FEASIBILITY STUDY

AUTOMATED PROGRAMMING ASSIGNMENT EVALUATOR

DATE - 2021/07/28

Group No: 20
Project ID: 19
Mentor: Dr. Buddhika Karunarathne

Thilina	180371K
Musab	180377J
Uthpala	180420L

Contributions

- ★ **180371K - Thilina Lakshan**
  - Introduction
    - Objectives
    - Scope
    - Existing systems : WebCAT
    - Technology : RestAPI
    - Technology : Remote Code Execution
  - Feasibility study
    - Technical feasibility
    - Consideration

- ★ **180377J - Musab**
  - Introduction
    - Deliverables
    - High level architecture
    - Existing systems : Algoexpert
    - Technology : Database
  - Feasibility study report
    - Risk feasibility
    - Social/Legal feasibility

- ★ **180420L - Uthpala Nethmini**
  - Introduction
    - Overview
    - Uses
    - Existing systems : Hackerrank
    - Technology : WebUI
    - References
  - Feasibility study report
    - Financial feasibility
    - Resource and time feasibility

Table of Contents

1. Introduction

1.1 Overview of the Project

One of the best ways to teach and learn programming languages and concepts is by practising. Therefore teachers and instructors always focus on evaluating students by giving programming assignments and coding challenges. The main problem with this case is that a large number of assessments have to be evaluated one by one manually. This can be turned into a difficult or even an impossible task and it is a time-wasting approach. If we can introduce an automated system to evaluate and grade programming assignments efficiently, then it would be a great beneficiary for both instructors as well as students.The proposed system will facilitate instructor post assignments with multiple questions with descriptions and relevant test cases. Students can submit answers to assignments. The system will evaluate and grade student submissions. It also provides error checking and plagiarism checking capabilities. Instructors can give feedback to the student submissions individually. The system will make it easier for students to write code, manage their submissions, view their progress, and share with others.

1.2 Objectives of the Project

- Design and implement a user-friendly web interface for users
- Provide the ability to conduct programming assignments securely
- Automate the evaluation process and grading process efficiently
- Provide the ability to check plagiarism among student submissions
- Provide modular architecture to integrate more programming languages
- Provide the ability to view and share relevant reports

1.3 The Need for the Project

- The main focus is to provide a platform for instructors to post, evaluate, and grade assignments and provide feedback to their students efficiently.
- The system can be used to conduct online programming competitions as well.
- Subsystems like Remote Code Execution, Plagiarism Checker can work as individual systems and can be used to build any other system.

1.4.1 Overview of Existing Systems

Web-CAT

Web-CAT is an advanced automated grading system for testing how students test their code. It is free and open-source software. It has a plugin-in-style architecture to provide additional features. It supports many models of grading, assignments, and feedback generations. The main focus of Web-CAT is to test the quality of the test case which is written by students for their code.[1][2][3]

Hackerrank

HackerRank is a platform that focuses on competitive programming challenges for both consumers and businesses, where developers compete by trying to program according to provided specifications. Hackerrank provides many facilities like practicing various programming languages, conducting online programming competitions.[4]

Algoexpert

Algoexpert is one of the newest leading interview preparation platforms for software engineers and programmers. It provides some modern features like explaining solutions to coding challenges using video tutorials.[5][6]

1.4.2 Technologies

WebUI

Single page web application (SPA) using VueJs with Vuetify. VueJs is Progressive JavaScript Framework which is used to build reactive web applications. Vuetify is a UI Framework which is built on top of VueJs. (*https://vuejs.org*, *https://vuetifyjs.com*)

Backend API

Rest API using spring framework along with Java 11. The Spring Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications. (*https://spring.io/projects/spring-framework*)

Runtimes (Remote Code Execution)

Separate servers (microservices) will be used to execute source codes. Relevant programming environments and languages will be used for those. Ex: Remote Code Execution server for python programming language will be implemented using Python language or Java language.
(*https://microservices.io*, *https://www.python.org*,
https://en.wikipedia.org/wiki/Java_(programming_language))

Backend Database

MongoDB or Postgresql databases will be used. MongoDB is an open source document oriented database. Postgresql is an open source advanced relational database. (*https://www.mongodb.com*, *https://www.postgresql.org*)

1.5 Scope of the Project

Instructor

- Create and post programming assignments
- Create, add, edit questions in a assignment
- Share assignment with group of students
- View progress of students (reports)
- Give feedback to students

Student

- Submit answers to an assignment
- Write code (in preferable lang.) and test result
- Share results and achievements with others

1.6 Deliverables

- Web based user interface for instructors and students.
- Web based administration panel for system admins.
- Backend API (which needs to be hosted separately.)

2. Feasibility Study

2.1 Financial Feasibility

All the technologies, plug-ins,libraries and APIs used in the proposed system are free and open source. So there will be no expense for the licence. Proposed database is also free and open source and the only cost for the database is hosting cost. When we considered the plagiarism checking part of the system we decided to use internal similarity checking. so there will be no cost for the plagiarism checking API.
In addition, all developers have programming devices and programming tools which are essential to the development are available in the devices and other necessary services freely available on the internet.

Therefore, the system has the cost only for web hosting and database hosting when deploying the system. A summary of projected costs per year as follows.

No.	**Cost Item**	**Cost per Year**
1	Programming Equipment	0.0
2	Programming Tools	0.0
3	Software License	0.0
4	Web Hosting	
	Front-End	$270
	Back-End	$780
	Database	$400
	Total Cost	$1450

In conclusion, it was determined that the project is financially feasible.

2.2 Technical Feasibility

After considering all kinds of risks that could happen in the development period and deployment period, it can be concluded that the project is risk wise feasible.

2.2.1 Frontend

Project is a web-based application. It can be divided into two major parts: backend and frontend. The frontend is the web application that will be developed using HTML, CSS, and Javascript. The backend will be developed using Java-based technologies.

The front-end will be developed using VueJs (https://vuejs.org). It is a modern, progressive, free, and open-source javascript framework that is used to develop reactive web applications. VuetifyJs (https://vuetifyjs.com) will be used as the component framework (for VueJs). VuetifyJs is known as a rapid development framework.

2.2.2 Backend

The backend consists of an API gateway, a Remote code execution service for each programming language, a Plagiarism checker, and a Database. Since the proposed system needs to be extensible, robust, and secure, the Spring framework (https://spring.io) with Java-based technologies matches best. Primarily the Postgresql (https://www.postgresql.org) database will be used as the persistent data storage but since the spring framework provides a clean abstraction for the data layer therefore the database can be changed to any other solution without much work. Ex: (MongoDB). All backend technologies mentioned are free and open-source.

2.2.3 Deployment

Deliverables for the frontend is a static web application that can be hosted using any static content server. The backend will be delivered as Docker images which can be hosted on most of the hosting services easily.

2.2.4 Development

For development, debugging, and testing will be done using tools such as Visual Studio Code, IntelliJ Idea, DBeaver, Insomnia will be used. All the mentioned tools are freely available. Thus it is clear the project is technically feasible.

2.2.5 Conclusion

After considering all kinds of technologies that will be used in the development period and deployment period, it can be concluded that the project is technology wise feasible.

2.3 Resource and Time Feasibility

2.3.1 Resource Feasibility

when we discuss the resources we can analyze it under a few categories as follows.

1. Programming Device (Laptops - already available)
2. Hosting Space
3. Programming Tools (All the technologies associated with the project are freely available)
4. Programming Individuals (project can be divided into 3 main modules and each person can be responsible for each module)

Therefore it is clear that the project has required resource feasibility.

2.3.2 Time Feasibility

The project time schedule has already been prepared and according to that there is enough time to complete the project. Therefore the project is feasible with time.

2.3.3 Conclusion

After considering resources and time which are needed for development and testing, it can be concluded that the project is feasible in terms of resources and time.

2.4 Risk Feasibility

2.4.1 Business Impact Risks

Revenue : This can be used as a standalone application or a plugin, thus it helps existing systems which need these facilities or organizations looking for this solution , thus it can generate revenue. But it is not possible to pay a huge amount of money in the initial stages, so we can make it open source and get support.

Deadline : As this has to be done within 14 weeks , we have to take the risk to work as planned,if there are any obstacles (eg: exams , developer unavailability) we will not be able to develop the extensions/plugin support.

Users : As the project has to be completed within 14 weeks we will not be able to get the opinions from a set of users. So we will have to be confined to the opinion of the three developers.

Costs : As the initial cost for hosting will be risky as students, we will have to go for free less performance services.

2.4.2 Customer Related Risks

As the customers will need different needs(for example , someone may need language that is not supported in our system) , we will have to work on that as the customers expand.

2.4.3 Development Environment Risks

Due to the current situation in our country , it seems impossible to have physical meetings, thus having group discussions on online platforms and working together is a bit of a risky thing.

2.4.3 Technology Risks

All the technologies we are going to use are well established technologies, and regarding the evaluation we will only evaluate the answers for correct or false , we will not be able to give partial answers.

2.4.4 Technical Issue Risks

Software code and code documentation will be freely available.

2.4.5 Conclusion

After considering all kinds of risks that could happen in the development period and deployment period, it can be concluded that the project is risk wise feasible.

2.5 Social/Legal Feasibility

The proposed system is going to use publicly available softwares tools for development, there will be no issues on development and license.

Since this new system creates a platform for students and instructors to work collaboratively, it will give a great impact in the programming learning section. Those features can be used to market this system and attract users from other existing systems. The system has to be made user friendly and with all the easiness to quickly get started . Thus the system is socially and legally feasible.

2.6 Considerations

2.6.1 Performance

Frontend and backend communication requires low bandwidth but there needs to be a considerable amount of processing power at the backend server because evaluating student submission and plagiarism checking requires a considerable amount of CPU load. Therefore If the number of users increases over time there need to be more instances for each remote code execution service and plagiarism checking service.

Response time to take from source code submission to showing output will depend on the code complexity but the system will add an upper limit for that. The best matching value for the limit is yet to be chosen while testing the system. Also, the time taken to generate a plagiarism report will be dependent on the number of submissions and the size of the submission. Other Than that response time for user events will be below 2 sec.

Throughput and Storage performance for the system is yet to be determined.

2.6.2 Security

User authentication is done by validating username and password. All the communication between the frontend and backend will be done using HTTPS protocol. Resetting the user password will be possible by verifying the user using the email address or the user can contact a system administrator.

The authorization will be done by using JWT [7] and its store in browser local storage with additional encryption. Each user will only be allowed to access the content that is allowed for them.

2.6.3 Usability and ease of use

Users will be provided with a complete user manual for the system. The user interfaces are designed to make it easy to use without any third-party guidance.

2.3.4 Capacity and Scalability

The system is built with extensibility in mind therefore capacity of the system can be easily increased by adding more microservices. Since the backend provides a data API the system can integrate with more clients. Ex: mobile application, Moodle plugin

2.3.5 Availability

The system will be available throughout the 24 hours per day. Meantime to failure or system upgrade will be decided according to needs. With good hosting space, availability can be guaranteed even while upgrading the system.

2.3.6 Maintainability

The system is designed using microservices architecture therefore it is highly maintainable.

3. References

[1] Web-cat.org. 2021. *Web-CAT: the Web-based Center for Automated Testing - Web-CAT*. [online] Available at: <https://web-cat.org/> [Accessed 24 July 2021].

[2] 2019. [video] Available at: <https://www.youtube.com/watch?v=jV202f4Xc_M&t=5s> [Accessed 24 July 2021].

[3] Web-cat.github.io. 2021. *What is Web-CAT? - Web-CAT*. [online] Available at: <https://web-cat.github.io/projects/Web-CAT/WhatIsWebCat.html> [Accessed 24 July 2021].

[4] HackerRank. 2021. *Developer Skills Platform - HackerRank*. [online] Available at: <https://www.hackerrank.com/products/developer-skills-platform/> [Accessed 24 July 2021].

[5] Algoexpert.io. 2021. *AlgoExpert | Ace the Coding Interviews*. [online] Available at: <https://www.algoexpert.io/product> [Accessed 24 July 2021].

[6] RealToughCandy. 2021. *Is AlgoExpert worth it in 2021? [Algoexpert.io review] - RealToughCandy*. [online] Available at: <https://realtoughcandy.com/is-algoexpert-worth-it/#:~:text=What%20is%20AlgoExpert%3F,also%20has%20expansive%20video%20solutions> [Accessed 24 July 2021].

[7] En.wikipedia.org. 2021. *JSON Web Token - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/JSON_Web_Token> [Accessed 27 July 2021].