CODE EVALUATION

Project Report Submitted To Gujarat University

In partial fulfilment of the requirements for the award to the Degree of

MASTER OF COMPUTER APPLICATIONS SEMESTER - VI

GUIDED BY: SUBMITTED BY:

Dr. Hardik Joshi Ajinkya Rathod



DEPARTMENT OF COMPUTER SCIENCE GUJARAT UNIVERSITY, AHMEDABAD

YEAR: 2021-22

CODE EVALUATION

Project Report Submitted To Gujarat University

In partial fulfilment of the requirements for the award to the Degree of

MASTER OF COMPUTER APPLICATIONS SEMESTER – VI

GUIDED BY: SUBMITTED BY:

Dr. Hardik Joshi Ajinkya Rathod (60024)



DEPARTMENT OF COMPUTER SCIENCE GUJARAT UNIVERSITY, AHMEDABAD

YEAR: 2021-22

Department of Computer Science

Gujarat University



Certificate

| R oll No :30 | | | Seat No : <u>60024</u> |
|------------------------------|----------------------------|-----------------|------------------------------------|
| This is to certify tha | t Mr. / Ms. | _Ajinkya Sreek | ant Rathod |
| student of MCA Semeste | er – VI has du | ly completed | his/ her project titled |
| Code Evaluation | for the sem | ester ending in | June 2022, towards |
| partial fulfillment of his/h | 4 degree of Mas | ster of Compute | er Applications. |
| | | | |
| Date of Submission | | | Internal Faculty |

Head of Department

23rd June, 2022



Project Completion Certificate

Date: June 17, 2022

This is to certify that the project entitled **Code Evaluation** has been successfully completed by Mr. Ajinkya Rathod at Incubyte Consulting LLP.

This project was carried out from 10th January 2022 to 17th June 2022.

*Attendance: No. of Present days 110 / 114 total working days (No. of leaves: 4)

Date: 17-06-2022

Place: Ahmedabad

Sign: Grohi Parikh

Name: Arohi Parikh

Table of Contents

| Acknowledgement | 6 |
|--------------------------------------------|----|
| About the Organization | 7 |
| Introduction | 8 |
| About the Project | 9 |
| Problem statement | 10 |
| Objective | 11 |
| System Engineering | 13 |
| System Overview | 13 |
| Features | 13 |
| Challenges | 14 |
| Proposed system | 15 |
| Requirement Analysis and Feasibility Study | 16 |
| Functional Requirements | 16 |
| Non-Functional Requirements | 17 |
| Technical Requirements | 17 |
| Libraries and Frameworks | 18 |
| Dynamic Program Analysis | 19 |
| API Requests and Responses | 21 |
| Advantages | 23 |
| Automated Code Assessment | 24 |
| Features | 24 |
| Security | 24 |
| Challenges for Automated Code Assessment | 25 |
| Collaboration and Interoperability | 25 |



| System Design | 26 |
|------------------------------------|----|
| Database design | 26 |
| Users Table | 26 |
| Question Table | 27 |
| Response Table | 28 |
| Diagrams | 29 |
| Use Case Diagram | 29 |
| Use Case Diagram | 30 |
| Sequence Diagram | 31 |
| Class Diagram | 32 |
| Screenshots | 33 |
| What if Errors? | 47 |
| System Testing | 48 |
| Development Style | 50 |
| Fakes, mocks and integration tests | 51 |
| Keep the unit small | 52 |
| Individual best practices | 52 |
| Example of TDD | 53 |
| Future Enhancement | 56 |
| Bibliography | 57 |
| Links | 57 |
| Books | 57 |
| Articles | 57 |



Acknowledgement

I'd want to show my gratitude to Incubyte Consulting LLP, particularly the co-founders Rushali Parikh and Sapan Parikh, for giving me this opportunity. Thanks in particular to Software Craftsperson and Team Lead for their unwavering assistance. Also, I want to express my gratitude to every member of the team, especially the Incubyte operations team, who assisted me directly or indirectly by providing required counsel and direction, as well as arranging all necessary facilities to help me get to this stage. I've chosen this moment to express my gratitude for their work.

I'd like to express my gratitude to Dr. Jyoti Pareek, our Department Head, for her advice and assistance. I'd also want to thank Dr. Hardik Joshi, my Internal Project Guide, for giving expert guidance and keeping me in touch during my project duration. Also, I'd like to thank each and every faculty member and department staff member for their unwavering support.

Lastly, I would be remiss in not mentioning my parents & everyone around me. Their belief in me has kept my spirits and motivation high during this process. I would also like to thank my classmates & friends for all the entertainment and emotional support.



About the Organization

Incubyte is a remote-first firm that assists clients in developing futureready software that can adapt to changing business needs. We encourage SDLC approaches that generate consistent and predictable product behavior since quality is at the forefront of our minds.



With best practices such as Software Craftsmanship, Extreme Programming, Agile Mindset, TDD, Constant Refactoring, Continuous Delivery, and Automation Testing, Incubyte aims to focus equally on the explicit and implicit aspects of software development, including better scalability, maintainability, testability, and security.

They assist in the development of new goods as well as the enhancement of existing ones' feature sets. We use software development best practices to give value on a regular basis, allow for quick feedback, and ensure that your product evolves quickly.

The majority of programmers end up writing software for machines. When they first begin programming for humans, they are constantly on the lookout for ways to improve their consumers' experience and make an impact. Without continual learning and upskilling, none of the attributes we discuss are conceivable. People that recognize this make a concerted effort to expand their skill set's depth and breadth. This is reflected not only in the work that is completed, but also in more efficient partnering sessions, educational lightning lectures, and fascinating conversations. These are the Incubyte teams and individuals who contribute to the creation of high-quality products, delighted people, and client satisfaction!



Introduction

Computer science courses are becoming increasingly popular. As a result, more students are enrolling in computer science studies. Many programming assignments are assigned by computer science lecturers to improve students' learning results, however the evaluation procedure for programming assignments is not simple.

The grading of assignments became a burden for instructors. Compiling and testing programs with multiple inputs are time-consuming tasks. The amount of programming assignments that an instructor can assign during a course session is limited by these types of evaluations.

Automating the evaluation process is one solution to this challenge. This project introduces a web platform for computer science programming classes that allows students to submit assignments online and receive immediate feedback.

The purpose of this project is to make it simple to construct and evaluate student programming assignments in a consistent and error-free manner.

This application decreases the amount of time and effort required to evaluate each assignment, allowing instructors to incorporate more assignments in their classes. Because the program operates on two different frameworks and platforms, it can take advantage of both resources and benefits. This allows the application to handle multiple users more quickly and effectively.



About the Project

Data mining in education institutions is helping to analyze students' details and provide an effective evaluation system in a short time.

With the advancement of new technologies student's evaluation procedure has changed from manual correction to automate process of correction and analysis.

This candidate coding assignment evaluation system using API is designed to evaluate students coding correction process through the automation process.

When a candidate submits an answer for a student's question online faculty will evaluate coding by sending data to API and get results or error messages. By checking these messages faculty will give marks to students.

This process is done through a web application that is developed in python programming language.



Problem statement

Candidates' assignment evaluation is a time taking process for faculty which required a manual process by checking each line of code and give marks to students.

The task of evaluating a student program is a time consuming, but important, part of any computer programming course. Generally, it is not feasible to test each student's program thoroughly during lab sessions. Another problem is the student does not always perform the programs.



Objective

The coding evaluation process can be automated by using available code checking API's which can be integrated into the college assignment assigning website. Using this process evaluation is completed just in a click and faculty can give marks based on result.

Since most of our programming courses are in Java, we decided to build this system for Java, specifically. Moreover, we needed it to withstand buggy code and a large number of submissions at a given time. Also, we wanted the environment to accommodate the flexibility of different courses in our curriculum and fairly complex programs that can often be designed in any number of ways. And thus we set out with these design goals in mind for the Candidate and Hiring Person.

We can help conduct coding interviews and test programming skills of developers at scale – turning a challenge into one of your greatest opportunities.

The dynamic world of technology is ever evolving and with that our reliance on software programming increases. Modern day software programming is basically an online method of testing where cooperation between the software engineer and the computer is essential.

Online coding tests are tests sent to possible or potential future employees by an organization commonly to fill in as a zero or first round of interview to get an initial understanding of the candidate's coding capabilities. They are frequently given to new graduates or fresh college pass outs, yet any technical candidate can be subjected to an online coding test.

Few of the quality candidates can get their code correct right in the first attempt since troubleshooting, debugging and testing is a significant part of the programming cycle. To evaluate or assess a candidate's



programming skills, you have to give the prospective candidate a credible programming or coding test in which you can test their coding skills.

Having an online coding test instead of a manually written one also yields some intriguing bits of insights and knowledge. You can follow how the candidates perform on each and every question.

So, you can quickly detect the topic areas that prove to be complex and appear to discriminate against good candidates. Another major benefit of having an online coding test is that you can individually track each and every candidate's progress which can be plotted in a visual format, also known as the grade trajectory to get a better understanding.

Accordingly, you can also identify candidates who have not cracked your test yet can be employed as a trainable resource based on their various deductions on each and every question.



System Engineering

System Overview

The Web IDE component is the user-facing component in our architecture. Therefore, one of the primary requirements for it is to be user-friendly and to expose a minimum of the needed features without the unnecessary interface clutter.

On the other hand, the system daemon component is used to facilitate communication between Web IDE and system components, such as the code compiler.

This approach is in contrast to direct compiler access, and was selected due to the increased security and flexibility it provides, despite the additional complexity in terms of development and deployment it can cause

Features

✓ It's easy on the eye

Our feature on question listing has a more attractive interface. Adding sections is now a lot easier. By selecting the topic and the difficulty level, you can add questions.

- ✓ Easy sign-up process
- ✓ Code storage for easy re-editing



Challenges

The primary challenge faced during this project was social, rather than technical, in nature. The need to adapt to a practical and professional environment with developer deadlines implied finding a balance between the theoretical optimum and the solution most applicable to practice, rather than aiming solely for the optimal solution as a purely academic project would require. In the words of Einstein-

In theory there is no difference between theory and practice, but in practice there is!

The most significant technical challenge of this project concerned the variety of programming languages targeted. While familiar with some of the languages considered, at the inception of this research I was by no means a security expert for any of them. Therefore, analyzing and identifying the various categories of security vulnerabilities for each language was a substantial undertaking

An option less demanding on the developers would be to have more thorough tools available than those selected during this project. This is partially achieved already, as some critical pieces of software have been verified with formal methods.



Proposed system

The student online coding evaluation system provides an automatic coding checking process through which faculty can assign coding assignments and get results from students and compile code in click and check result and give marks.

In this work, we will propose a system that allows teachers to assign programming homework to their pupils. In this case, he can describe the programming task and associated verification program, and the students will submit their work. According to the teacher's instructions, the system verifies the submitted programming exercises. This evaluation process can assist any computer study center with programming-related courses such as the Program code. It will assist the instructor in promoting the subject and improving course performance as well as understanding quality.

A Web IDE:

Our prototype is only the initial step in implementing the concept presented in this paper. It uses a g++ compiler installed on a Linux Web server. Although some of its features are still rudimentary the IDE enables writing C++, Java and Python programs their execution and debugging. The online C++ IDE was therefore developed as an example of a learning-oriented Web IDE

Overheads for set up

✓ The overheads for setting up an assignment to allow for automatic evaluation are I general design in terms of functions/parts that can be given for independent assessment, and (ii) delivering a minimal body of code to exercise the parts/functions supplied in a submission.



✓ Our approach assists instructors in reducing the amount of time it takes to set up an assignment and supports the reuse of set-up for future assignments.

Requirement Analysis and Feasibility Study

Functional Requirements

Compared with the traditional mode, new system can dynamically manage various examination information, including automatic generation of various test papers, automatic calculation of test results and so on. At the same time, it also has a high degree of flexibility, wherever the students are, they can visit the online evaluation system through the browser.

The main goal of the online evaluation system is to realize the system to send the question and examination automatically. After the examination is over, the system can also read the paper under a certain plan according to the reference answer. To achieve this goal, we need to systematically and effectively manage examination questions, automatically generate examination papers, and retrieve results on time

- ✓ Allow users to login to their account
- ✓ Allow users to edit their account
- ✓ Allow users to edit files within the web page
- ✓ Allow instructors and admins to create assessments
- ✓ Allow instructors to edit assessments
- ✓ Allow instructors to open or close assessments
- ✓ Allow students to enroll in assessments
- ✓ Allow students to submit assignment files
- ✓ Allow graders to function as an instructor without the ability to create courses
- ✓ Allow students to view assessments once submitted



Non-Functional Requirements

The non-functional requirements outline the system's expected behaviour or functions. It's also referred to as the system architecture's quality qualities. Non-functional requirements include things like response time, scalability, reliability, maintainability, and usability.

Technical Requirements

The system has to make calls to API and shall handle concurrency issues well. To ensure that the system is working properly, automated tests will be done. For a summary of the test results, a report must be produced. A minimum of 90% of code should be covered by tests.



Libraries and Frameworks

Submit depends on multiple libraries and frameworks. These include Ruby on Rails, Django, Celery, services for networking, and user interface styling. These libraries and frameworks are listed in this section

Rails and Django both are web application development framework running on Ruby and Python language respectively which are designed to make programming web applications easier. They are an open source software and has an elegant and compact design which effectively creates a domain specific language for writing a web application.

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.

Advantages of Django

- a lightweight and standalone web server for development and testing
- a form serialization and validation system that can translate between HTML forms and values suitable for storage in the database
- a template system that utilizes the concept of inheritance borrowed from object-oriented programming
- a caching framework that can use any of several cache methods

Ruby on Rails

• Rails is a full-stack framework. It ships with all the tools needed to build amazing web apps on both the front and back end.



Ruby on Rails, or Rails, is a server-side web application framework written in Ruby under the MIT License. Rails is a model-view-controller (MVC) framework, providing default structures for a database, a web service, and web pages. It encourages and facilitates the use of web standards such as JSON or XML for data transfer and HTML, CSS and JavaScript for user interfacing. In addition to MVC, Rails emphasizes the use of other well-known software engineering patterns and paradigms, including convention over configuration (CoC), don't repeat yourself (DRY), and the active record pattern.

It allows us to write less code while achieving more than many other languages and frameworks

The PostgreSQL database supports multi-threaded and multi-user structured queries (SQL). PostgreSQL is built to manage high-volume systems. The information is kept in organized tables that can be accessed by various threads. The PostgreSQL database was chosen because of its reliability, ease of setup, and use of the standard SQL language. Prior to application creation, we must install PostgreSQL server and client, as well as create tables in the database. Learning SQL queries was made easier with the help of PostgreSQL manuals.

Dynamic Program Analysis

Even if the program is basic, it can be difficult to figure out the outcomes of a manual assessment because the program may have numerous alternative executive paths and output results. Dynamic analysis evaluates a program by running it on test data that is either created automatically or provided manually. In other words, dynamic program analysis is the act of analyzing computer software by running programs created from that software on a real or virtual processor.



The target program must be run with enough test inputs to provide interesting behavior for dynamic program analysis to be effective. Software testing approaches like code coverage ensure that a sufficient portion of the program's collection of possible actions has been seen. In addition, effort must be taken to minimize the impact of instrumentation on the target program's execution (including temporal aspects).



API Requests and Responses

http://127.0.0.1:2358/submissions

Request

```
{
    "source_code": "public class Main{public static void main(String
args[]) {System.out.println(\"hello\");}}",
    "language id": "62",
    "number of runs": "1",
    "stdin": "",
    "expected output": null,
    "cpu_time_limit": "2",
    "cpu_extra_time": "0.5",
    "wall time limit": "5",
    "memory limit": "128000",
    "stack limit": "64000",
    "max processes and or threads": "60",
    "enable per process and thread time limit": false,
    "enable per process and thread memory limit": false,
    "max_file_size": "1024"
}
Response
    "token": "bb89ecf6-9cc9-4a21-bc64-b803261e71c6"
```



http://127.0.0.1:2358/submissions/bb89ecf6-9cc9-4a21-bc64-b803261e71c6

Request:

-

```
Response:
```

```
"stdout": "hello\n",
"time": "0.115",
"memory": 55796,
"stderr": null,
"token": "bb89ecf6-9cc9-4a21-bc64-b803261e71c6",
"compile_output": null,
"message": null,
"status": {
    "id": 3,
    "description": "Accepted"
}
```



Advantages

- The entire process of assigning to evaluation is done online and coding evaluation is done in one click.
- API is used for checking errors in code and give grading.
- Test candidates for programming skills and provides detailed reports on the same
- Increase in number of candidates assessed
- Enables you to upload customized questions
- Reduces manual effort and time
- Create customized tests in minutes



Automated Code Assessment

Assessing a code for education generally means to review it to be able to either assign a mark to it which reflects evaluation criteria or to produce feedback to learners to support their learning. These aspects are not incompatible, as good assessments typically combine both of them, with feedback explaining the mark.

A first aspect to examine is whether programs compile or not. To be more general, the goal is to ensure that a code is syntactically correct. A second level of verification that is sometimes conducted is related to plagiarism. The goal is to be sure that the submitted codes are authentic and have not been copied. A third aspect, a large part of the existing research being focused on it, is code functional correctness. The objective is to verify that the results produced by the execution of assessed codes agree with the expected solution described by any kind of specifications. A fourth level consists of assessing code performance aspects, such as the execution time or the memory consumption. Finally, the last level is related to code quality aspects, including style considerations.

Features

Four key characteristics of automated code assessment systems: Organizing assignments, receiving and storing contributions from students, assisting with automatic or semi-automated assessment, and delivering comments

Security

One of the most significant aspects of automated code assessment systems is that computer security should be prioritized. They are, after all, often running scripts submitted by students, or at least analysis of them, which can be damaging, whether voluntary or not.



Challenges for Automated Code Assessment

Finally, topic focuses on the major obstacles to developing automated code assessment systems. The technological obstacles are the first group of issues that have been recognized.

New procedures and strategies must be developed in order to acquire a better assessment with more valuable and relevant feedback. Only a few research make use of artificial intelligence or big data approaches, which could be due to the domain's infancy and the lack of huge datasets.

Collaboration and Interoperability

The lack of collaborations is another issue with automated code assessment methods. A huge number of platforms are being developed, as the review presented in this study shows.

These platforms are frequently not built to allow the reuse of analysis components in other platforms, which makes collaboration difficult.



System Design

Database design

Users Table

| # | Name | Data Type | Constraint |
|----|--------------|-----------|-------------|
| 1 | Id | Big int | PRIMARY KEY |
| 2 | Username | string | NOT NULL |
| 3 | First Name | string | NOT NULL |
| 4 | Last Name | string | UNIQUE |
| 5 | Email | string | NOT NULL |
| 6 | Password | string | NOT NULL |
| 7 | Is Active | string | NOT NULL |
| 8 | Is Staff | string | NOT NULL |
| 9 | Is Superuser | string | NOT NULL |
| 10 | Date Joined | string | NOT NULL |
| 11 | Last Login | string | NOT NULL |



Question Table

| # | Name | Data Type | Constraint |
|---|-------------------------|-----------|-------------|
| 1 | ld | Big int | PRIMARY KEY |
| 2 | Question Title | string | UNIQUE |
| 3 | Question Description | string | NOT NULL |
| 4 | Default App Code | string | NOT NULL |
| 5 | Default Test Code | string | NOT NULL |
| 6 | Test Code for Java | string | NOT NULL |
| 7 | Test Code for Python | string | NOT NULL |
| 8 | Publish | boolean | NOT NULL |

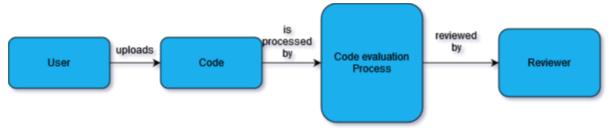


Response Table

| # | Name | Data Type | Constraint |
|----|---------------------|---------------|-------------|
| 1 | Id | Big int | PRIMARY KEY |
| 2 | Language used | string | UNIQUE |
| 3 | App Code | string | NOT NULL |
| 4 | Test Code | string | NOT NULL |
| 5 | Question Id | big int | NOT NULL |
| 6 | User Id | big int | NOT NULL |
| 7 | Can Submit | boolean | NOT NULL |
| 8 | Has Submitted | boolean | NOT NULL |
| 9 | Average Nesting | decimal(5, 2) | - |
| 10 | Maximum Nesting | int(2) | - |
| 11 | Code Commented % | int(2) | - |

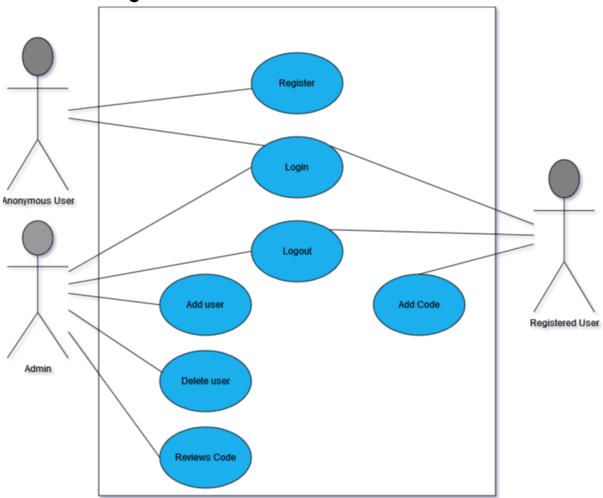


Diagrams



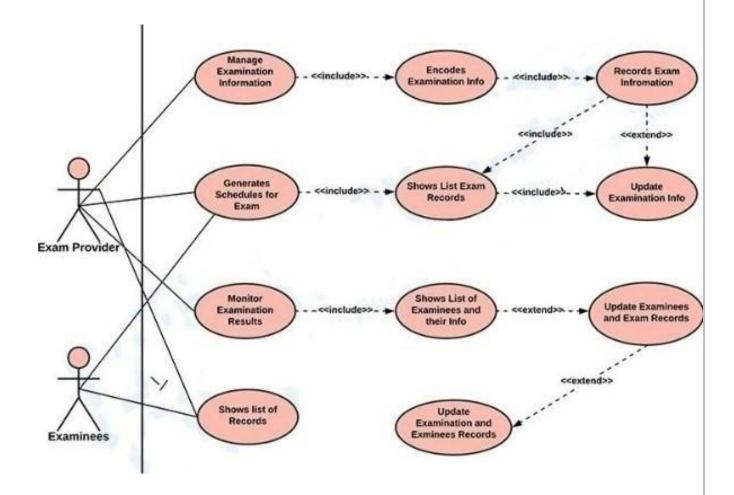
Flow of Application

Use Case Diagram



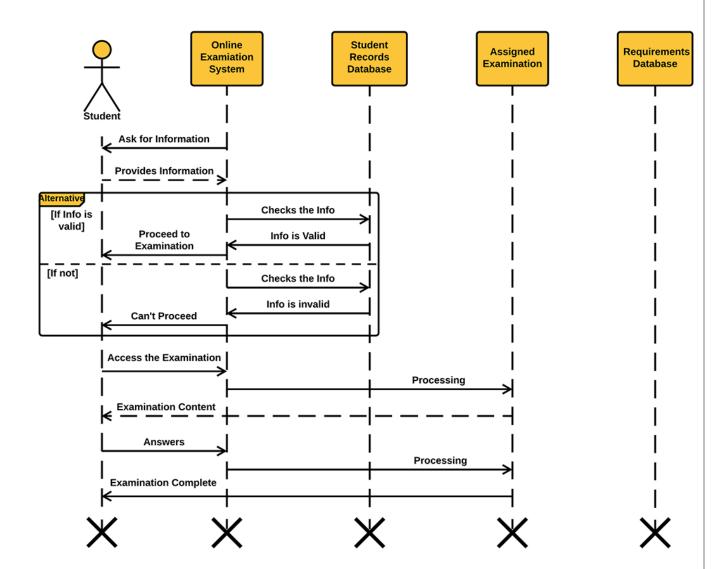


Use Case Diagram



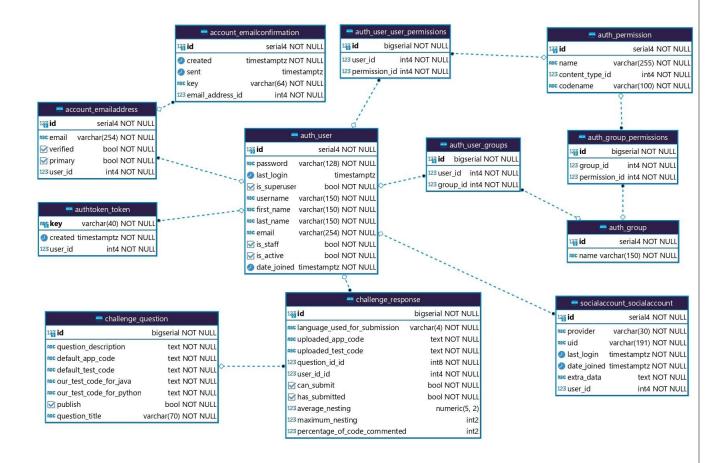


Sequence Diagram



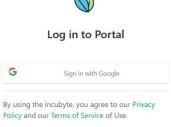


Class Diagram





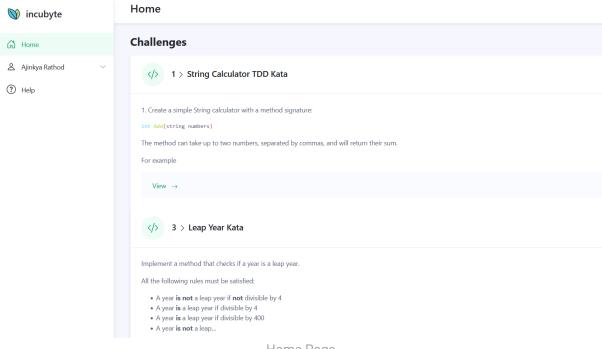
Screenshots



Designed with • by and Incubyte Team.

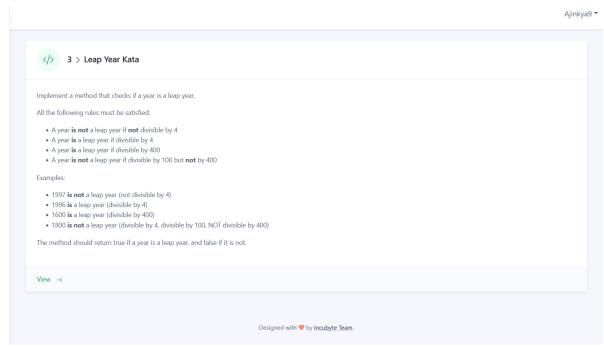


Login Page

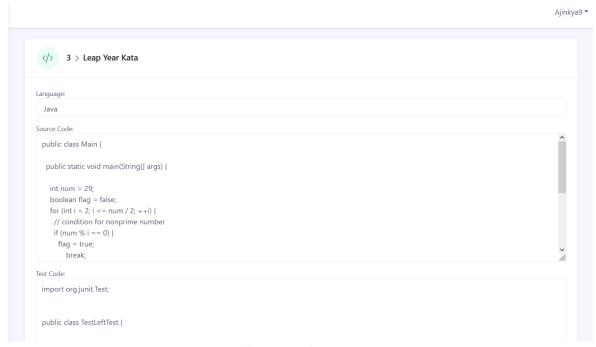


Home Page



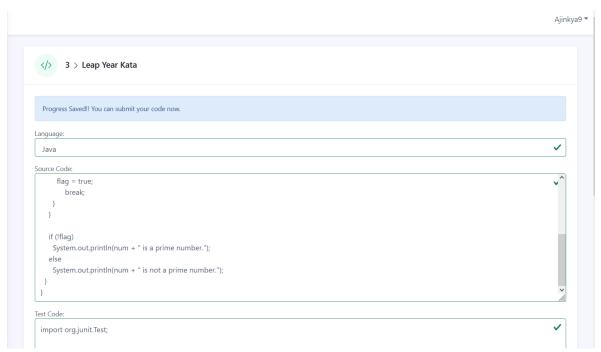


Full Question

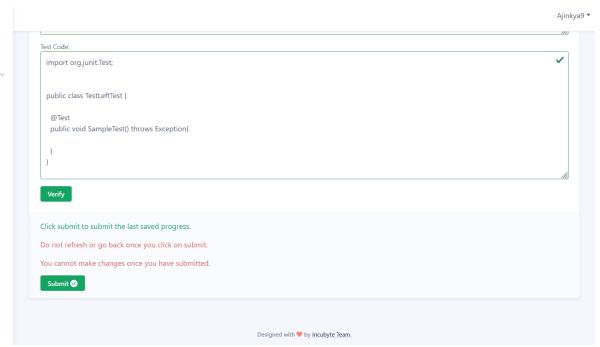


Response Page



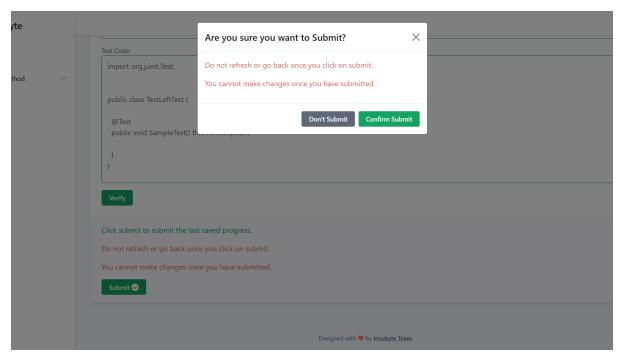


Response With Success

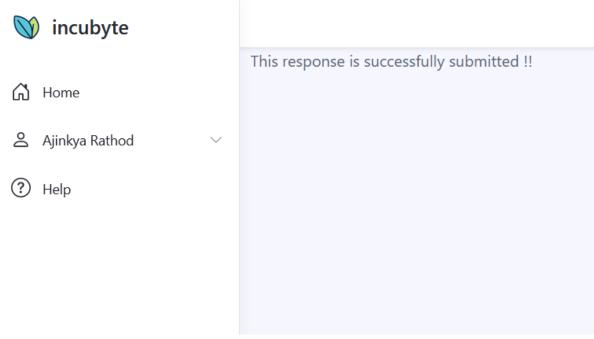


Submit Button Enabled



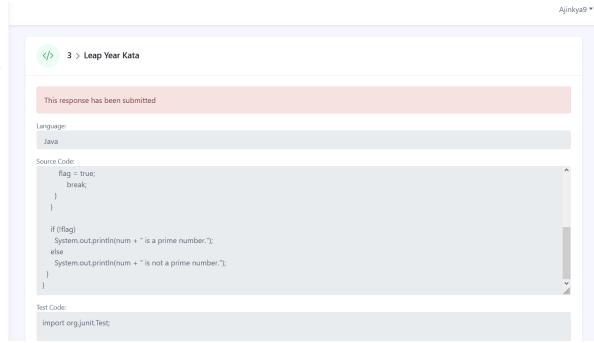


Confirm Submit?

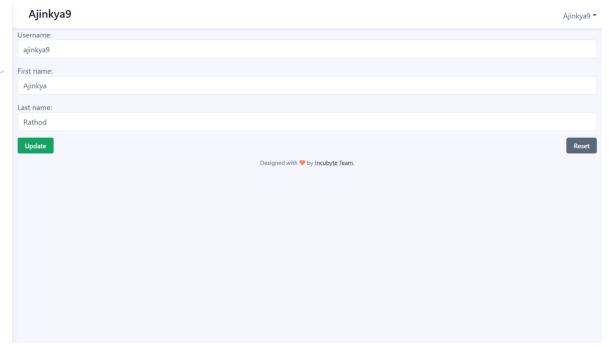


Submission Successful



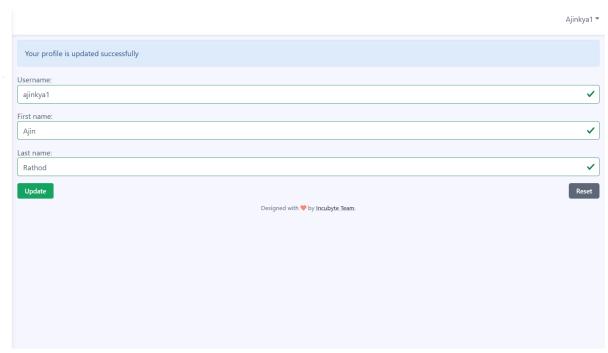


View Submitted code.

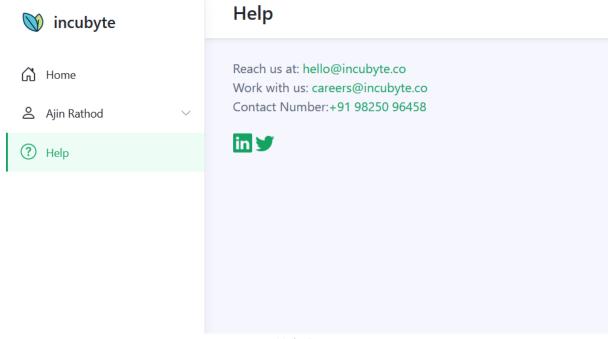


User Details Page



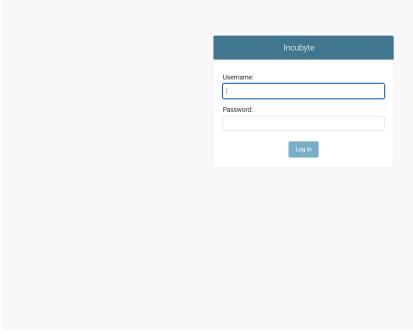


Editing User Details



Help Page



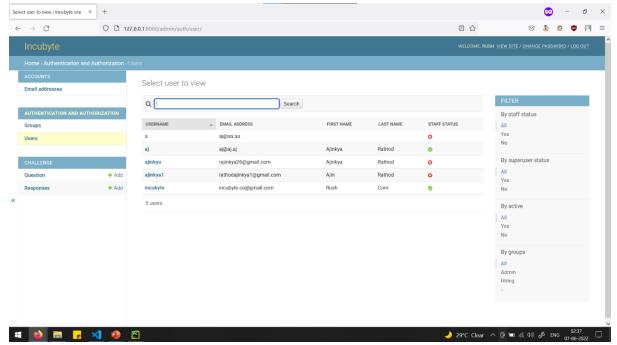


Admin Login Page

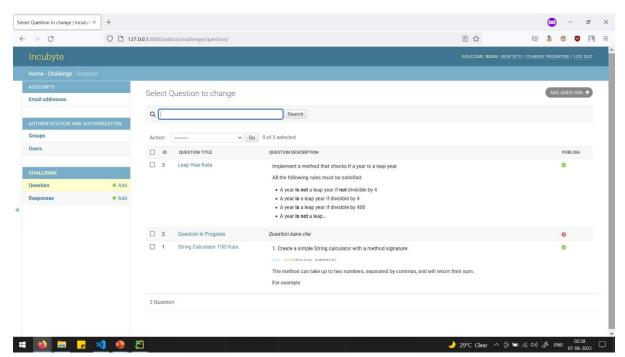


Admin Home Page



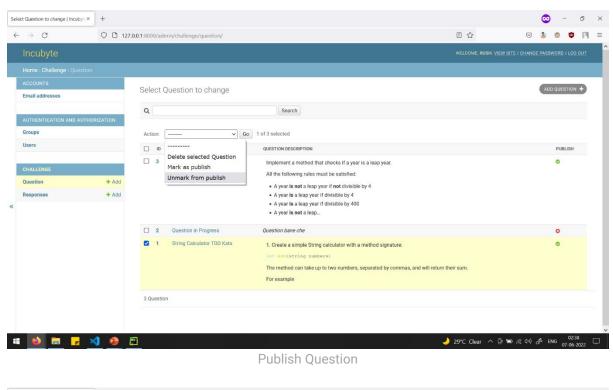


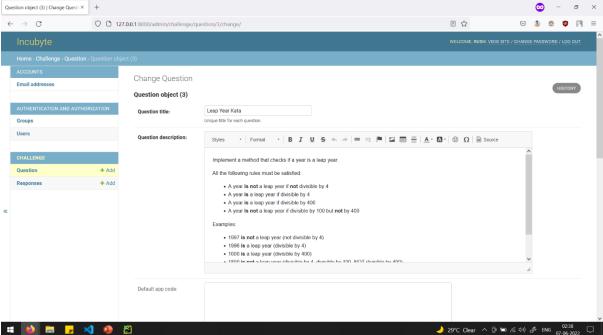
Admin Viewing User Details



Admin Viewing Responses

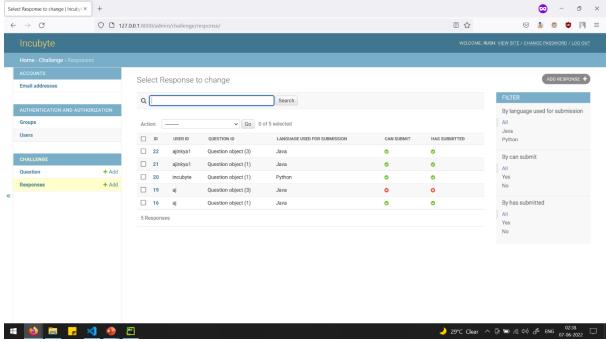




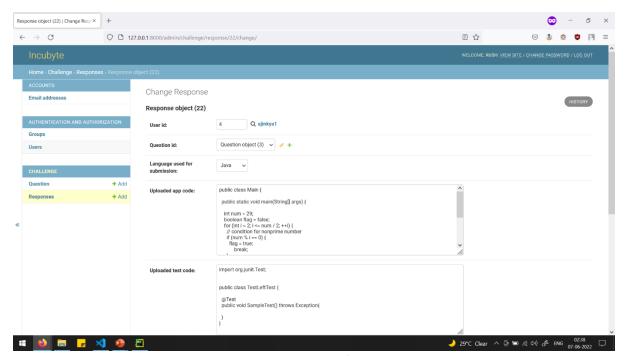


Edit Question just like WYSIWYG



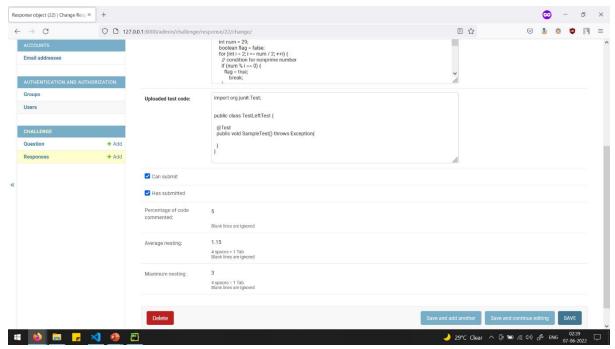


Admin Viewing Responses



Admin Viewing Response of Each Candidate





Checking other User Details like Nesting and Code Commented

Code Editor

```
finclude -lostreams
using namespace std;
the name;
to cout-e-Hello World";

6
)
```

Code Writing Area



Code Editor

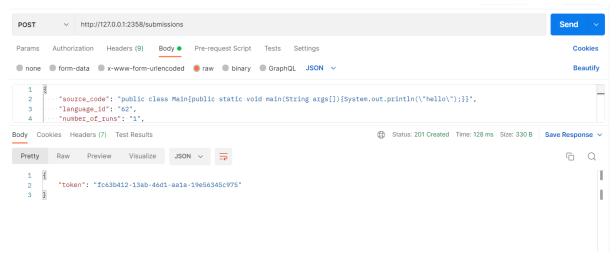
C++

Output of Code

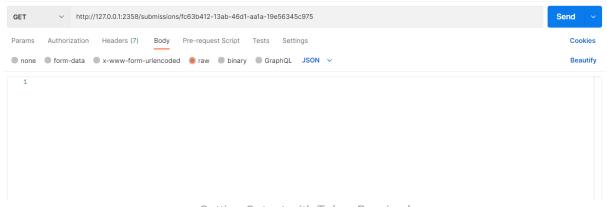
Hello World

API Call in Postman



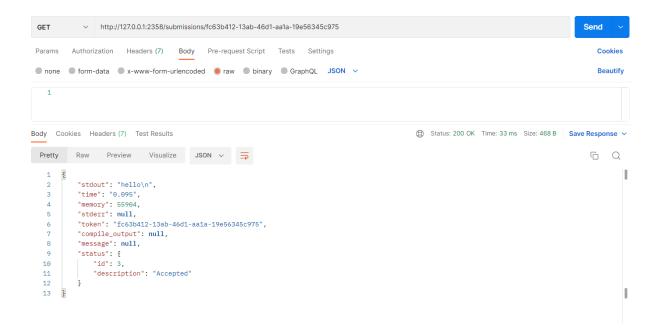


API Call Response with Status 200



Getting Output with Token Received

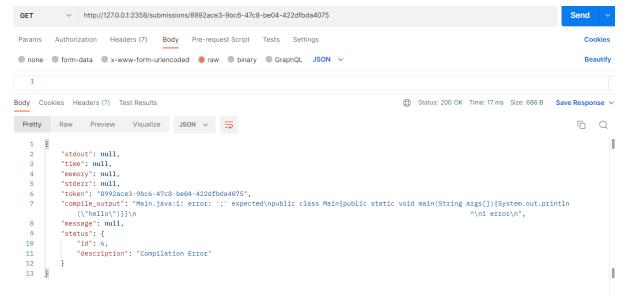




Response of Code Sent



What if Errors?



Error in Source Code API Response



System Testing

Test-driven development (TDD) is a software development process relying on software requirements being converted to test cases before software is fully developed, and tracking all software development by repeatedly testing the software against all test cases.

This is as opposed to software being developed first and test cases created later

The following sequence is based on the book Test-Driven Development by Example:

1. Add a test

The adding of a new feature begins by writing a test that passes if and only if the feature's specifications are met. The developer can discover these specifications by asking about use cases and user stories. A key benefit of test-driven development is that it makes the developer focus on requirements before writing code. This is in contrast with the usual practice, where unit tests are only written after code.

2. Run all tests. The new test should fail for expected reasons

This shows that new code is actually needed for the desired feature. It validates that the test harness is working correctly. It rules out the possibility that the new test is flawed and will always pass.

3. Write the simplest code that passes the new test

Inelegant or hard code is acceptable, as long as it passes the test. The code will be honed anyway in Step 5. No code should be added beyond the tested functionality.

4. All tests should now pass



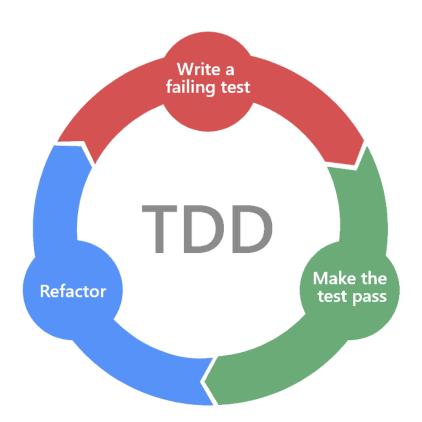
If any fail, the new code must be revised until they pass. This ensures the new code meets the test requirements and does not break existing features.

5. Refactor as needed, using tests after each refactor to ensure that functionality is preserved

Code is refactored for readability and maintainability. In particular, hard-coded test data should be removed. Running the test suite after each refactor helps ensure that no existing functionality is broken.

Examples of refactoring.

- o moving code to where it most logically belongs
- o removing duplicate code
- o making names self-documenting
- o splitting methods into smaller pieces
- o re-arranging inheritance hierarchies





Repeat

The cycle above is repeated for each new piece of functionality. Tests should be small and incremental, and commits made often. That way, if new code fails some tests, the programmer can simply undo or revert rather than debug excessively.

When using external libraries, it is important not to write tests that are so small as to effectively test merely the library itself, unless there is some reason to believe that the library is buggy or not feature-rich enough to serve all the needs of the software under development.

Development Style

Utilizing test-driven development has many benefits, such as adhering to the "keep it simple, stupid" (KISS) and "You aren't gonna need it" philosophies (YAGNI). Designs can frequently be simpler and clearer than those produced by other techniques by concentrating on creating only the code required to pass tests. Kent Beck also proposes the maxim "Fake it till you make it" in Test-Driven Development by Example.

Tests that generate that design are written in order to attain some advanced design concept, such as a design pattern.

The code may remain simpler than the target pattern, but still pass all required tests. Although first uncomfortable, this enables the developer to concentrate solely on what is crucial.

Writing the tests first: The functionality that has to be tested should come before the tests. Numerous advantages have been touted for this. The developers must think about how to test the application from the beginning rather than adding it later, which helps to ensure that it is written for testability. Additionally, it makes certain that tests are written for each feature. Additionally, writing the tests first leads to a deeper and earlier grasp of the product requirements, ensures the effectiveness of the test



code, and keeps a consistent focus on software quality. Developers and companies have a propensity to rush developers who are building feature-first code to the next feature, sometimes even skipping testing entirely.

Because the classes and methods needed by the first TDD test may not yet exist, the test may initially fail to even compile. However, that initial test serves as the start of an executable specification.

Fakes, mocks and integration tests

Fake and mock object methods that return data, ostensibly from a data store or user, can help the test process by always returning the same, realistic data that tests can rely upon. They can also be set into predefined fault modes so that error-handling routines can be developed and reliably tested. In a fault mode, a method may return an invalid, incomplete or null response, or may throw an exception. Fake services other than data stores may also be useful in TDD: A fake encryption service may not, in fact, encrypt the data passed; a fake random number service may always return 1. Fake or mock implementations are examples of dependency injection.

Test doubles are of a number of different types and varying complexities:

- Dummy A dummy is the simplest form of a test double. It facilitates linker time substitution by providing a default return value where required.
- **Stub** A stub adds simplistic logic to a dummy, providing different outputs.
- Spy A spy captures and makes available parameter and state information, publishing accessors to test code for private information allowing for more advanced state validation.



- Mock A mock is specified by an individual test case to validate test-specific behavior, checking parameter values and call sequencing.
- Simulator A simulator is a comprehensive component providing a higher-fidelity approximation of the target capability (the thing being doubled). A simulator typically requires significant additional development effort

Keep the unit small

For TDD, a unit is most commonly defined as a class, or a group of related functions often called a module. Keeping units relatively small is claimed to provide critical benefits, including:

- Reduced debugging effort When test failures are detected, having smaller units aids in tracking down errors.
- Self-documenting tests Small test cases are easier to read and to understand

Individual best practices

Some best practices that a person could adhere to include keeping each test oracle focused on only the results required to validate its test, keeping each test oracle separated into test support services used by the appropriate test cases, and designing time-related tests to allow tolerance for execution in non-real time operating systems. The standard practice of allowing a 5–10% cushion for late execution lowers the likelihood of false negatives during test execution. Additionally, it is advised to treat test and production code equally. The test code must be durable, legible, and maintainable, and it must function successfully in both positive and negative circumstances.



Example of TDD

Here in this Test Driven Development example, we will define a class password. For this class, we will try to satisfy following conditions.

A condition for Password acceptance:

• The password should be between 5 to 10 characters

First in this TDD example, we write the code that fulfills all the above requirements.



Scenario 1: To run the test, we create class PasswordValidator ();

```
package Prac;

public class PasswordValidator {
  public boolean isValid(String Password)
  {
    if (Password.length()>=5 && Password.length()<=10)
    {
      return true;
    }
    else
      return false;
  }
}</pre>
This is main condition checking length of password. If meets return true otherwise false.

}
```

We will run above class TestPassword (); Output is PASSED as shown below;

Output:



Scenario 2: Here we can see in method TestPasswordLength() there is no need of creating an instance of class PasswordValidator. Instance means creating an object of class to refer the members (variables/methods) of that class.

We will remove class PasswordValidator pv = new PasswordValidator () from the code. We can call the isValid () method directly by PasswordValidator. IsValid ("Abc123").

Scenario 3: After refactoring the output shows failed status (see image below) this is because we have removed the instance. So there is no reference to non –static method is Valid ().

So we need to change this method by adding "static" word before Boolean as public static boolean isValid (String password). Refactoring Class PasswordValidator () to remove above error to pass the test.

Output:

After making changes to class PassValidator () if we run the test then the output will be PASSED as shown below.



Future Enhancement

- ✓ We will try to accomplish each and every factor possible to check whether the code submitted is perfect as per requirement or not
- ✓ Applying Machine Learning on Clean Code
- ✓ Check for gibberish or misleading comments
- ✓ Check if proper names are given to variables



Bibliography

Links

- √ https://docs.djangoproject.com/en/4.0/
- √ https://www.nltk.org/

Books

- ✓ Software Craftsman, The: Professionalism, Pragmatism, Pride (Robert C. Martin Series)
- ✓ Clean Code: A Handbook of Agile Software Craftsmanship
- ✓ Clean Coder, The: A Code of Conduct for Professional Programmers (Robert C. Martin Series)

Articles

✓ Hofer, Thomas. Evaluating static source code analysis tools. No. STUDENT. 2010.

