Velankani Coding Challenge Tool

Abstract :

Velankani is currently facing recruitment challenges in hiring good programmers at all levels.

The current hiring process has multiple rounds of interviews where coding rounds are sometimes done and sometimes skipped. It depends on the panel and the position being interviewed for. There is no standard way of assessing the candidate’s programming ability in the current process.

This project aims to overcome that by way of developing an internal coding challenge web application similar to the lines of hackerrank, codewars, challengerocket and other tools. The objective is to have the aspiring candidate undergo a coding challenge round as the first round of interview using this project.

This project henceforth called the “Velankani Coding challenge Tool” is planned to be developed internally.

Scope:

The scope of this document is to provide high level requirements to a tool called “ Velankani coding Challenge Tool”. This is a web application and is to be developed in-house for internal consumption.

Functional diagram

<TBD>

**Phase 1 requirements:**

**Functional Requirements for the “ Velankani coding Challenge tool”**

1. The tool should support providing coding challenges via programming questions for Java .Only java support will be added in phase 1.
2. In future it should be easy to plug in any new challenge support. Eg. C++ , .net..
3. It should have a databank of coding challenges and objective type questions.
4. It should have the ability to pick a random question from the data bank of questions from a database for a particular test. It will also support picking specific questions from databank.
5. The answers are to be evaluated by the tool using a testing framework. An example is Junit for Java programs.
6. The coding challenge tool shall have a web interface to provide the user to type the program similar to a java editor in popular IDEs like eclipse. Eg. Hackerrank is a good example for the interface to be provided to the user.
7. The tool should have a backend process to compile the entered java program and return the errors if any to the user.
8. It should have a report of the challenges and the results per user.

**Screens:**

1. Login screen
   1. User name and password should be in input fields. After successful authentication it should redirect to either error screen or next screen based on user type.
   2. In case the user is admin – it should redirect to admin dashboard
   3. In case the user is recruitment user – it should redirect to recruitment user dashboard
   4. In case the user is candidate dashboard it should redirect to test screen.
2. Admin Dashboard
   1. Display the past coding test reports
   2. Display the objective type coding test reports
   3. Display the tests in progress
   4. Provision to upload questions and their answers
   5. Provision to create new users and assign roles
3. Recruitment dashboard
   1. Display the list of candidates for whom coding test should be scheduled
   2. Display the list of questions for the test
   3. Provision to choose a candidate and assign questions for test. It couple be objective type or coding test questions
   4. View the test results
   5. Provision to forward the result to interview panel via email
4. Candidate screen
   1. This screen should be a minimal screen for the candidate to answer the preselected question
   2. It should have provision to submit answers depending on the question type
5. Interview Panel screen
   1. This screen will have the result of the questions answered by the candidate
   2. The assigned interview panel can see the answers to the specific question as determined by the Recruitment user in the recruitment dashboard.

**RBAC Requirements**

1. This tool should support five user types ( roles)
   1. Admin user – who can add and delete questions and can add new users
   2. Recruitment user – who can schedule the coding challenges to candidates and check the result
   3. Candidate user – who should be able to login to the tool to take coding challenge
   4. Interview Panel user – who should be able to see the results for candidates assigned to him for interview
   5. Guest user – who should be able to sign up from login page. This type of user will have only one default coding challenge
2. There should be a login for the candidate to provide basic details like Name and yrs of exp.
3. There should be a separate admin login to load new questions and to create a new user for the candidate to login

**General Requirements:**

1. Alternately admin user can also choose specific questions to be answered by the candidate.
2. It should preferably be a web application and should use the latest technologies as much as possible
3. It should be extensible – should be easy to plugin a new type of challenge in a different programming language.
4. Architecture should be flexible to integrate with third party applications like the recruitment portals. REST interface support is suggested.

**Phase 2 requirements:**

**Integration with Velankani recruitment portal – Requirements <TBD>**

1. This will be defined for phase 2.
2. Python support
3. Once the challenge result is available it should be emailed to the panel who is supposed to take the next round of interview
4. Questions should be randomly selected based on the yrs of experience of the candidate.