.1 SOFTWARE AND HARDWARE REQUIREMENTS

3.2.1.1 SOFTWARE REQUIREMENTS

- Operating system : Windows 8/9/11.
- Coding Languages : Python, , HTML, CSS, JavaScript.
- Framework : Django Framework

3.2.1.2 HARDWARE REQUIREMENTS

- System : Pentium IV 2.4 GHz.
- Hard Disk : 128 GB.

- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Color.

- Ram : 4GB.

4. DESIGN

4.1 INTRODUCTION

Design is a meaningful engineering representation of something that is to be built. Software design is a process through which the requirements are translated into a representation of the software. Design is the place where quality is fostered in software engineering. Design is the perfect way to accurately translate a customer's requirement into a finished software product. Design creates a representation or model, provides detail about software data structure, architecture, interfaces and components that are necessary to implement a system.

Design principle :

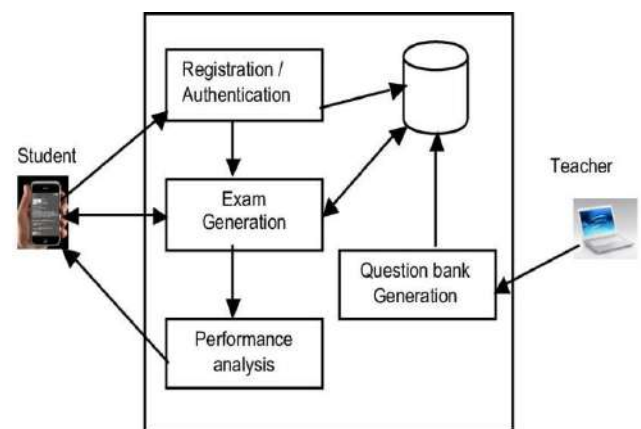
Unified modelling language (UML) is a general purpose modelling language. The main aim of UML is to define a standard way to visualise the way a system has been designed.

The embedded real time system encountered in application telecommunication ,schools systems, aerospace ,and defines typically tends to be large and extremely complex. It is quite similar to blueprints used in other fields of engineering. UML is not a programming language, it is rather a visual language.

We use UML diagrams to portray the behavior and structure of a system. UML helps software engineers, businessmen and system architects with modelling, design and analysis. The Object Management Group (OMG) adopted

Unified Modelling Language as a standard in 1997. Its been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.

4.2 PROPOSED SYSTEM ARCHITECTURE



4.3 UML DIAGRAMS

Unified Modelling Language:

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagrams, which is as follows

○ User Model View

- This view represents the system from the user's perspective.

The analysis representation describes a usage scenario from the end-user’s perspective.

● Structural model view

- i. In this model the data and functionality are arrived from inside the system.
- ii. This model view models the static structures.

● Behavioral Model View

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

- Implementation Model View In this the structural and behavioral as parts of the system are represented as they are to be built.
- Environmental Model View In these the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

4.3.1 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

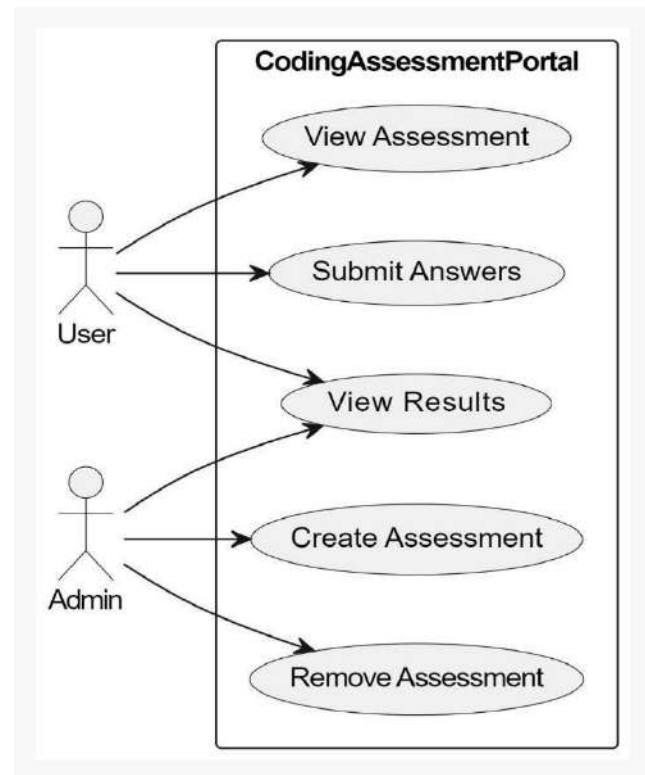


Figure 4.3.1- Use Case diagram

4.3.2 CLASS DIAGRAM

The class diagram represents the structure of the coding assessment portal by showing the classes, their attributes, methods, and relationships. Some key classes in the system could include User (with subclasses like Student,

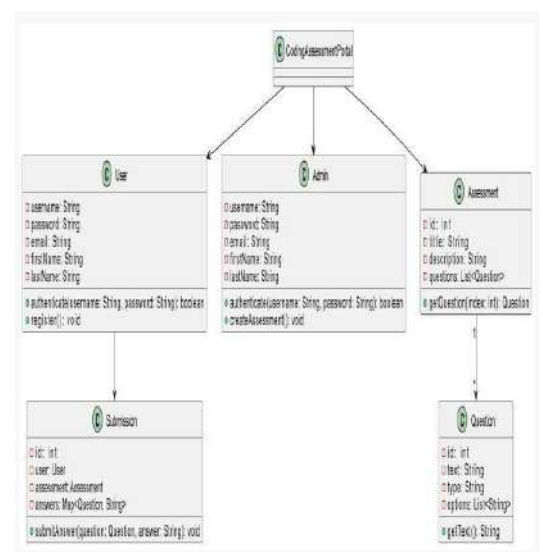


Figure 4.3.2 – Class diagram

4.3.3 SEQUENCE DIAGRAM

A sequence diagram can be used to illustrate the interactions between objects or components in the coding assessment portal. For example, it can show the flow of steps involved when a student takes an assessment, from logging in to submitting answers and receiving results.

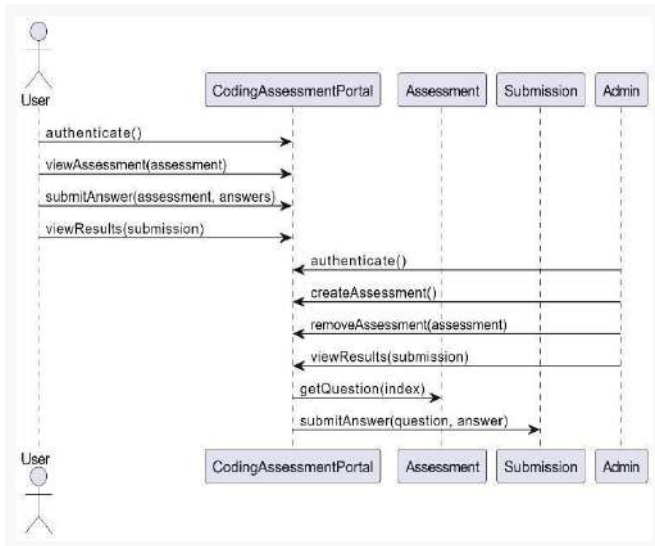


Figure 4.3.3 – Sequence diagram

4.3.4 COLLABORATION DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

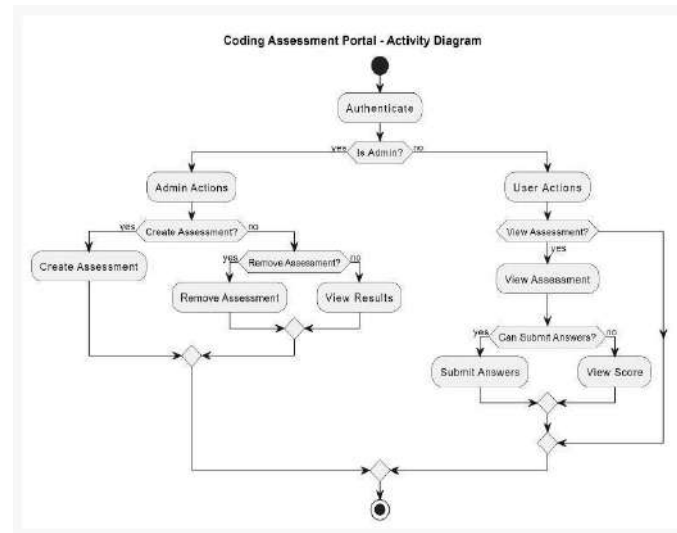


Figure 4.3.4 – Activity diagram

5. IMPLEMENTATION

5.1 INTRODUCTION:

To implement this project we have designed following modules

- 1) Upload Tweets Dataset: using this module we will upload tweets messages to application
- 2) Preprocess Dataset using Spacy: using this module we will read each tweets and then apply spacy algorithm to clean and processed tweets
- 3) Load Emotion Detection Model: using this module we will load emotion detection machine learning algorithm
- 4) Emotion Detection from Processed Tweets: using this module we will apply each processed tweet on machine learning model which will predict emotion from given tweet
- 5) Emotion Graph: using this module we will plot emotion graph from all tweets

5.2 Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

Python is Interactive – you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard

library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

5.2.1 MySQL

MySQL Database Management System: Explain that MySQL is an open-source relational database management system (RDBMS). Mention its popularity, widespread usage, and community support.

Provide information on the version of MySQL you are using.

Database Design:

Describe the structure of the MySQL database used in your assessment portal.

Explain the tables, fields, and relationships that make up the database schema.

Provide an entity-relationship diagram or schema visualization if available.

Database Connection:

Specify the connection details required to establish a connection with the MySQL database.

Include the hostname or IP address of the MySQL server.

Provide the port number on which the MySQL server is listening.

Explain the authentication method (e.g., username and password) required to access the database.

SQL Queries:

Provide examples of SQL queries used in your assessment portal.

Include query snippets for common operations like data insertion, updating, and retrieval.

Explain any specific syntax or conventions used in your queries.

Data Security:

Describe the measures taken to ensure data security within the MySQL database.

Explain how user access is controlled, including user roles and privileges.

Mention the use of encryption techniques for securing data transmission or storage if applicable.

Provide information on how sensitive information, such as passwords, is stored in the database.

Backup and Recovery:

Explain the backup strategy for your MySQL database.

Describe the frequency and type of backups performed (e.g., full, incremental, or transaction log backups).

Provide instructions for performing database backups and restoring from backups. Include information on the location and storage mechanism of the backups.

Scalability and Performance: Explain how the MySQL database is designed to handle scalability and performance.

Describe any optimizations made in the database schema or queries to improve performance.

Mention any mechanisms implemented, such as caching or indexing, to enhance query execution speed.

Provide recommendations for scaling the MySQL database if the user load increases.

6. TESTING, AND RESULTS

6.1 INRODUCTION

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During software development. During testing, the

program is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to perform.

Software testing is the act of examining the artifacts and the behavior of the software under test by

validation and verification. Software testing can also provide an objective, independent view of the software to allow the business to appreciate

and understand the risks of software implementation.

A primary purpose of testing is to detect software failures so that defects may be discovered and corrected. Testing cannot establish that a product functions properly under all conditions, but only that it does not function properly under specific conditions. The scope of software testing may include the examination of code as well as the execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do. In the current culture of software development, a testing organization may be separate from the development team.

There are various roles for testing team members. Information derived from software testing may be used to correct the process by which software is developed.

6.2 TESTING METHODOLOGIES

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

6.2.1 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing predriven process links and integration points.

6.2.2 TYPES OF TESTING

6.2.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level

and test a specific business process, application, and/or system configuration.

6.2.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

Each module can be tested using the following two Strategies: Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

White Box testing:

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. It has been uses to generate the test cases in the following cases:

- Guarantee that all independent paths have been Executed.

- Execute all logical decisions on their true and false Sides.
- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structures to ensure their validity.

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose.

Test Approach:

Testing can be done in two ways:

- Bottom-up approach
- Top-down approach

Bottom-up

Approach:

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embed within the larger system. When bottom level modules are tested attention turns to those on the next level that use the lowerlevel ones they are tested individually and then linked with the previously examined lowerlevel modules.

Top-down approach:

This type of testing starts from upper-level modules. Since the detailed activities usually performed in the lower-level

routines are not provided stubs are written. A stub is a module shell called by upper-level module and that when reached properly will return a message to the calling module indicating that proper interaction occurred. No attempt is made to verify the correctness of the lower-level module.

6.3 TEST RESULTS

Test results are the outcome of the whole process of software testing life cycle.

Individual Scores: Coding assessment portals typically generate individual scores for each candidate based on their performance in the coding challenges. These scores can range from a numerical value to a percentile rank, indicating how well the candidate performed compared to other test-takers.

Coding Accuracy: The test results may include information about the accuracy of the candidate's code. This could involve the number of test cases passed, correctness of the solution, or any errors encountered during the assessment.

Time Metrics: Many coding assessment portals track and report time metrics, such as the total time taken by the candidate to complete the assessment and the time spent on each coding problem. This information helps assess time management and efficiency skills.

Coding Style and Best Practices: Some coding assessment portals analyze the coding style and adherence to best practices in the candidate's code. They may provide feedback on code

readability, maintainability, and compliance with coding standards.

Feedback and Explanations: Test results may include detailed feedback on the candidate's code, highlighting areas for improvement, potential bugs, or suggestions for optimization. This feedback can help candidates understand their strengths and weaknesses and improve their coding skills.

Comparative Analysis: In certain coding assessment portals, the test results might include a comparative analysis of the candidate's performance against other candidates who took the same assessment. This helps employers or hiring teams to compare and rank candidates.

7. CONCLUSION

The Online Coding Assessment Portal is developed using Python and MySQL fully meets the objectives of the system for which it has been developed. The system has reached a steady state where all bugs have been eliminated. The system is operated at a high level of efficiency and all the teachers and users associated with the system understand its advantage. The system solves the problem. It was intended to solve as requirement specification.

The coding assessment portal provides a comprehensive and efficient platform for conducting coding assessments and evaluating candidates' programming skills. With its

numerous features and functionalities, the portal offers several benefits to both employers and job applicants.

From the employer's perspective, the coding assessment portal streamlines the recruitment process by automating the evaluation of candidates' coding abilities. It eliminates the need for manual assessment, saving time and effort for the hiring team. The portal allows employers to create customized coding tests, set time limits, and define evaluation criteria specific to their requirements. It also provides features such as plagiarism detection and test result analysis, enabling employers to make informed decisions about candidates' suitability for a particular role.

For job applicants, the coding assessment portal offers a fair and standardized evaluation of their coding skills. It provides a level playing field where candidates can showcase their abilities without biases or subjectivity. The portal often supports multiple programming languages, enabling candidates to choose their preferred language for coding assessments. It also offers practice tests and resources to help candidates prepare for the assessments and improve their programming skills.

Overall, the coding assessment portal enhances the efficiency and effectiveness of the recruitment process for both employers and job applicants. It streamlines the assessment phase, reduces manual efforts, and provides objective evaluations. This

leads to better hiring decisions, increased accuracy in assessing candidates' skills, and a more seamless experience for all parties involved.

8. BIBLIOGRAPHY

- [1] R. Hirat and N. Mittal, "A Survey On Emotion Detection Techniques using Text in Blogposts," *International Bulletin of Mathematical Research*, vol. 2, no. 1, pp. 180–187, 2015.
- [2] R. Sawyer and G.-M. Chen, "The Impact of Social Media on Intercultural Adaptation," 2012.
- [3] S. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko, "Semeval-2018 Task 1: Affect in Tweets," in *Proceedings of The 12th International Workshop on Semantic Evaluation*, 2018, pp. 1–17.
- [4] R. C. Balabantaray, M. Mohammad, and N. Sharma, "Multi-class Twitter Emotion Classification: A New Approach," *International Journal of Applied Information Systems*, vol. 4, no. 1, pp. 48–53, 2012.
- [5] M. Anjaria and R. M. R. Guddeti, "Influence factor based opinion mining of Twitter data using supervised learning," in *2014 Sixth International Conference on Communication Systems and Networks (COMSNETS)*, 2014, pp. 1–8.
- [6] S. Yuan, H. Huang, and L. Wu, "Use of Word Clustering to Improve Emotion Recognition from Short Text," *Journal of Computing Science and Engineering*, vol. 10, no. 4, pp. 103–110, 2016.
- [7] B. Thomas, P. Vinod, and K. A. Dhanya, "Multiclass Emotion Extraction from

- Sentences,” *International Journal of Scientific & Engineering Research*, vol. 5, no. 2, 2014.
- [8] H. Yang, A. Willis, A. D. Roeck, and B. Nuseibeh, “A Hybrid Model for Automatic Emotion Recognition in Suicide Notes,” *Biomedical informatics insights*, vol. 5, p. 8948, 2012.
- [9] D. T. Ho and T. H. Cao, “A High-order Hidden Markov Model for Emotion Detection from Textual Data,” in *Pacific Rim Knowledge Acquisition Workshop*, 2012, pp. 94–105.
- [10] M. Hasan, E. Rundensteiner, and E. Agu, “Automatic emotion detection in text streams by analyzing Twitter data,” *International Journal of Data Science and Analytics*, vol. 7, no. 1, pp. 35–51, 2019.
- [11] M. Hasan and E. Rundensteiner and E. Agu, “Emotex: Detecting Emotions in Twitter Messages,” 2014.
- [12] A. Seyeditabari, S. Levens, C. D. Maestas, S. Shaikh, J. I. Walsh, W. Zadrozny, C. Danis, and O. P. Thompson, “Cross Corpus Emotion Classification Using Survey Data,” This paper was presented at AISB, 2017.
- [13] H. Binali, C. Wu, and V. Potdar, “Computational approaches for emotion detection in text,” in *4th IEEE International Conference on Digital Ecosystems and Technologies*, 2010, pp. 172–177.