An Automated Proctoring System for Online Examination

by

Jewel Mahmud Nimul Shamim

A Report submitted to the

School of Graduate Studies

in partial fulfilment of the

requirements for the degree of

Master of *Science*

Supervisor: Dr. Edward Brown

Department of *Computer Science*

Memorial University of Newfoundland

August 2020

St. John's Newfoundland and Labrador, Canada

Abstract

The online exam system has grown popularity over countries because of its adaptability, user-friendliness, and candidates' ability to take the exam remotely. According to the research community, the major challenge in the online exam is the proctoring technique. Since there is no rule-based proctoring system, it opens new opportunities for defrauding and accessing internet resources for reference during an exam. Some of the available proctoring systems offer to stop these activities by keeping eyes on test-takers through a camera, microphone, or applying machine learning and image processing techniques by analyzing face features points using facial recognition and head posture.

Ironically, none of the proctoring systems monitor the students' network traffic (inbound and outbound) contents, which they access, and this motivates us to question what sort of proctoring application can automate the proctoring process using some predefined rules and observe traffic contents during an examination.

In this project, our developed proctoring system offers a novel way to automate the proctoring process and detect fraudulent activity during an online exam by intercepting network inbound and outbound traffic content using MITMproxy and analyze it with Natural Language Processing. To test our application's feasibility, we applied several flexible proctoring rules and sample exam questions to enforce proctoring rules using content analysis to limit the access resources. In most testing situations, the proctoring application performs better if the exam question is a sequence of words or a sentence rather than complicated mathematical or chemical reaction related questions.

Acknowledgements

I would like to express my gratitude to Dr Edward Brown, my project supervisor, for his patient guidance, encouragement, and useful critique of this research work.

Table of Contents

Abstract ii

Acknowledgements iii

List of Figures v

1 Introduction 7

2 Background 8

2.1 Multi-modal online proctoring system 8
2.2 Heuristic-based automatic proctoring system 9
2.3 A Design of Continuous User Verification 10
2.4 ProctorU 11
2.5 Mettl 12

3 Architecture 13

3.1 System design 13
3.2 Student monitoring application 16
3.3 Proctor application 18
3.4 OpenVPN 20
3.5 MITMproxy 21
3.6 Natural language processing using Fuzzy string matching 22
3.6.1 Levenshtein distance metric algorithm 22
3.7 Database schema diagram 23
3.8 RESTful API 24

4 Testing 25

4.1 Content analysis with proctoring rules 25
4.2 Optional Features 30
4.2.1 Networking monitoring during the exam 30
4.2.2 Block specific IP/URL 30

5 Results 32

5.1 Proctoring rules and features 32
5.2 Content analysis 35
5.3 Limitations 36

6 Conclusion and future work 37

References 39

List of Figures

Figure 1: Multi-modal proctoring system architecture. (Prathish, S., & Bijlani, 2016).8
Figure 2: Multi-modal online proctoring system architecture. (Raj, Narayanan, & Bijlani, 2015). 9
Figure 3: Convolution process in CNN (Asep & Bandung, 2019). 10
Figure 4: Continuous user verification architecture (Asep & Bandung, 2019). 11
Figure 5: Flowchart of ProctorU systems (ProctorU, n.d.). 12
Figure 6: Prototype of the proposed automated proctoring system. 13
Figure 7: System architecture of the developed proctoring system. 14
Figure 8: Design of the proctoring System. 15
Figure 9: Package Diagram of Proctoring System. 16
Figure 10: Student permission section. 17
Figure 11: Student login section. 17
Figure 12: Accessing students network interface information. 18
Figure 13: Proctor web application login page. 19
Figure 14: Proctoring rules setting section. 19
Figure 15: Proctoring rules and monitoring section. 20
Figure 16: How open VPN works. 21
Figure 17: How MITMproxy works. 21
Figure 18: Intercepted network inbound and outbound traffic by MITMproxy. 22
Figure 19: Levenshtein distance metric algorithm. 23
Figure 20: Database Enhanced entity-relationship diagram. 24
Figure 21: Visual representation of the RESTful API works in the proctor system...24
Figure 22: Proctoring rules setting section. 25
Figure 23: Software engineering question to test the proctor application. 26
Figure 24: Searching with question. 27
Figure 25: Application block the content. 27
Figure 26: Search result from google. 28
Figure 27: Content is blocked by proctor application. 28
Figure 28: Searching result from google. 28
Figure 29: Traffic content matches with more than 5 question keywords. 28
Figure 30: Student tries to submit the question to Stack Overflow. 29
Figure 31: Block submission using traffic content analysis. 29
Figure 32: Student application monitor students network interface. 30
Figure 33: Block specific IP/URLs. 31
Figure 34: An Algebra sample question to test the proctoring application. 32
Figure 35: The proctoring application block the content which match the question...33
Figure 36: A chemistry sample question to test the proctoring application. 34
Figure 37: A student search with the chemistry exam question (1). 35
Figure 38: The proctor application blocks the content. 35

Figure 39: A student search with the chemistry exam question (2)............................35
Figure 40: The proctor application cannot block the content. 35
Figure 41: Mathematical equations are in image format on a website. 36

1 Introduction

Modern Web technologies include an innovative dimension in examinations. Online examination methods have become more simplified with the innovative process of Massive Open Online Courses (MOOCs). Before exams meant using pen, paper, and proctor but can now be conducted with the computer and the internet, which helps students take the exam remotely without any designated place. Virtual proctoring allows students to write a test online remotely while keeping the examination's integrity. Although online testing has been around in different ways for the past 20 years, in the current scenario, due to this travel ban and lockout in multiple countries, online proctoring has become more popular.

Online examination also introduces a new form of cheating or impersonation. Students either use someone else to take the exam or use smart devices (mobile phones, tablets) for solutions on the internet. Open-book exams are also becoming difficult to conduct because students spend too much time searching for answers rather than solving the exam problems. ProctorU, Mettl, are example of proctoring systems in the market. These proctoring systems stop cheating by keeping eyes on test-takers via camera, microphone, or active window monitoring feature. A recently proposed proctor system prevents cheating by analyzing face feature points using face recognition and head posture using machine learning and image processing techniques (Raj, Narayanan, & Bijlani, 2015). But these features are not complete since none of the proctoring systems can set rules or monitor students to network traffic content using content analysis and prevent them from accessing restricted content during the exam. The proctoring rules and monitoring network traffic content are essential because they need to know what resources students are accessing during the exam.

Furthermore, proctoring rules need to be flexible, so that the proctor may change the rules depending on the exam. The rules may limit students' access to the answer or restrict their internet search during exams. Also, some students post the question on the internet for clarification during the test. Interestingly, none of the proctoring applications in the online proctoring program manages these types of scenarios. Such rules and monitoring process are beneficial, in addition to monitoring through webcam and the active window monitoring function. Our proposed proctoring application has four different sections, which will monitor the candidates: a) Network inbound and outbound traffic, b) System information c) Video analysis via webcam d) Audio analysis via microphone.

In this study, we develop a proctoring system that automates the proctoring process by defining flexible proctoring rules, monitoring students' inbound, and outbound traffic content during an online exam—also restricting online resources. The implemented proctoring system is also focused on observing a) Network inbound and outbound traffic, and b) System information. The developed proctoring system has a desktop student application and web-based proctor application. The student desktop application allows students to connect to the exam server during an exam. The web-based proctor application is for monitoring students, setting rules, and restricting resources during an exam. On the server-side, the proctoring application uses the MITMproxy library to intercept all HTTP and HTTPs network inbound and outbound traffic content and analyze these intercepted contents using Natural Language processing. The entire process we develop is unique. Finally, we tested our proposed proctoring application with four different rules. As none of the existing proctoring system have an option to set rules for the exam, we tried to make our proctoring rules very flexible. In most testing situations, the proctoring application performs better if the exam question is a sequence of words or a sentence rather than mathematical or chemical reaction related questions.

2 Background

2.1 Multi-modal online proctoring system

There are many popular and commercial online proctoring systems available to control online test likes Kryterion, ProctorU, and Software secure. (Prathish, S., & Bijlani, 2016) Researchers are also trying to improve solutions to create an automated system to handle proctoring difficulties. Some of the proctoring applications only focus on advanced machine learning and face detection technologies in an attempt to present accurate and reliable exam proctoring that beat human proctors. Swathi Prathish, Athi Narayanan, and Kamal Bijlani proposed a multi-modal system that avoids physical attendance during the exam. The inference system captures the audio and video using the webcam and the active window. With the webcam, the system monitors the examinee's head posture by analyzing face feature points (Prathish, S., & Bijlani, 2016).

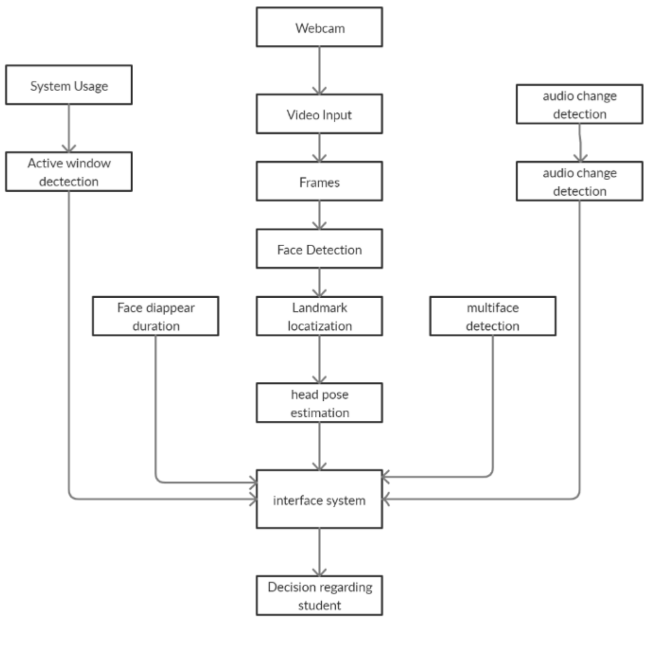


Figure 1: Multi-modal proctoring system architecture. (Prathish, S., & Bijlani, 2016).

The inference system uses the webcam to track the facial movements of the candidate along with audio data (Prathish, S., & Bijlani, 2016). The system will detect the occurrences of malicious actions using these audio data. The video input is separated into several frame rates and analyses of the feature points and estimates the head posture using these framerates (Cheung, 2015). Misconduct can be detected using yaw angle variations, audio presence, and active window monitoring if the candidate is trying to connect any device through USB that is also recognized by the process capture (Prathish, S., & Bijlani, 2016).

However, the system is only working for one to one exam sessions, which is a significant flaw in the proctoring process. The proctor must monitor students via webcam continuously. If there are more than 20 students in the class, this system is difficult and challenging for supervising students during the exam.

2.2 Heuristic-based automatic proctoring system

A team from India proposed a proctoring system following the multi-modal method combination of image and audio processing and monitoring the computer throughout the exam. N.L Clarke, P.Dowland & S.M. Furnell proposed a method to remotely proctor students throughout the exam using transparent authentication and verifying the identity of the candidate (Clarke, Dowland, & Furnell, 2013).

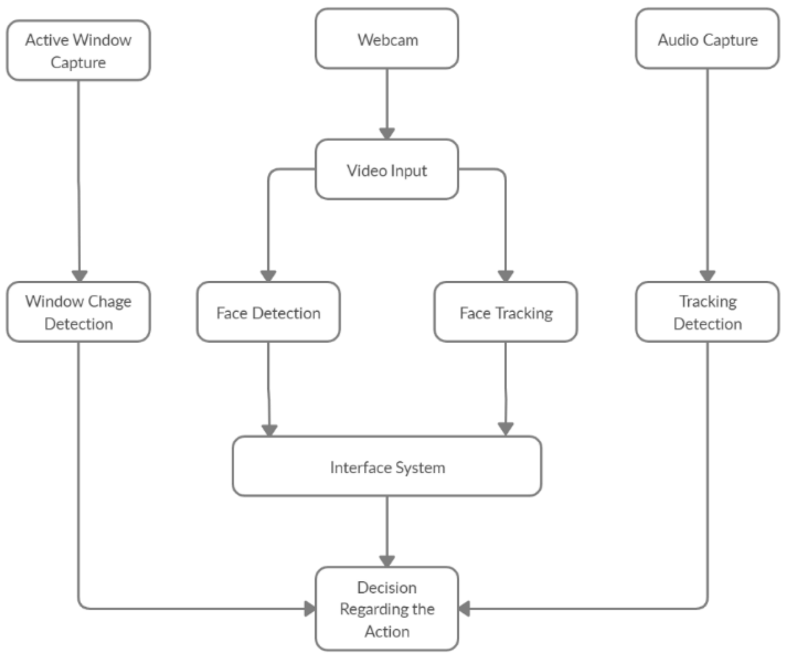


Figure 2: Multi-modal online proctoring system architecture. (Raj, Narayanan, & Bijlani, 2015).

In Vishnu Raj, Athi Narayanan and Kamal Bijlani proposed system consisting of checking the whole room and detecting the face of the student (Raj, Narayanan, & Bijlani, 2015). It will also specify whether unwanted people are talking inside the room during the examination. The active windows feature will detect if the students try to browse on the internet or copy from any files. By analyzing the data, the system determines whether there is any violation or not.

The heuristic-based automatic proctoring system has the same flaw: it only works in one to one exam session; when the number of students increases, this system design approach does not serve anymore because of this limitation. Moreover, the proctoring process will be complicated because it is impossible to keep eyes on more students during an online exam.

2.3 A Design of Continuous User Verification

Face verification requires an online exam session. However, the problem for the face recognition section is system robustness for pose and lighting variation. That is why face identification is significant for the online proctoring system. The proposed method by Asep Hadian and Yoanes Bandung to reduce cheating by a visual confirmation for the entire exam addresses this problem (Asep & Bandung, 2019). The system uses the training data set collected from individual m-learning online lecture sessions by using incremental training to improve algorithm robustness. CNN uses a filter to identify the features of the image. The convolution process will produce a high value when an element exists in the region; otherwise, it is low.

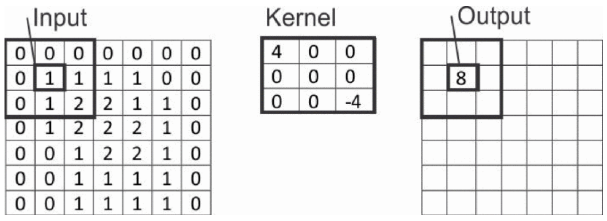


Figure 3: Convolution process in CNN (Asep & Bandung, 2019).

The online exam proctoring system consists of two different parts. 1) Online lecture session, which has a) registration; b) data collection and c) dataset training. At the same time, the second section has two modules: Face detection and Face Verification (Asep & Bandung, 2019). The verification process uses the CNN methods to verify a user by face recognition and continuous user verification until the user finishes the exam (Guo, yu, & yao, 2008).

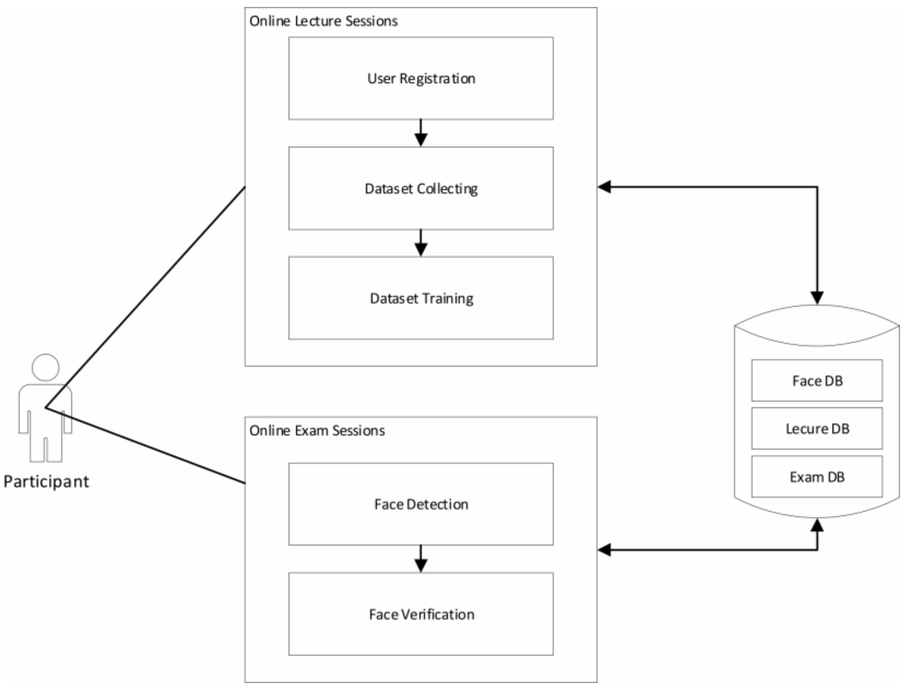


Figure 4: Continuous user verification architecture (Asep & Bandung, 2019).

Since the proposed proctoring system does not monitor student network traffic content, it cannot track what the students are browsing or searching during the exam. Moreover, there is no option to set exam rules for the individual exam.

2.4 ProctorU

One of the most popular online proctoring systems is ProctorU that allows students to complete their exams at home. It is very convenient for students who are taking courses remotely. ProctorU requires a webcam, microphone, and internet connection to take students' exams. ProctorU works by scheduling an appointment for taking the exam and connecting it with a proctor who will verify the candidate's identity. After that, it will follow the exam process, and the proctor can also monitor the exam taker screen. The ProctorU authorities are responsible for solving any technical difficulties that happen during the exam session.

ProctorU has some requirements to follow during the exam session:

1. Authenticate the user identity.
2. Monitor user activity via webcam.
3. Exams must be taken in a secure place. (Libraries or coffeehouse are not acceptable)
4. Cellphones and taking a break during the exam are not allowed.

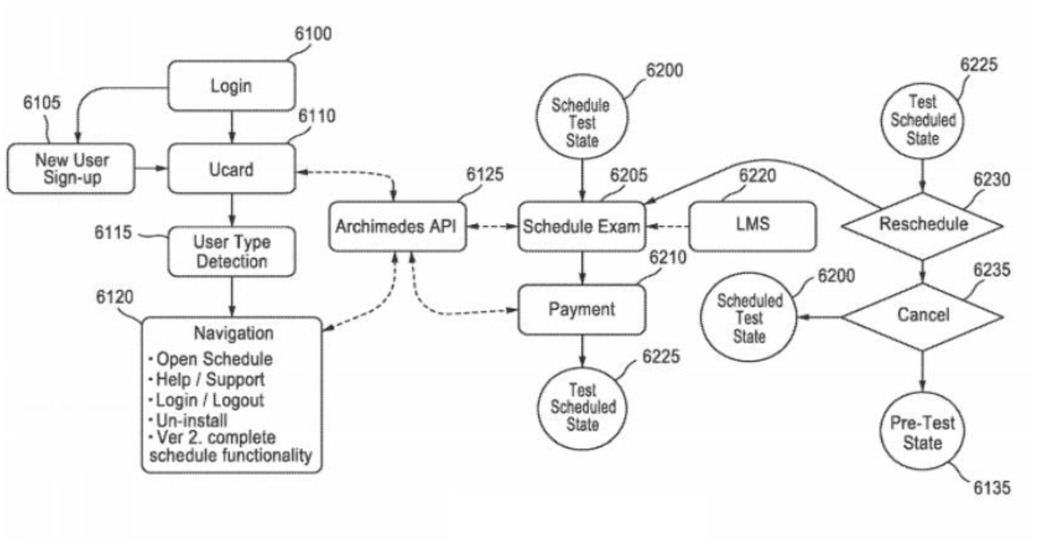


Figure 5: Flowchart of ProctorU systems (ProctorU, 2020).

2.5 Mettl

Mettl is an online system that helps preserve academic integrity & has allowed to conduct exams. Mettl also provides some features for online proctoring, which has pre-authentication using facial recognition. Besides authentication, Mettl has an active window detection system that can also detect multiple tabs open in the browser. It also prevents mobile phone use during exams using advanced image processing technology to identify the device.

Live online proctoring:

Live online proctoring monitors the audio-visual and screen sharing in real-time. A live proctor invigilates every candidate on an online platform during the test. Proctor has permission to disable the test if any cheating happens during the exam.

Advance automated proctoring:

Automated proctoring is an advanced proctoring system which provides audio-visual, and screen monitoring/sharing of the examines during the test. Besides, it monitors the feeds using advanced audio-video analytics to identify any suspicious activity throughout the exam.

In this background section, we observed that most of the proctoring applications have almost the same features. None of these proctoring systems monitors students' network traffic content or sets any specific traffic content rules for the exam. Most of these proctoring systems only support for one-to-one interviews or exam sessions. With these systems, it is challenging to proctor more than 20 students. Furthermore, none of these proctoring applications has an option to set rules by the proctor. As the proctoring process changes depending on the exam, these existing proctoring systems do not serve all types of scenarios.

3 Architecture

3.1 System design

In the background section, we analyzed several existing proctoring systems, and we concluded none of the existing proctoring systems directly monitor candidates' network inbound and outbound traffic content during an exam. In this section, we design an original prototype of a proctoring application, including and extending the existing proctoring features. We divide the proctoring application into four different sections, which will monitor candidates: a) Network inbound and outbound traffic, b) System information c) Video analysis via webcam d) Audio analysis via microphone Figure 6.

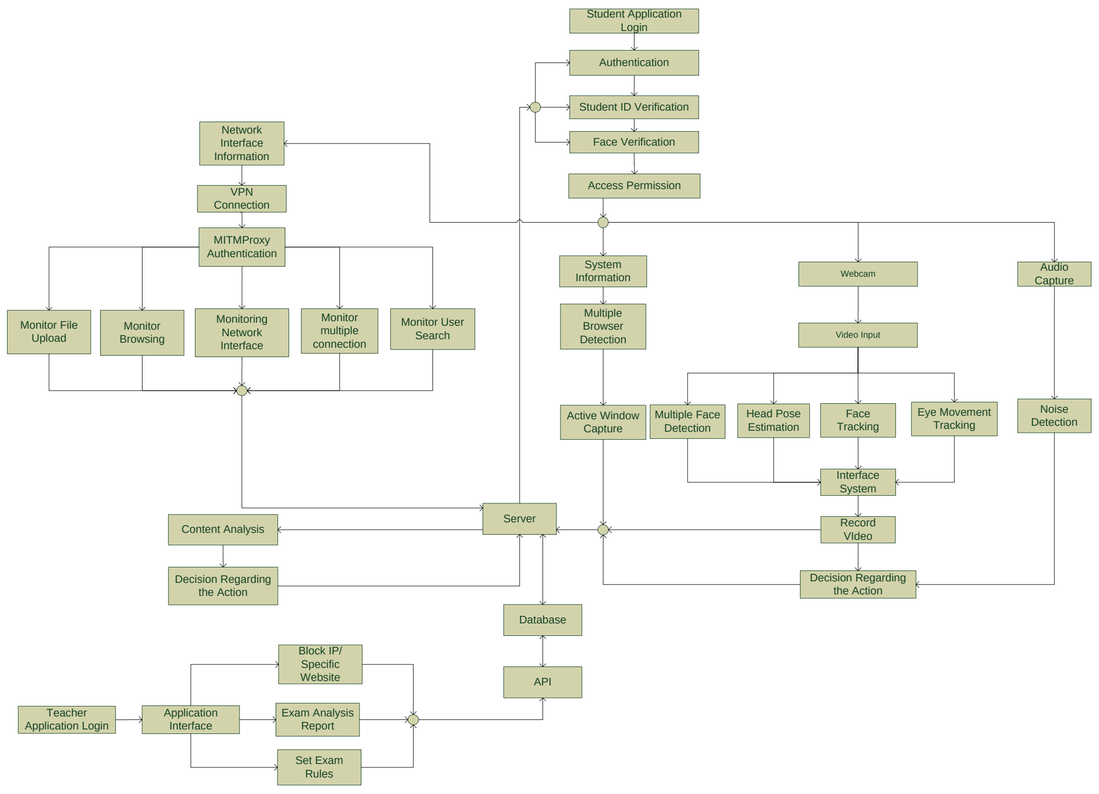


Figure 6: Prototype of the proposed automated proctoring system.

In this project, the developed proctoring system automates the proctoring process by specifying flexible proctoring rules, monitoring students' inbound and outbound network traffic content of the students during an online exam and restricting online resources. The proctoring student application also monitors students' network interface information, which prevents students from split-tunneling. The proctor imports the questions and sets rules for the exams through the proctoring application, which is stored on the database. The server fetches those rules and activates them during the exam.

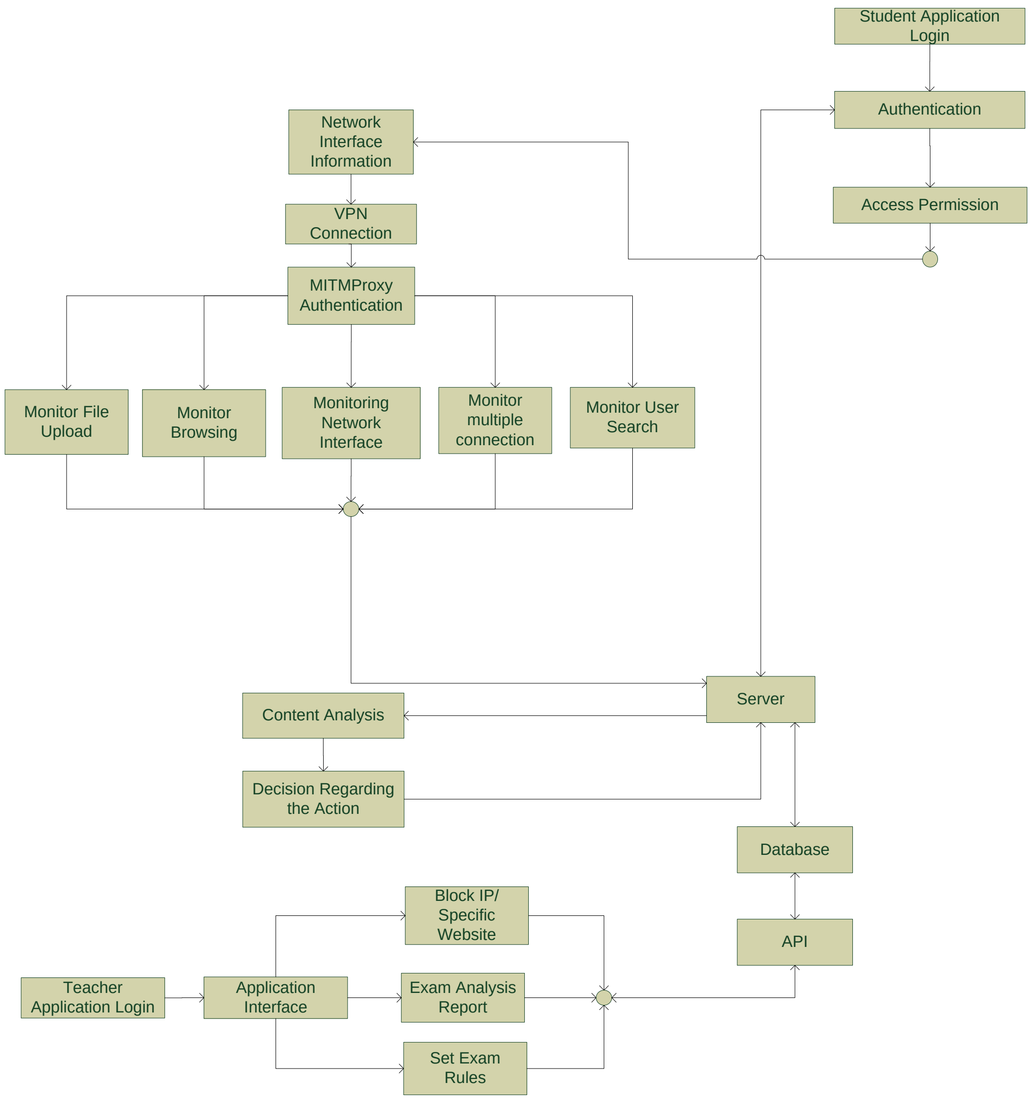


Figure 7: System architecture of the developed proctoring system.

The proposed proctoring system has a student desktop application that allows students to connect to the exam server using OpenVPN GUI. Mitmproxy authenticates the browser certificate first and then starts intercepting HTTP and HTTPs inbound and outbound traffic. The browser certificate establishes when a candidate connects to the exam server.

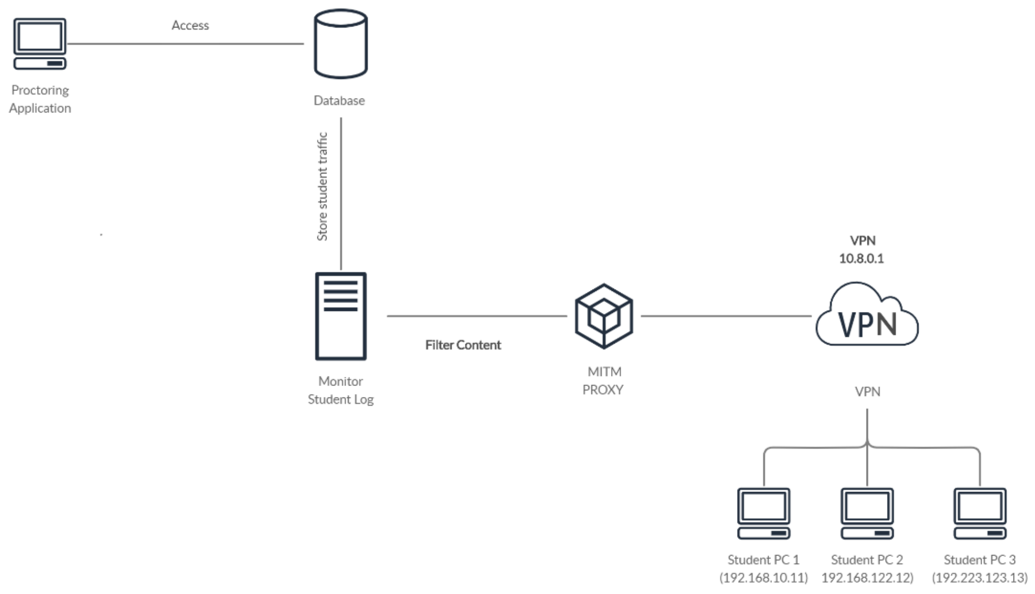


Figure 8: Design of the proctoring System.

The server implements the MITMproxy API, which separates the traffic content depending on the POST/GET request and analyzes it. To analyze the traffic content, we use Natural Language Processing using fuzzy string matching, which implements the Levenshtein distance metric algorithm. The application database stores the intercepted inbound and outbound network traffic content during the exam. It deletes confidential information, i.e., public IP address, private IP address, MAC address from the database after the exam. The proctor application also has an option to block specific IP/URLs during an exam.

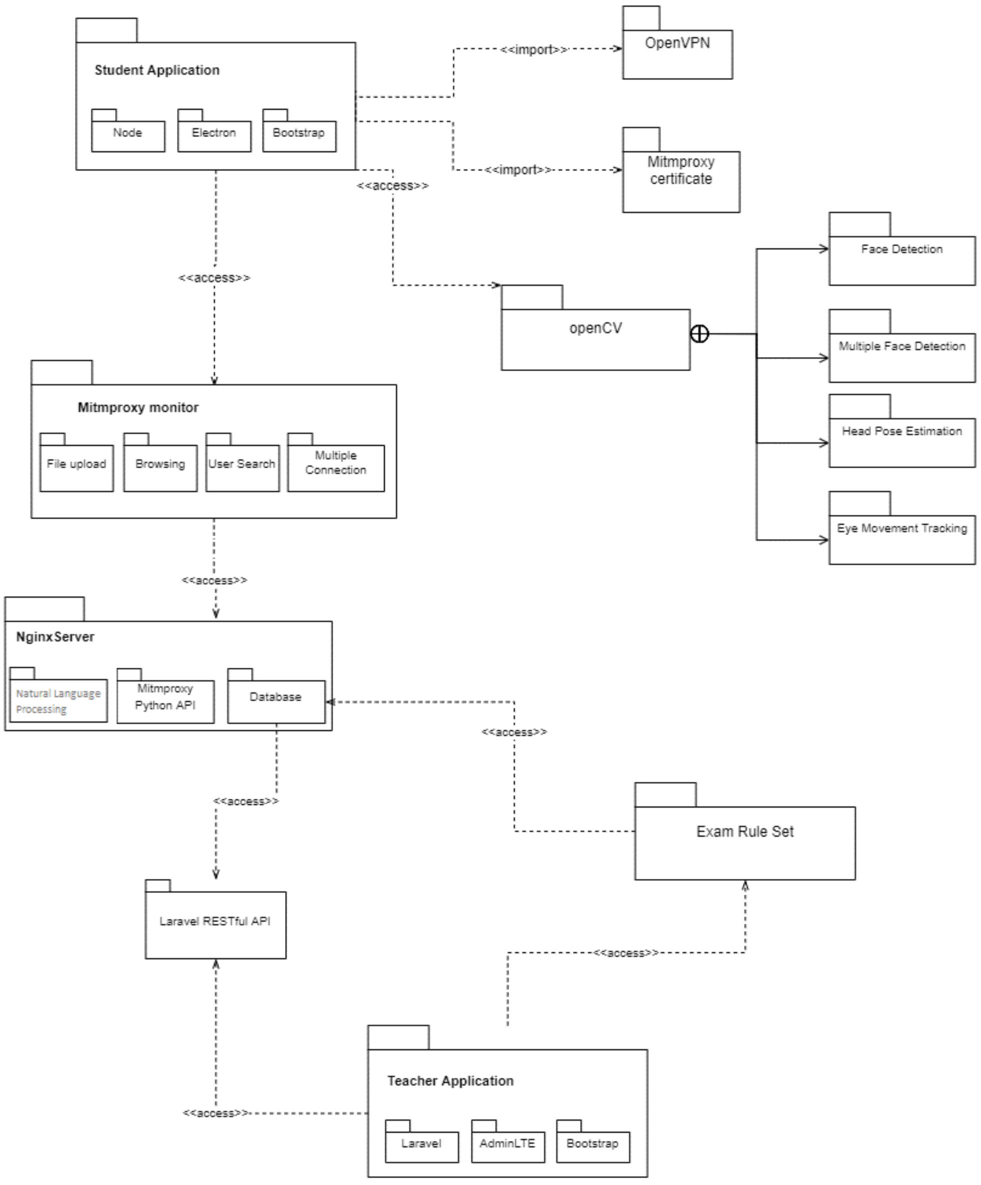
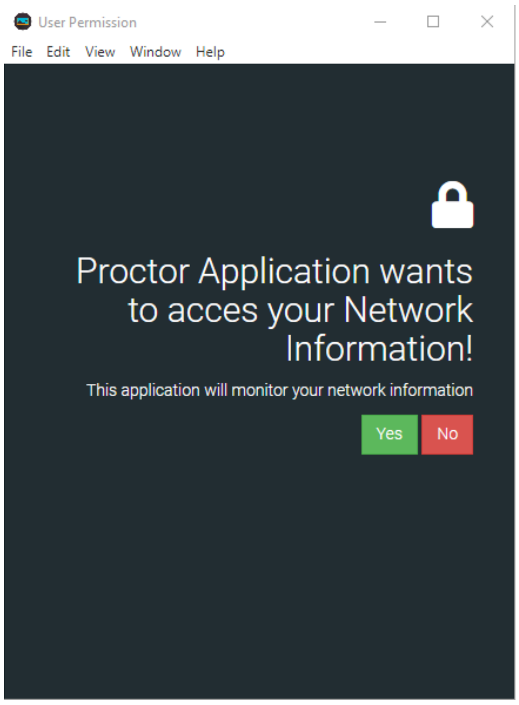


Figure 9: Package Diagram of Proctoring System.

3.2 Student monitoring application

The student desktop application connects students to the exam server and monitors the candidates' network interface information during the exam. The desktop application is developed using Electron JS(a runtime JavaScript framework) and Bootstrap3 (a CSS framework) (Electron JS, 2020). Electron is an open-source library to develop cross-platform desktop applications using HTML, CSS, and JavaScript.

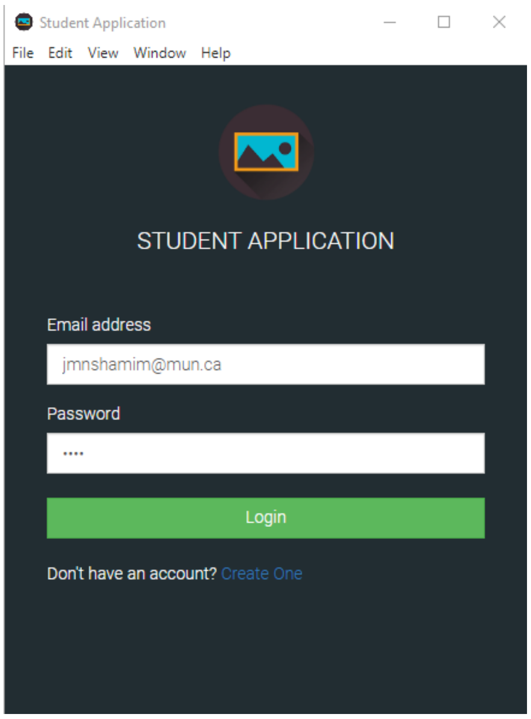

User Permission

File Edit View Window Help

Proctor Application wants to acces your Network Information!

This application will monitor your network information

Figure 10: Student permission section.


Student Application

File Edit View Window Help

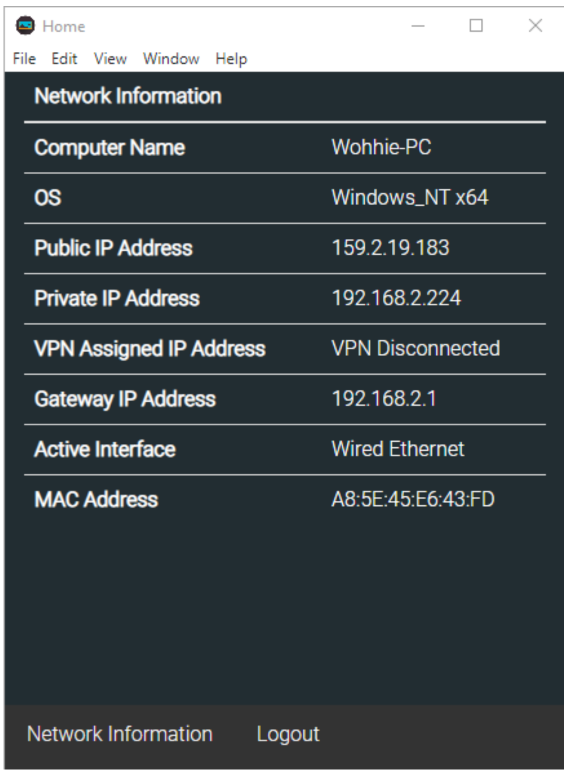
STUDENT APPLICATION

Email address: jmnshamim@mun.ca

Password: ····

Figure 11: Student login section.

The student desktop application verifies the user with an email address and password and fetches individual candidate course information through the API. The API also uses the post request to insert the network interface information to the database.



Computer Name	Wohhie-PC
OS	Windows_NT x64
Public IP Address	159.2.19.183
Private IP Address	192.168.2.224
VPN Assigned IP Address	VPN Disconnected
Gateway IP Address	192.168.2.1
Active Interface	Wired Ethernet
MAC Address	A8:5E:45:E6:43:FD

Figure 12: Accessing students network interface information.

The application also requests permission from the student because it accesses information from the network interface during the examination. It also accesses and stores students private and public IP addresses, mac addresses, type of OS, and active interface information. As these pieces of information are confidential, the application will delete that information from the database after completing the exam.

3.3 Proctor application

The web-based proctor application helps the proctor to set up an exam for individual courses and to monitor information about the network interface of the students. The proctor may also set rules for the examination with the application for proctoring. The application is developed on Laravel using AdminLTE and Bootstrap3. AdminLTE is one of the popular open-source templates for admin dashboards (Almsaeed, 2017). The responsive HTML5 template based on CSS3 framework Boostrap3. Laravel is a server-side web framework of PHP which follows the MVC architecture (Laravel, 2020).

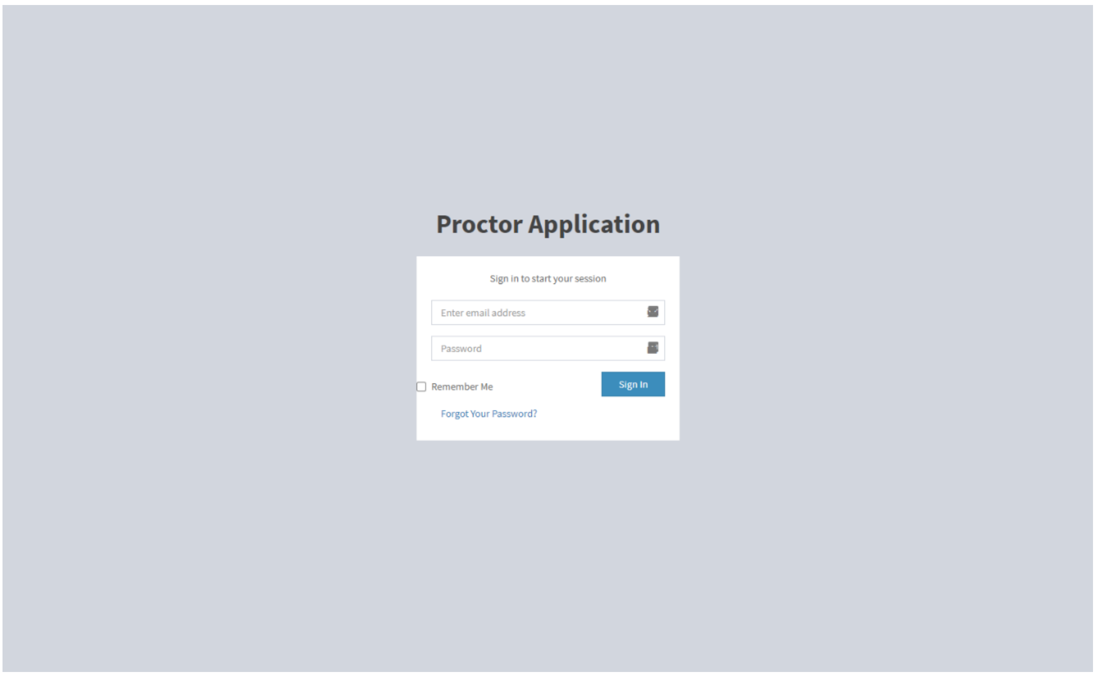
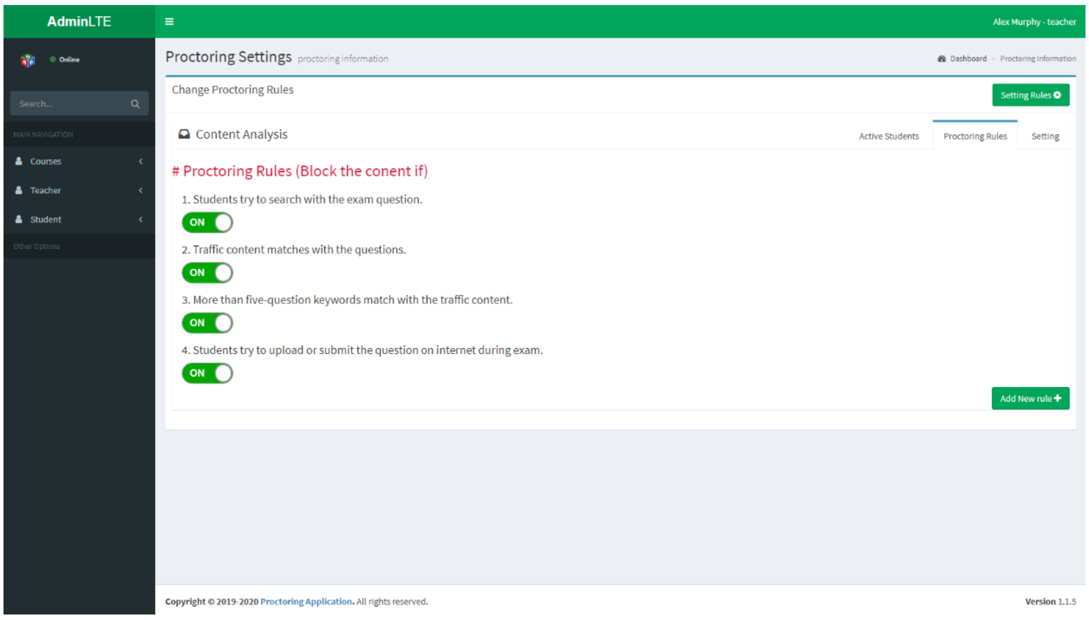


Figure 13: Proctor web application login page.

The web-based proctor application has a login section, where the proctor can log in with their login credentials (email address and password (Figure 12). After login, the proctor can set rules for individual exams using the proctoring setting section (Figure 13). We use four different types of proctoring rules in the system, which is very common in the exam. We tried to make these proctoring rules flexible and adjustable, depending on exams. The proctor can enable/disable these rules and during an exam.



Proctoring Settings proctoring information

Change Proctoring Rules

Content Analysis

Proctoring Rules (Block the conent if)

1. Students try to search with the exam question.
ON

2. Traffic content matches with the questions.
ON

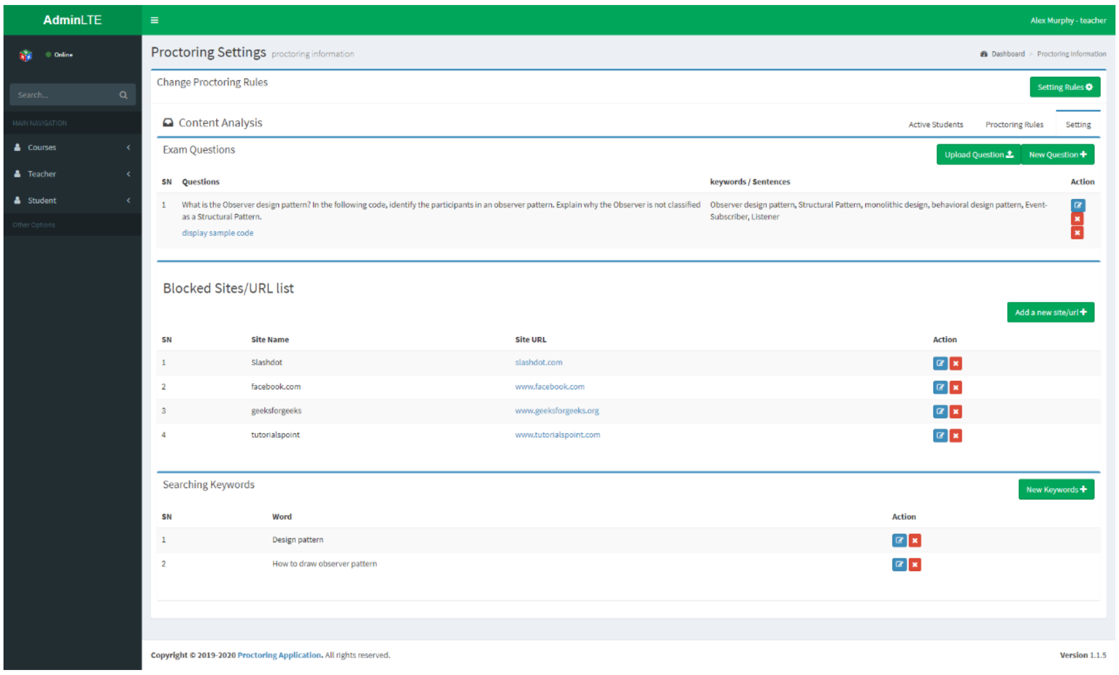
3. More than five-question keywords match with the traffic content.
ON

4. Students try to upload or submit the question on internet during exam.
ON

Copyright © 2019-2020 Proctoring Application. All rights reserved. Version 1.1.5

Figure 14: Proctoring rules setting section.

The proctor application has options to monitor students' network information and upload questions for an individual exam. The proctor can also block sites/URLs using the proctoring setting tab, which will block those websites throughout the exam (Figure 14).



Proctoring Settings proctoring information

Change Proctoring Rules

Content Analysis

Exam Questions

SN	Questions	keywords / Sentences	Action
1	What is the Observer design pattern? In the following code, identify the participants in an observer pattern. Explain why the Observer is not classified as a Structural Pattern. display sample code	Observer design pattern, Structural Pattern, monolithic design, behavioral design pattern, Event-Subscriber, Listener	

Blocked Sites/URL list

SN	Site Name	Site URL	Action
1	Slashdot	slashdot.com	
2	facebook.com	www.facebook.com	
3	geeksforgeeks	www.geeksforgeeks.org	
4	tutorialspoint	www.tutorialspoint.com	

Searching Keywords

SN	Word	Action
1	Design pattern	
2	How to draw observer pattern	

Copyright © 2019-2020 Proctoring Application. All rights reserved. Version 1.1.5

Figure 15: Proctoring rules and monitoring section.

3.4 OpenVPN

OpenVPN has the technology for building point-to-point secure connections in routed or remote access facilities.

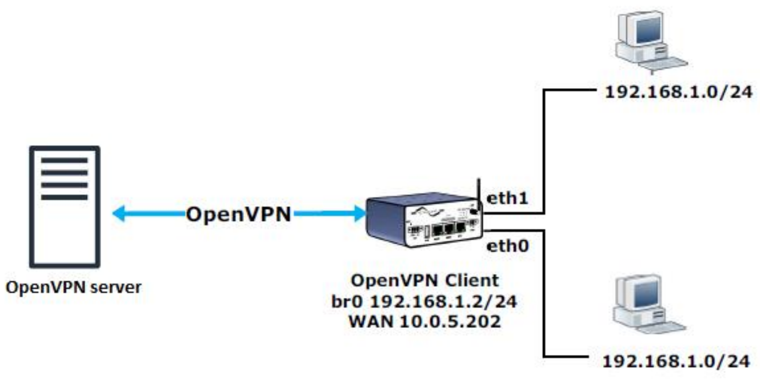


Figure 16: How open VPN works.

The desktop student program first imports the OpenVPN config file (Students.openvpn), and then students must use the OpenVPN Interface to connect to the exam server to start the examination. The student application will monitor students' network interface information while connected to the VPN and observe if any change happens on the network interface throughout the exam.

3.5 MITMproxy

MITMproxy is an HTTP proxy/HTTP monitor that allows developers to view all HTTP and SSL/HTTPS traffic within the client and the server. The MITM stands for Man-In-The-Middle; the basic idea behind it is to represent the server to the client and be the client to the server, while in the middle, we can monitor the traffic coming from both sides on MITM proxy monitor.

The Certificate Authority policy is to create protected connections to a server via the Internet, providing a trusted party to cryptographically confirm a server's documents to validate that they are legit.

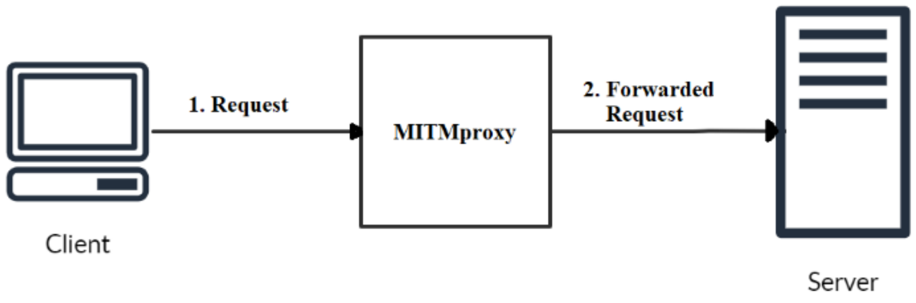
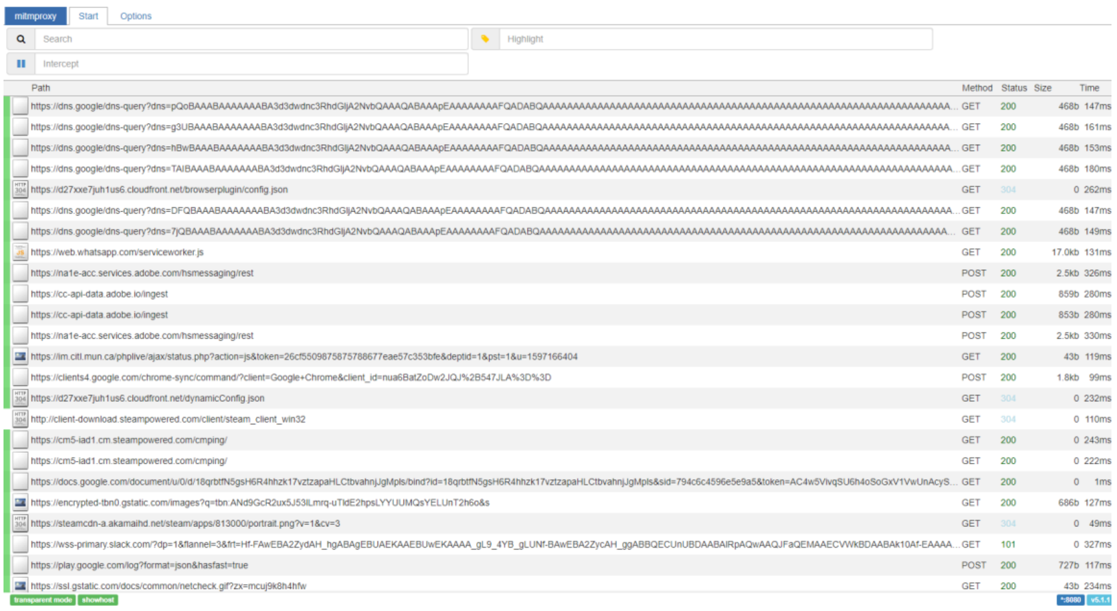


Figure 17: How MITMproxy works.

As we know that the server has a Certificate authority with both public and private keys, and the client who has these keys can only connect securely to the Server. If this credential does not match, a secure client will drop the connection and refuse to proceed.



Path	Method	Status	Size	Time
https://dns.google/dns-query?dns=pQoBAAABAAAAAAAABA3d3dwdnc3RhdGljA2NvbQAAAQABAAApEAAAAAAAAFQADABQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...	GET	200	468b	147ms
https://dns.google/dns-query?dns=g3UBAAABAAAAAAAABA3d3dwdnc3RhdGljA2NvbQAAAQABAAApEAAAAAAAAFQADABQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...	GET	200	468b	161ms
https://dns.google/dns-query?dns=hBwBAAABAAAAAAAABA3d3dwdnc3RhdGljA2NvbQAAAQABAAApEAAAAAAAAFQADABQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...	GET	200	468b	153ms
https://dns.google/dns-query?dns=TAIBAAABAAAAAAAABA3d3dwdnc3RhdGljA2NvbQAAAQABAAApEAAAAAAAAFQADABQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...	GET	200	468b	180ms
https://d27xxe7juh1us6.cloudfront.net/browserplugin/config.json	GET	304	0	262ms
https://dns.google/dns-query?dns=DFQBAAABAAAAAAAABA3d3dwdnc3RhdGljA2NvbQAAAQABAAApEAAAAAAAAFQADABQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...	GET	200	468b	147ms
https://dns.google/dns-query?dns=7jQBAAABAAAAAAAABA3d3dwdnc3RhdGljA2NvbQAAAQABAAApEAAAAAAAAFQADABQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...	GET	200	468b	149ms
https://web.whatsapp.com/serviceworker.js	GET	200	17.0kb	131ms
https://na1e-acc.services.adobe.com/hsmessaging/rest	POST	200	2.5kb	326ms
https://cc-api-data.adobe.io/ingest	POST	200	859b	280ms
https://cc-api-data.adobe.io/ingest	POST	200	853b	280ms
https://na1e-acc.services.adobe.com/hsmessaging/rest	POST	200	2.5kb	330ms
https://im.citl.mun.ca/phplive/ajax/status.php?action=js&token=26cf5509875875788677eae57c353bfe&deptid=1&pst=1&u=1597166404	GET	200	43b	119ms
https://clients4.google.com/chrome-sync/command/?client=Google+Chrome&client_id=nua6BatZoDw2JQJ%2B547JLA%3D%3D	POST	200	1.8kb	99ms
https://d27xxe7juh1us6.cloudfront.net/dynamicConfig.json	GET	304	0	232ms
http://client-download.steampowered.com/client/steam_client_win32	GET	304	0	110ms
https://cm5-iad1.cm.steampowered.com/cmping/	GET	200	0	243ms
https://cm5-iad1.cm.steampowered.com/cmping/	GET	200	0	222ms
https://docs.google.com/document/u/0/d/18qrbtfN5gsH6R4hhzk17vztzapaHLCtbvahnjJgMpls/bind?id=18qrbtfN5gsH6R4hhzk17vztzapaHLCtbvahnjJgMpls&sid=794c6c4596e5e9a5&token=AC4w5VlvqSU6h4oSoGxV1VwUnAcyS...	GET	200	0	1ms
https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR2ux5J53lLmrq-uTldE2hpsLYYUUMQsYELUnT2h6o&s	GET	200	686b	127ms
https://steamcdn-a.akamaihd.net/steam/apps/813000/portrait.png?v=1&cv=3	GET	304	0	49ms
https://wss-primary.slack.com/?dp=1&flannel=3&frt=Hf-FAwEBA2ZydAH_hgABAgEBUAEKAAEBUwEKAAAA_gL9_4YB_gLUNf-BAwEBA2ZycAH_ggABBQECUnUBDAABAIRpAQwAAQJFaQEMAAECVWkBDAABAk10Af-EAAAA...	GET	101	0	327ms
https://play.google.com/log?format=json&hasfast=true	POST	200	727b	117ms
https://ssl.gstatic.com/docs/common/netcheck.gif?zx=mcuj9k8h4hfw	GET	200	43b	234ms

Figure 18: Intercepted network inbound and outbound traffic by MITMproxy.

The proctor application also implements the MITMproxy python API (mitmdump) on the server-side. On the server, the python script separates the intercepted traffic content by MITMproxy depending on the POST / GET request (Figure 18). Also, the python script fetches all the proctoring rules from the database using the Laravel RESTful API and applies those rules in the server. The content analysis part is also implemented inside of the python script, which is a combination of Natural Language Processing using fuzzy string and the Levenshtein distance metric algorithm.

3.6 Natural language processing using Fuzzy string matching

Fuzzy string matching is a way of finding strings that match a pattern approximately. This check and seeks matches even if the word is mispronounced or the word is incomplete. This is called approximate matching. The proctor application uses the fuzzy string matching to match content between the exam question and the internet content. The fuzzy string matching applies the algorithm for editing distance or the Levenshtein distance algorithm to measure whether two strings are identical and Levenshtein distance is most often used for mapping and comparing genomes.

3.6.1 Levenshtein distance metric algorithm

The Levenshtein metric distance algorithm estimates the difference between the two-word sequences (Okuda, Tanaka, & Kasai, 1976). In other words, it calculates the minimum number of edits needed to alter the sequence of one word to another, and the changes can be insertions, deletions, or replacements.

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j) + 1 \\ \operatorname{lev}_{a,b}(i,j-1) + 1 \\ \operatorname{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Figure 19: Levenshtein distance metric algorithm (Veena G, 2015).

Where 1 $_{(ai \neq bj)}$ denotes 0 when a=b and 1 otherwise.

It is vital to note that the rows on the minimum above apply to a deletion, an insertion, and a substitution in that order.

3.7 Database schema diagram

The database design figure17 shows the database design of the proctoring system. The student and proctor application both share the same database and store information. The design of the database schema diagram shows the relationship between tables.

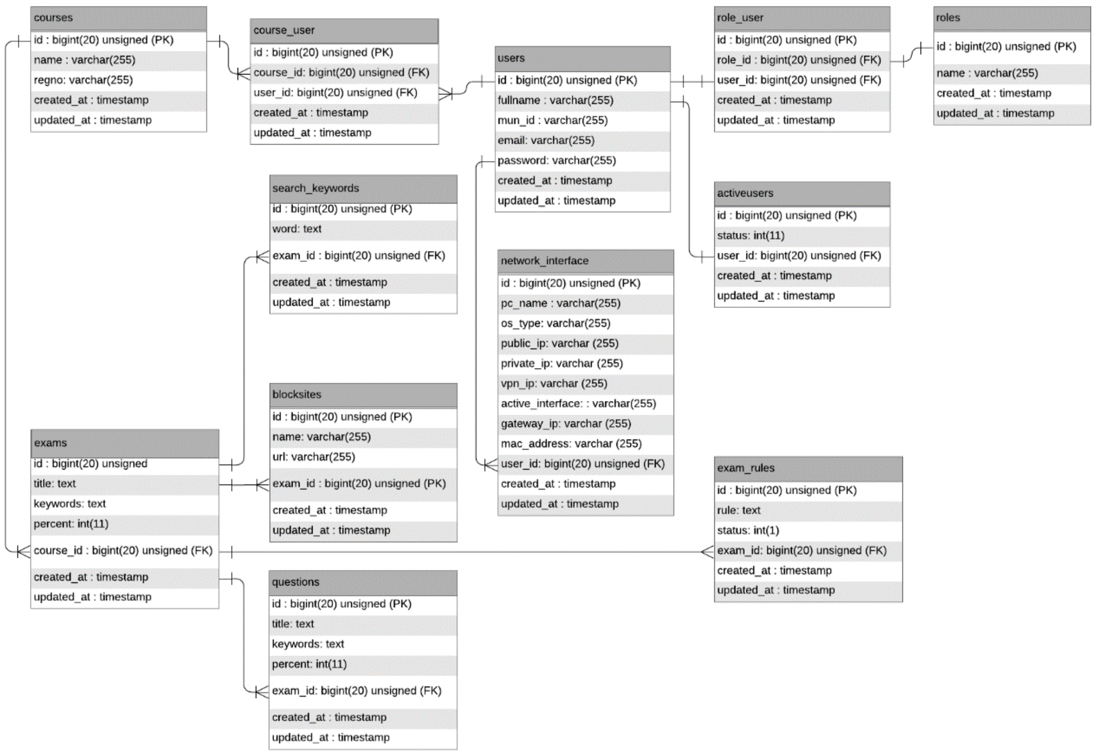


Figure 20: Database Enhanced entity-relationship diagram.

3.8 RESTful API

Representational State Transfer RESTful web services are a way of implementing interoperability among computer methods on the internet. The RESTful API is an architectural style that allows the system to access and handle textual descriptions of web resources. In this project, the RESTful API plays a vital role in exchanging information between the proctoring web-based and student desktop applications.

A RESTful web service call includes an endpoint URL(domain, port, path) and the HTTP method. The proctoring application uses mainly three HTTP methods (GET, POST, and DELETE) to do any endpoint (CRUD) operation.

Examples:

- a **GET** request to */rules/* returns a list of proctoring rules on the system
- a **POST** request to */rules/1234* creates a proctoring rule with the ID *1234* using the body data.
- a **PUT** request to */rules/1234* updates rules *1234* with the body data.
- a **GET** request to */rules/1234* returns the details of the rule *1234*
- a **DELETE** request to */rules/1234* deletes rule *1234*

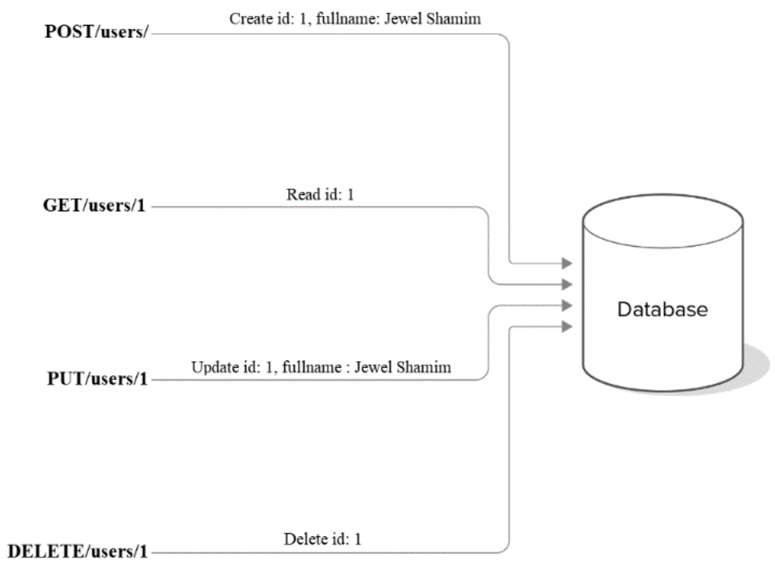


Figure 21: Visual representation of the RESTful API works in the proctor system.

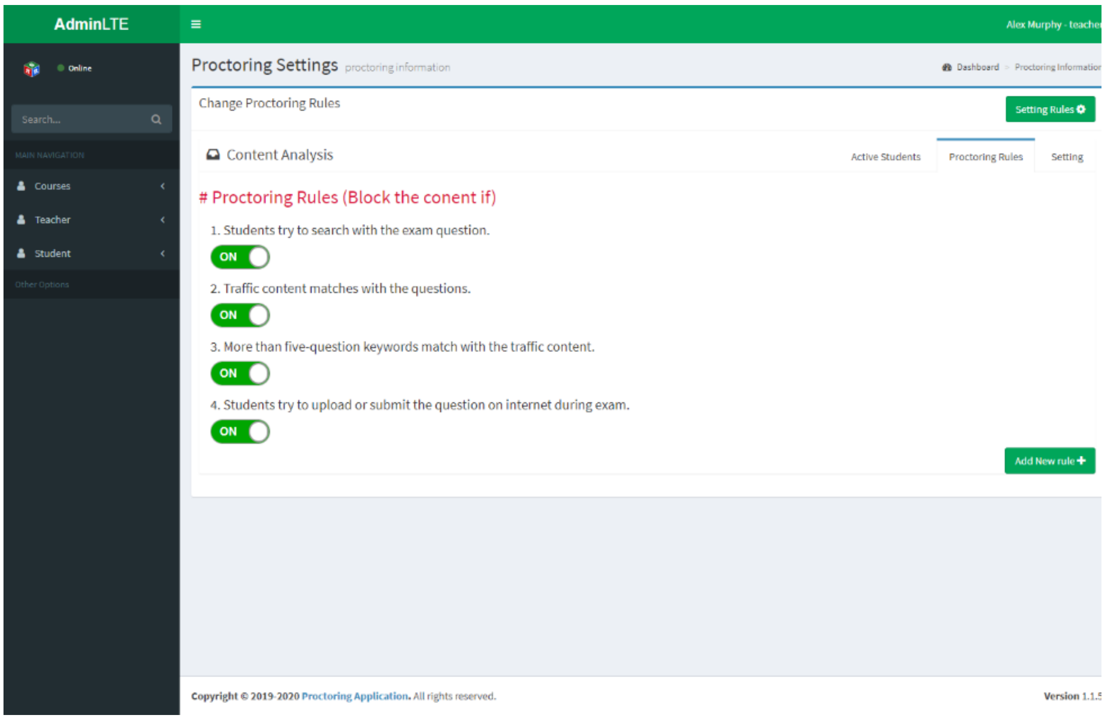
4 Testing

4.1 Content analysis with proctoring rules

The web-based proctoring application is used to set the proctoring rules and monitor exam takers' network interface information during the exam. We tested our application with four proctoring rules. The proctoring rules we use in the project are flexible and adjustable depending on the exam requirement.

<u>Block the traffic inbound and outbound content if:</u>

1. Students attempt to search with the exam question.
2. Network inbound and outbound traffic content matches with the exam questions.
3. More than five-question keywords match with the network inbound and outbound traffic content.
4. Students try to post the question during the exam.
   For example: Posting the question to Stack Overflow /Slashdot etc.



Proctoring Settings proctoring information

Change Proctoring Rules

Content Analysis

Proctoring Rules (Block the conent if)

1. Students try to search with the exam question.
ON

2. Traffic content matches with the questions.
ON

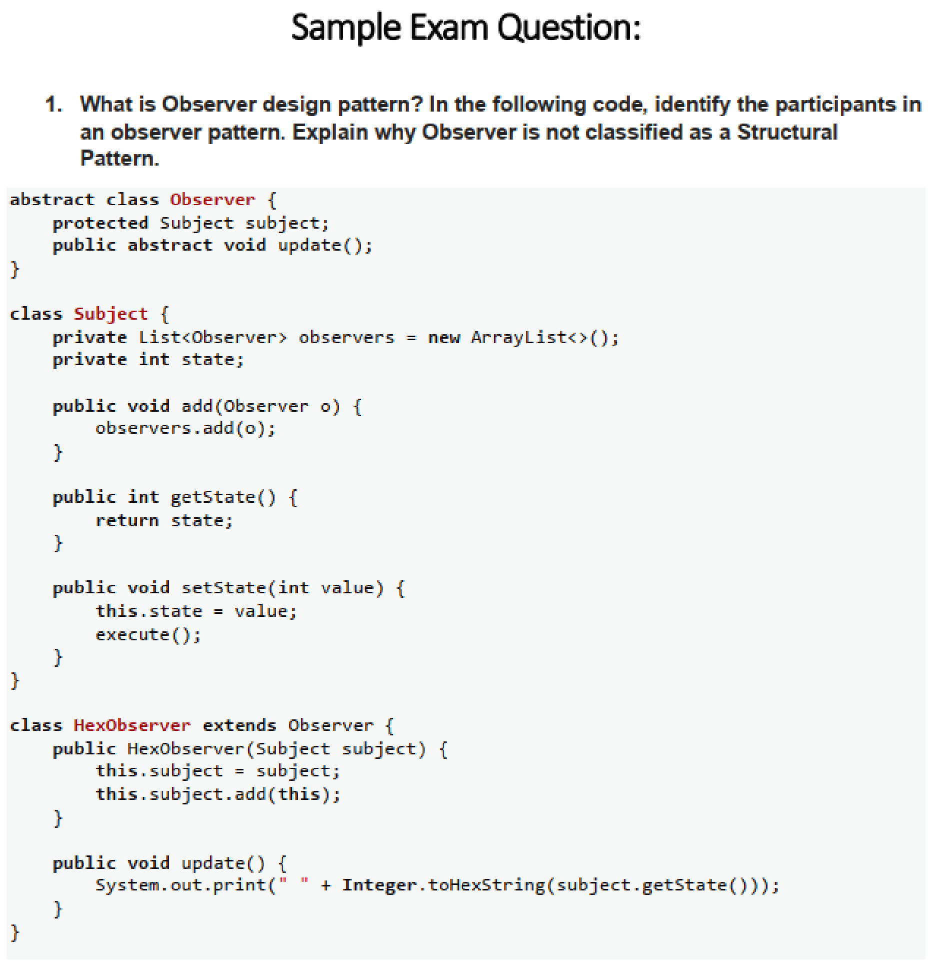
3. More than five-question keywords match with the traffic content.
ON

4. Students try to upload or submit the question on internet during exam.
ON

Copyright © 2019-2020 Proctoring Application. All rights reserved.
Version 1.1.5

Figure 22: Proctoring rules setting section.

We tested our proctoring system with these four proctoring rules and included each test performance in the testing section. We mentioned that our proctoring rules are adjustable because proctor can enable/disable the proctoring rules depending on the exam (Figure 21). We tested our proctoring application, where the internet is accessible during the exam. In the testing section, we will show the proctoring application process with a software engineering question (Figure 23).



Sample Exam Question:

1. **What is Observer design pattern? In the following code, identify the participants in an observer pattern. Explain why Observer is not classified as a Structural Pattern.**

```
abstract class Observer {
    protected Subject subject;
    public abstract void update();
}

class Subject {
    private List<Observer> observers = new ArrayList<>();
    private int state;

    public void add(Observer o) {
        observers.add(o);
    }

    public int getState() {
        return state;
    }

    public void setState(int value) {
        this.state = value;
        execute();
    }
}

class HexObserver extends Observer {
    public HexObserver(Subject subject) {
        this.subject = subject;
        this.subject.add(this);
    }

    public void update() {
        System.out.print(" " + Integer.toHexString(subject.getState()));
    }
}
```

Figure 23: Software engineering question to test the proctor application.

<u>Rule #1: Block content if students try to search with the exam question</u>:

When the internet is accessible during an exam, students can search for anything on the internet. In this proctoring system, we consider this as cheating because the proctor set the rule as students cannot search with exam questions. So, the proctoring application will block the content if any student tries to search with the exam question during an online exam.

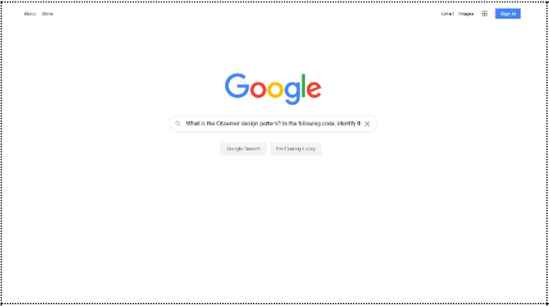
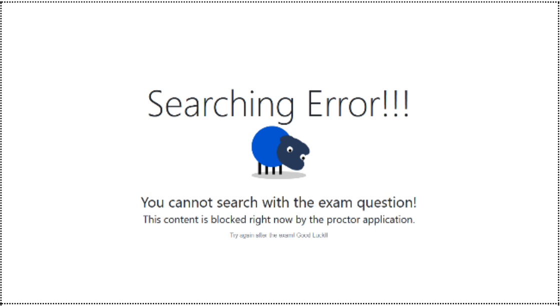

What is the Observer design pattern? In the following code, identify tl

Figure 24: Searching with question.


Searching Error!!!

You cannot search with the exam question!

This content is blocked right now by the proctor application.

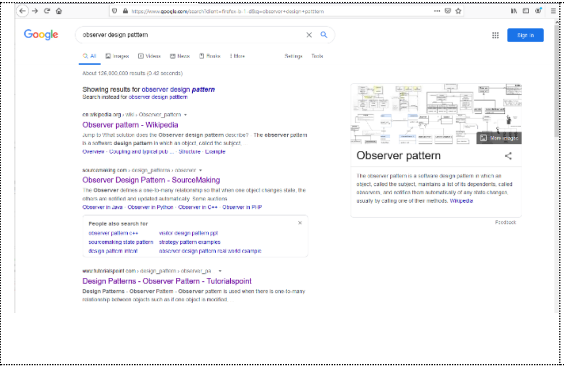
Try again after the exam! Good Luck!!

Figure 25: Application block the content.

In this example (Figure 24), one of the students tries to search on google with the exam question. The Mitmproxy intercepts the inbound and outbound network traffic and separates the content depending on the POST/GET requests. In the server, the mitmdump python API fetches the content, and compares the intercepted traffic contents with the exam questions. The Natural Language Processing using fuzzy string matching finds the approximate similarity ratio between the intercepted traffic content and the exam questions. The proctoring application blocked the content as it matched with the exam question.

Rule #2: Block the content if traffic content matches with the exam questions:

Sometimes it is very common when the exam question is already on the internet. Moreover, it is easy for students to copy the content from the internet. Our second proctoring rule will block the content if the network traffic content (inbound and outbound traffic) matches the exam questions.


https://www.google.com/search?client=firefox-b-1-d&q=observer+design+patttern

observer design patttern

About 126,000,000 results (0.42 seconds)

Showing results for observer design *pattern*
Search instead for observer design patttern

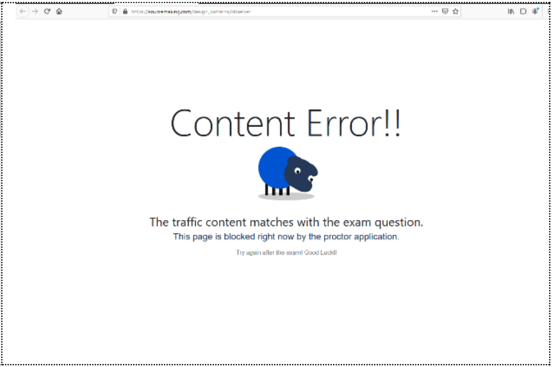
en.wikipedia.org › wiki › Observer_pattern
Observer pattern - Wikipedia
Jump to What solution does the Observer design pattern describe? · The observer pattern is a software design pattern in which an object, called the subject, ...
Overview · Coupling and typical pub ... · Structure · Example

sourcemaking.com › design_patterns › observer
Observer Design Pattern - SourceMaking
The Observer defines a one-to-many relationship so that when one object changes state, the others are notified and updated automatically. Some auctions ...
Observer in Java · Observer in Python · Observer in C++ · Observer in PHP

People also search for
observer pattern c++ | visitor design pattern ppt
sourcemaking state pattern | strategy pattern examples
design pattern intent | observer design pattern real world example

www.tutorialspoint.com › design_pattern › observer_pa...
Design Patterns - Observer Pattern - Tutorialspoint
Design Patterns - Observer Pattern - Observer pattern is used when there is one-to-many relationship between objects such as if one object is modified, ...

Observer pattern
The observer pattern is a software design pattern in which an object, called the subject, maintains a list of its dependents, called observers, and notifies them automatically of any state changes, usually by calling one of their methods. Wikipedia


Content Error!!

The traffic content matches with the exam question.

This page is blocked right now by the proctor application.

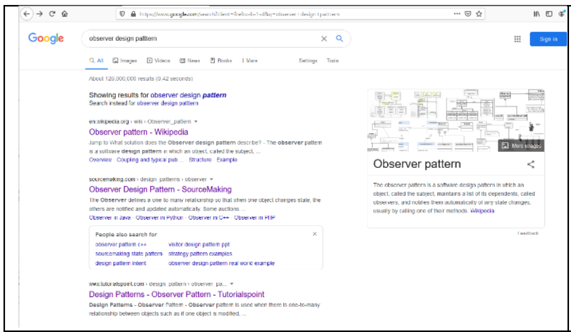
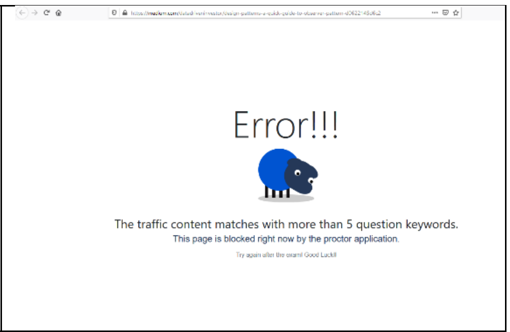
Try again after the exam! Good Luck!!

Figure 26: Search result from google.	Figure 27: Content is blocked by proctor application.

Example:
A student searches with *"Observer Design Pattern"* on google and opens some related links of observer design patterns (Figure 26). The MITMproxy first intercepts all the inbound and outbound network traffic and mitmdump python API fetches the website's content. Then, the Natural Language Processing using Fuzzy string matching compares the similarities between intercepted content and the exam question. The proctor application blocks the content as it matches the website (https://sourcemaking.com/design_patterns/observer) content.

<u>Rule #3: Block the content if more than five-question keywords match the network inbound and outbound traffic content:</u>

The proctor application has a feature where the teacher can set keywords for a question. The keyword can be related to questions or answers. Depending on the keyword, if the specific website content matches more than a five-question keyword, then the proctor application will block the traffic of the content.

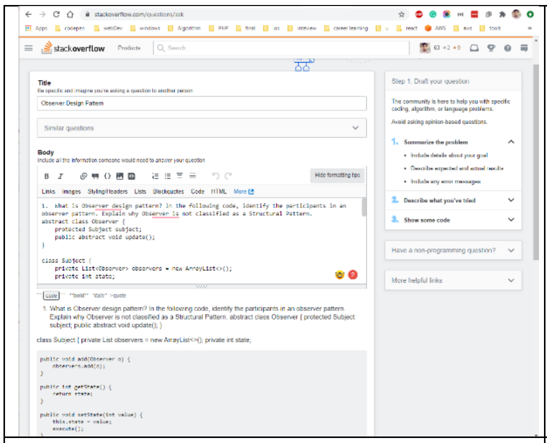
 observer design pattern About 125,000,000 results (0.42 seconds) Showing results for observer design *pattern* Search instead for observer design pattern en.wikipedia.org › wiki › Observer_pattern Observer pattern - Wikipedia Jump to What solution does the **Observer design pattern** describe? - The **observer pattern** is a software **design pattern** in which an object, called the subject, ... Overview · Coupling and typical pub ... · Structure · Example sourcemaking.com › design_patterns › observer Observer Design Pattern - SourceMaking The **Observer** defines a one to many relationship so that when one object changes state, the others are notified and updated automatically. Some auctions ... Observer in Java · Observer in Python · Observer in C++ · Observer in PHP People also search for observer pattern c++ · visitor design pattern ppt sourcemaking state pattern · strategy pattern examples design pattern intent · observer design pattern real world example www.tutorialspoint.com › design_pattern › observer_pa... Design Patterns - Observer Pattern - Tutorialspoint **Design Patterns - Observer** Pattern - **Observer** pattern is used when there is one-to-many relationship between objects such as if one object is modified, ... Observer pattern The observer pattern is a software design pattern in which an object, called the subject, maintains a list of its dependents, called observers, and notifies them automatically of any state changes, usually by calling one of their methods. Wikipedia	 Error!!! The traffic content matches with more than 5 question keywords. This page is blocked right now by the proctor application. Try again after the exam! Good Luck!!
Figure 28: Searching result from google.	Figure 29: Traffic content matches with more than 5 question keywords.

In this example, one of the students searched with "Observer Design Pattern" on google and opened some related links of observer design patterns. The MITMproxy first intercepts all the inbound and outbound network traffic and mitmdump python API fetches the website's content. Then, the Natural Language Processing using Fuzzy string matching compares the similarities between intercepted content and the question keywords. The content of the website (https://medium.com/datadriveninvestor/design-patterns-a-quick-guide-to-observerpattern) matches with more than five keywords of the question. That is why the proctor application blocks the content of the website during the exam.

Rule #4: Students try to upload or submit the question anywhere during the exam.

It is a familiar scenario where the exam taker tries to upload the question on some problem-solving website, such as Stack overflow, Slashdot, etc. Many websites solve exam questions. Our fourth proctoring rule is that students cannot post or upload the question to any website during the exam. The exam will be terminated using the response traffic analysis, and we block the upload during the exam.

We analyze all the POST and GET requests happening through the network. As MITMproxy intercepts all the HTTPs and HTTP traffic, we also intercept all the POST and GET requests during the exam. Moreover, analyze the content of the request with all the questions. With the help of Fuzzy string matching and the Levenshtein distance metric algorithm, we also analyze the content. If content exactly or approximately matches the question, the proctoring application will block the traffic to upload /submission.



stackoverflow.com/questions/ask

Title
Be specific and imagine you're asking a question to another person

Observer Design Pattern

Body
Include all the information someone would need to answer your question

```
1.  What is Observer design pattern? In the following code, identify the participants in an
observer pattern. Explain why Observer is not classified as a Structural Pattern.
abstract class Observer {
    protected Subject subject;
    public abstract void update();
}

class Subject {
    private List<Observer> observers = new ArrayList<>();
    private int state;
```

1. What is Observer design pattern? In the following code, identify the participants in an observer pattern. Explain why Observer is not classified as a Structural Pattern. abstract class Observer { protected Subject subject; public abstract void update(); }

class Subject { private List observers = new ArrayList<>(); private int state;

```
public void add(Observer o) {
    observers.add(o);
}

public int getState() {
    return state;
}

public void setState(int value) {
    this.state = value;
    execute();
```

Step 1. Draft your question

The community is here to help you with specific coding, algorithm, or language problems.

Avoid asking opinion-based questions.

1. Summarize the problem
- Include details about your goal
- Describe expected and actual results
- Include any error messages

2. Describe what you've tried

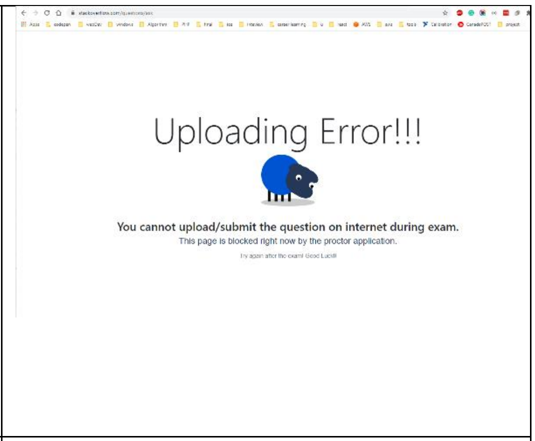
3. Show some code

Have a non-programming question?

More helpful links

Products
Search
Similar questions
Hide formatting tips
Links Images Styling/Headers Lists Blockquotes Code HTML More

Figure 30: Student tries to submit the question to Stack Overflow.



Uploading Error!!!

You cannot upload/submit the question on internet during exam.

This page is blocked right now by the proctor application.

Try again after the exam! Good Luck!!

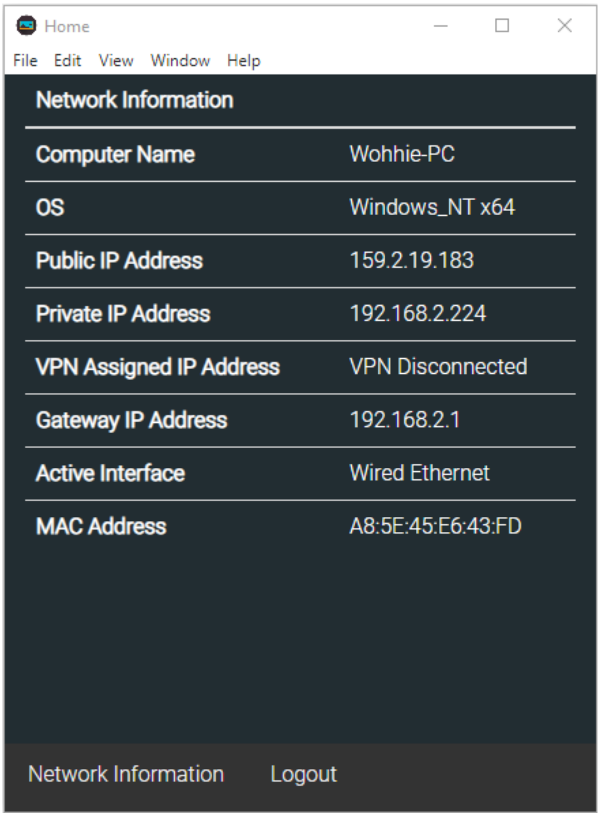
Figure 31: Block submission using traffic content analysis.

In this scenario, one of the students tries to upload a question to *Stack Overflow* during the exam. However, the proctoring application intercepts the POST request using MITMproxy and analyzes the content of the traffic. As the content matched with the question, the proctor application blocks the upload to *Stack Overflow*.

4.2 Optional Features

4.2.1 Networking monitoring during the exam

The proctoring application has some other features, which monitor students' network interface information during the exam. Some VPN services have an option to do split-tunneling. We already disable this split-tunneling service from our exam server. But if any student tries to do the split-tunneling during the exam, the proctoring application will terminate the exam and cancel the VPN connection.



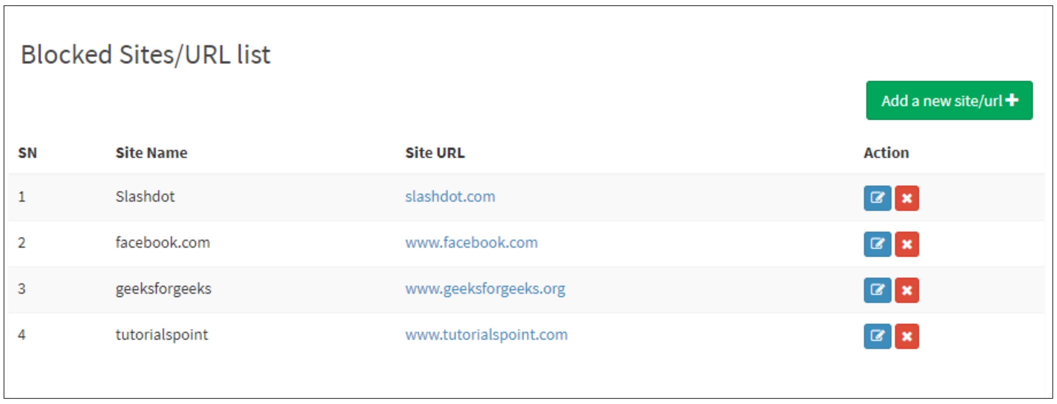
Network Information

Computer Name	Wohhie-PC
OS	Windows_NT x64
Public IP Address	159.2.19.183
Private IP Address	192.168.2.224
VPN Assigned IP Address	VPN Disconnected
Gateway IP Address	192.168.2.1
Active Interface	Wired Ethernet
MAC Address	A8:5E:45:E6:43:FD

Figure 32: Student application monitor students network interface.

4.2.2 Block specific IP/URL

The proctor application has a feature from where the proctor can set specific IP/URL to block for an individual exam. We set this option open because sometimes students share the question solution using social media (Facebook, WhatsApp, Messenger).



Blocked Sites/URL list

SN	Site Name	Site URL	Action
1	Slashdot	slashdot.com	
2	facebook.com	www.facebook.com	
3	geeksforgeeks	www.geeksforgeeks.org	
4	tutorialspoint	www.tutorialspoint.com	

Figure 33: Block specific IP/URLs.

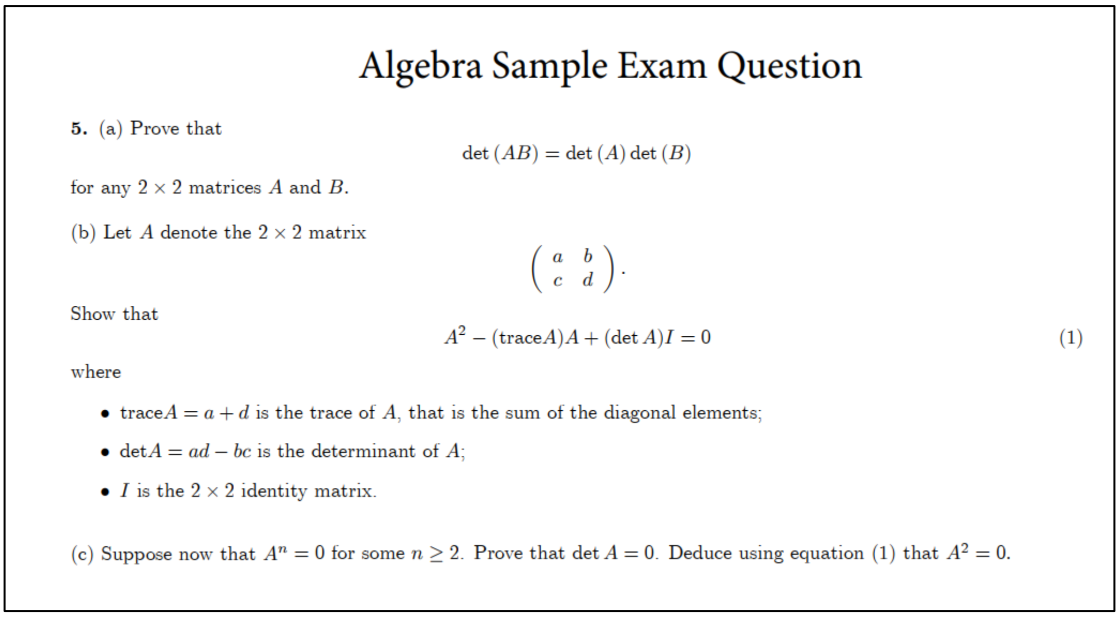
5 Results

In this section, we discuss the performance of the proctoring application and the testing result of our system with a sample software engineering related question (Figure 23), and algebra exam question (Figure 34), and a chemistry exam question (Figure 36). Also, we will discuss the limitations of the proctoring system we found during this testing session.

5.1 Proctoring rules and features

The testing result with an algebra exam question:

In this section, we tested our proctoring application with an algebra sample exam question (Figure 34). To test the proctoring application limit, we made the algebra exam question complex with some equations as the proctoring application performs better if the question is a sequence of words or straight lines. But in this test, the proctoring application successfully blocked the traffic content, and even the question has complicated mathematical equations (Figure 34). It also restricts the user from opening PDF files that have related content of the exam question (Figure 35).



Algebra Sample Exam Question

5. (a) Prove that

$$\det(AB) = \det(A)\det(B)$$

for any 2×2 matrices A and B.

(b) Let A denote the 2×2 matrix

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

Show that

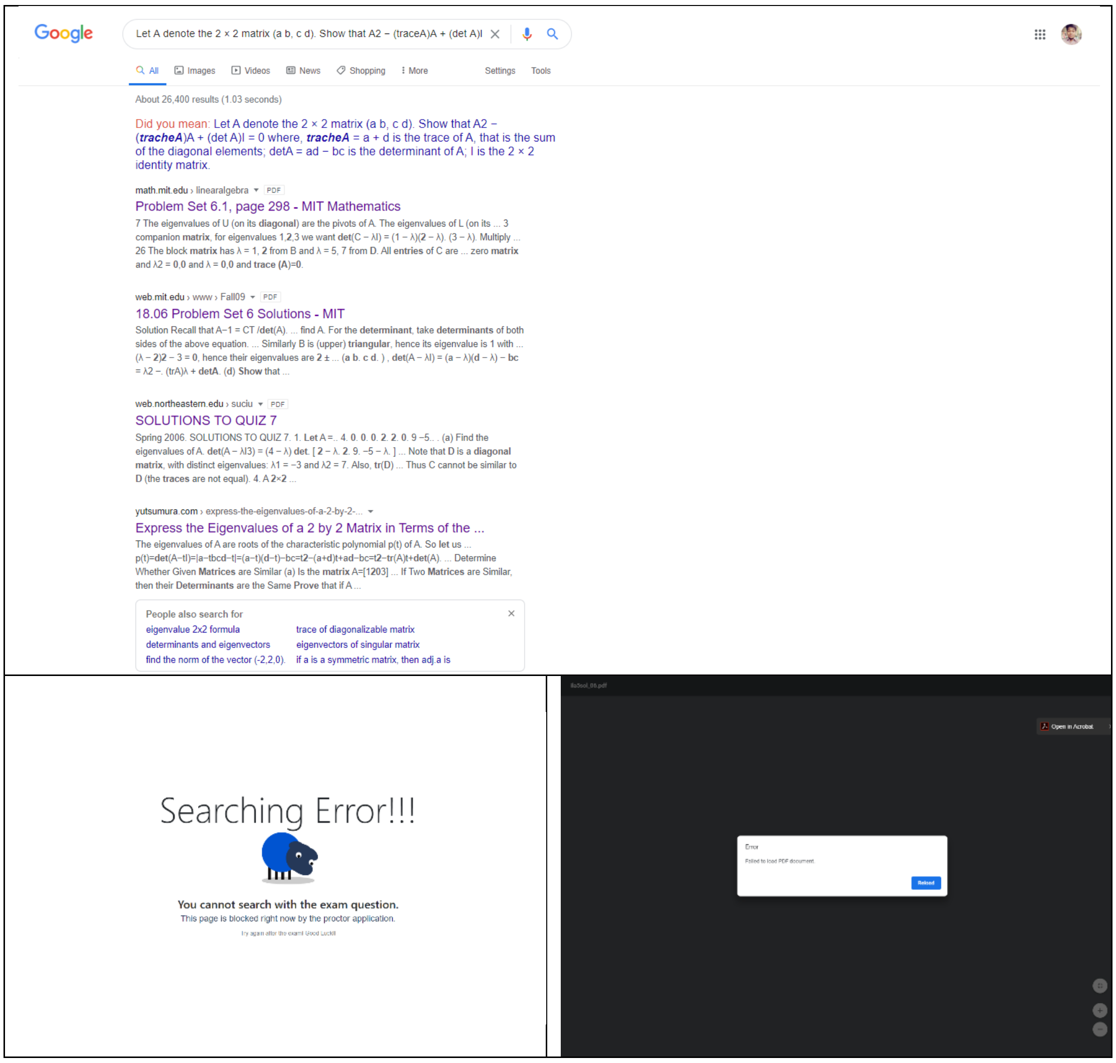
$$A^2 - (\text{trace}A)A + (\det A)I = 0 \qquad (1)$$

where

- $\text{trace}A = a + d$ is the trace of A, that is the sum of the diagonal elements;
- $\det A = ad - bc$ is the determinant of A;
- I is the 2×2 identity matrix.

(c) Suppose now that $A^n = 0$ for some $n \geq 2$. Prove that $\det A = 0$. Deduce using equation (1) that $A^2 = 0$.

Figure 34: An Algebra sample question to test the proctoring application.



Let A denote the 2 × 2 matrix (a b, c d). Show that A2 − (traceA)A + (det A)I

About 26,400 results (1.03 seconds)

Did you mean: Let A denote the 2 × 2 matrix (a b, c d). Show that A2 − (***tracheA***)A + (det A)I = 0 where, ***tracheA*** = a + d is the trace of A, that is the sum of the diagonal elements; detA = ad − bc is the determinant of A; I is the 2 × 2 identity matrix.

math.mit.edu › linearalgebra PDF
Problem Set 6.1, page 298 - MIT Mathematics
7 The eigenvalues of U (on its **diagonal**) are the pivots of A. The eigenvalues of L (on its ... 3 companion **matrix**, for eigenvalues 1,**2**,3 we want **det**(C − λI) = (1 − λ)(**2** − λ). (3 − λ). Multiply ... 26 The block **matrix** has λ = 1, **2** from B and λ = 5, 7 from D. All **entries** of C are ... zero **matrix** and λ2 = **0**,**0** and λ = **0**,**0** and **trace** (**A**)=**0**.

web.mit.edu › www › Fall09 PDF
18.06 Problem Set 6 Solutions - MIT
Solution Recall that A−1 = CT /**det**(A). ... find A. For the **determinant**, take **determinants** of both sides of the above equation. ... Similarly B is (upper) **triangular**, hence its eigenvalue is 1 with ... (λ − **2**)**2** − 3 = **0**, hence their eigenvalues are **2** ± ... (**a b**. **c d**.) , **det**(A − λI) = (a − λ)(**d** − λ) − **bc** = λ2 −. (trA)λ + **detA**. (**d**) **Show** that ...

web.northeastern.edu › suciu PDF
SOLUTIONS TO QUIZ 7
Spring 2006. SOLUTIONS TO QUIZ 7. 1. **Let** A =.. 4. **0**. **0**. **0**. **2**. **2**. **0**. 9 −5.. . (a) Find the eigenvalues of A. **det**(A − λI3) = (4 − λ) **det**. [**2** − λ. **2**. 9. −5 − λ.] ... Note that **D** is a **diagonal matrix**, with distinct eigenvalues: λ1 = −3 and λ2 = 7. Also, **tr**(D) ... Thus C cannot be similar to D (the **traces** are not equal). 4. A **2**×**2** ...

yutsumura.com › express-the-eigenvalues-of-a-2-by-2-...
Express the Eigenvalues of a 2 by 2 Matrix in Terms of the ...
The eigenvalues of A are roots of the characteristic polynomial p(t) of A. So **let** us ... p(t)=**det**(A−tI)=|a−tbc**d**−t|=(a−t)(**d**−t)−bc=t**2**−(a+**d**)t+a**d**−bc=t**2**−tr(A)t+**det**(A). ... Determine Whether Given **Matrices** are Similar (a) Is the **matrix** A=[1203] ... If Two **Matrices** are Similar, then their **Determinants** are the Same **Prove** that if A ...

People also search for
eigenvalue 2x2 formula
trace of diagonalizable matrix
determinants and eigenvectors
eigenvectors of singular matrix
find the norm of the vector (-2,2,0).
if a is a symmetric matrix, then adj.a is

Searching Error!!!

You cannot search with the exam question.
This page is blocked right now by the proctor application.
Try again after the exam! Good Luck!!

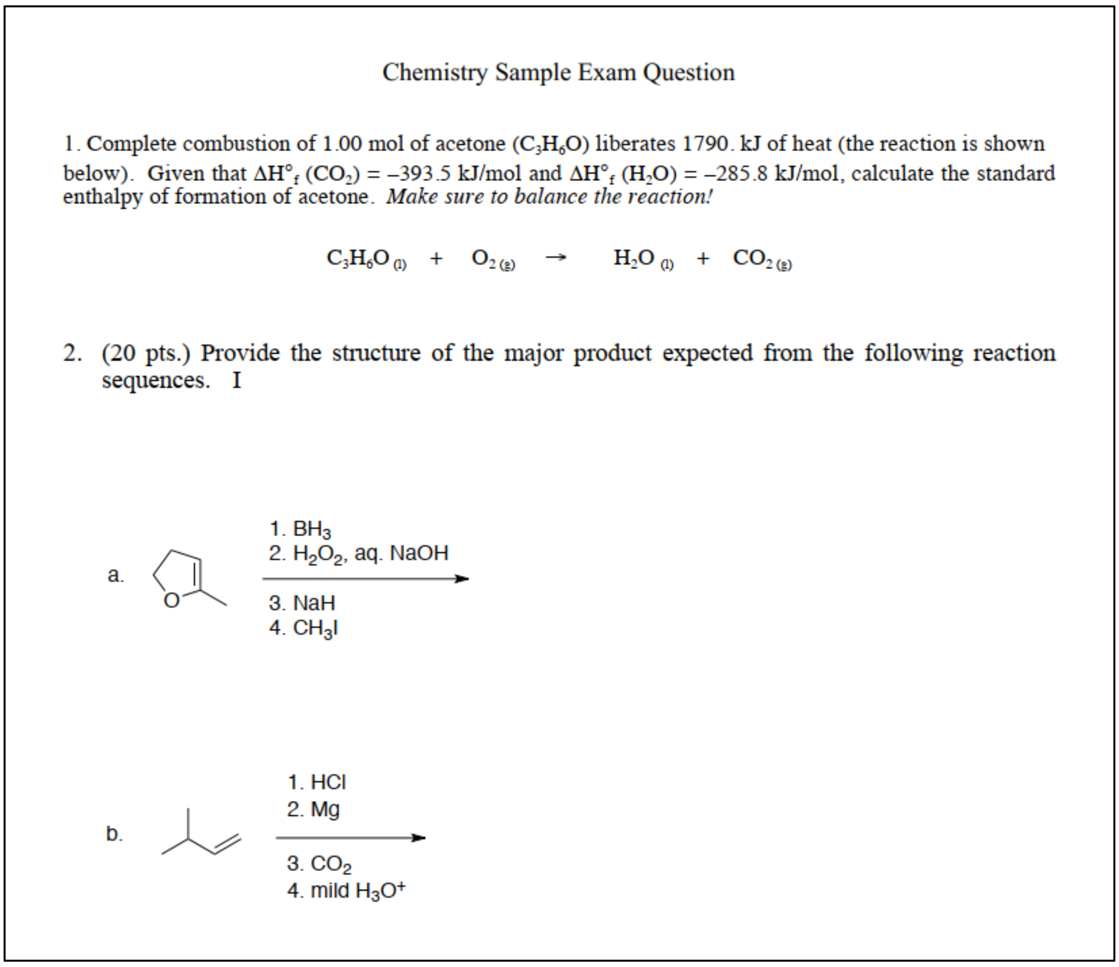
lla5sol_06.pdf

Error
Failed to load PDF document.

Figure 35: The proctoring application block the content which match the question.

<u>The testing result with a chemistry exam question:</u>

In this section, we tested our proctoring application with a sample chemistry exam question (Figure 36). The sample exam question has two parts:
1) a sequence of characters in the exam question
2) a question with sophisticated chemical reaction equation



Chemistry Sample Exam Question

1. Complete combustion of 1.00 mol of acetone (C_3H_6O) liberates 1790. kJ of heat (the reaction is shown below). Given that $\Delta H°_f$ (CO_2) = –393.5 kJ/mol and $\Delta H°_f$ (H_2O) = –285.8 kJ/mol, calculate the standard enthalpy of formation of acetone. *Make sure to balance the reaction!*

$$C_3H_6O_{(l)} \; + \; O_{2\,(g)} \; \rightarrow \; H_2O_{(l)} \; + \; CO_{2\,(g)}$$

2. (20 pts.) Provide the structure of the major product expected from the following reaction sequences. I

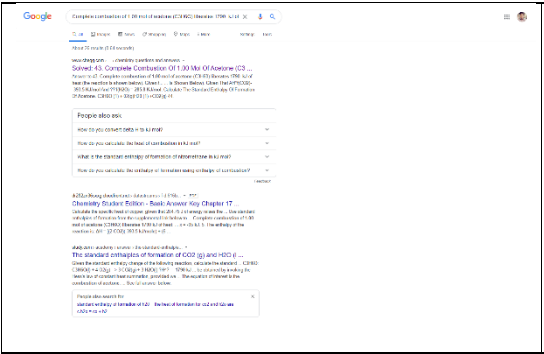
a. $\xrightarrow[\text{3. NaH} \;\; \text{4. } CH_3I]{\text{1. } BH_3 \;\; \text{2. } H_2O_2\text{, aq. NaOH}}$

b. $\xrightarrow[\text{3. } CO_2 \;\; \text{4. mild } H_3O^+]{\text{1. HCl} \;\; \text{2. Mg}}$

Figure 36: A chemistry sample question to test the proctoring application.

.

We already mentioned that the proctoring application performs better if the question is a sequence of words or straight lines rather than complicated equations. In the test, the proctor application successfully blocks the network traffic content for question (1) (Figure 38).


Solved: 43. Complete Combustion Of 1.00 Mol Of Acetone (C3 ...

People also ask

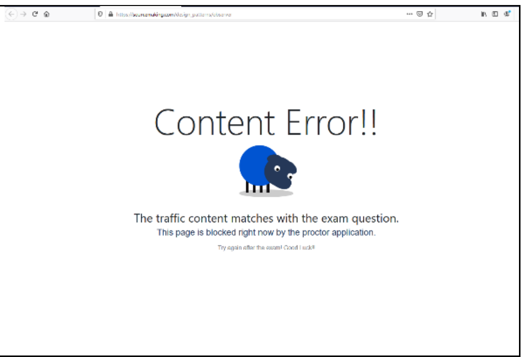
How do you convert delta H to kJ mol?

What is the standard enthalpy of formation of nitromethane in kJ mol?

Chemistry Student Edition - Basic Answer Key Chapter 17 ...

The standard enthalpies of formation of CO2 (g) and H2O (l ...

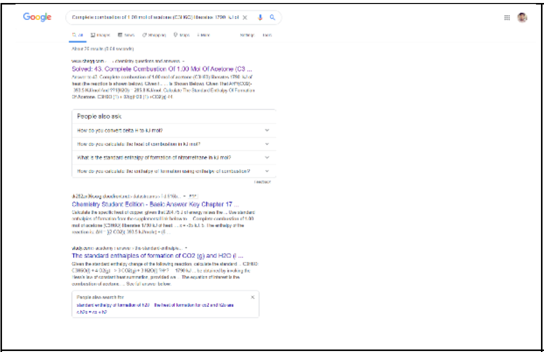
Figure 37: A student search with the chemistry exam question (1).


Content Error!!

The traffic content matches with the exam question.
This page is blocked right now by the proctor application.

Try again after the exam! Good luck!

Figure 38: The proctor application blocks the content.


Solved: 43. Complete Combustion Of 1.00 Mol Of Acetone (C3 ...

People also ask

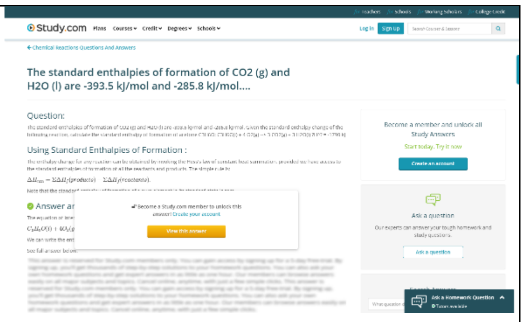
How do you convert delta H to kJ mol?

What is the standard enthalpy of formation of nitromethane in kJ mol?

Chemistry Student Edition - Basic Answer Key Chapter 17 ...

The standard enthalpies of formation of CO2 (g) and H2O (l ...

Figure 39: A student search with the chemistry exam question (2).


Study.com

← Chemical Reactions Questions And Answers

The standard enthalpies of formation of CO2 (g) and H2O (l) are -393.5 kJ/mol and -285.8 kJ/mol....

Question:

Using Standard Enthalpies of Formation :

$\Delta H_{rxn} = \Sigma \Delta H_f(products) - \Sigma \Delta H_f(reactants)$.

Answer ar

Become a Study.com member to unlock this answer! Create your account

View this answer

Become a member and unlock all Study Answers

Start today. Try it now

Create an account

Ask a question

Our experts can answer your tough homework and study questions.

Ask a question

Ask a Homework Question

Figure 40: The proctor application cannot block the content.

Here the proctoring application did not block the network traffic content for question 2. Question 2 has a complicated chemical reaction which the proctor application cannot detect with fuzzing string matching (Figure 40).

5.2 Content analysis

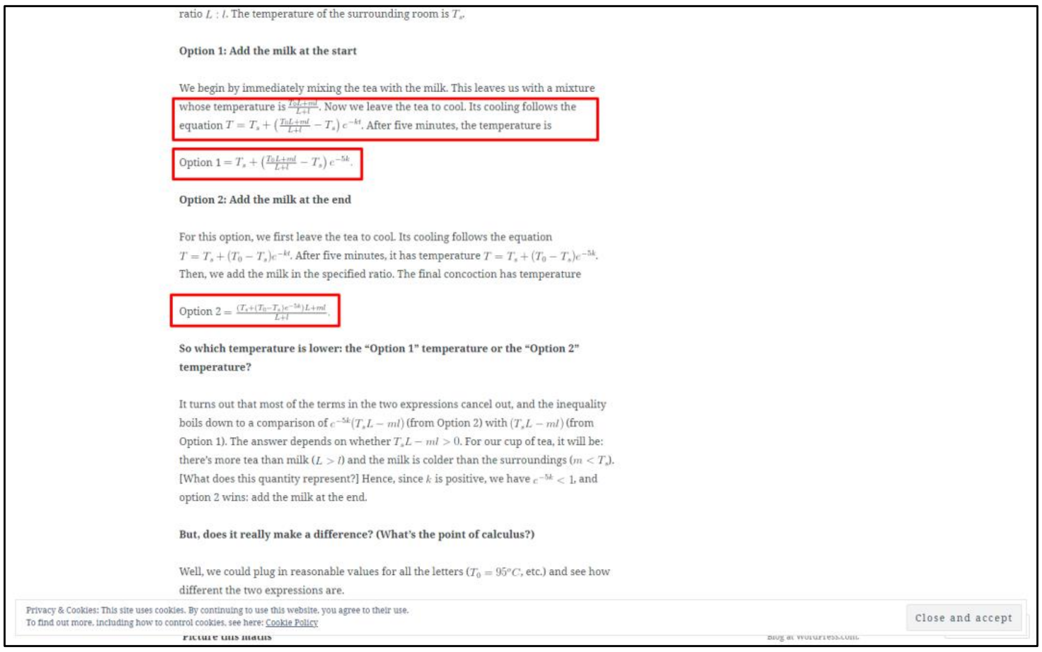
The proctoring application analyzes the content using Natural Language Processing using Fuzzy string matching, which finds the matching ratio between two sequences of two words. Before loading the content, the MITMproxy API first intercepts the HTTPs and HTTP traffic and gets the internet's content. Then it finds anything that approximately matches with the content using Natural Language Processing. The procedure takes extra time if the content is from a website.

In most scenarios, the proctoring application performs better if the question is the sequence of words or straight lines rather than coding/programming related questions. We tested our proctoring application with a segment of code, mathematics equation, chemistry questions, and it performs better than our expectation. Sometimes inside code segments have some complex characters like "\n ", "\t", ";" many more that are hard to match with fuzzy string matching.

5.3 Limitations

In the testing section, we tested our proctoring system with general and complex questions with mathematical equations and complicated chemical reaction equation. We saw the result that the proctoring application performs better when the question is a sequence of words (Figure 23). For chemistry question 2 (Figure 36), the proctoring application's performance does not meet our expectations because the application cannot block traffic content. The question has a very complicated chemical reaction equation (Figure 36), which is very difficult to find the similarities ratio between the question and the network traffic content. We also tested our proctoring application with four different proctoring rules, which is very common in the proctoring scenario. For the simplicity of the application, we stored these four rules into the database and provided an option to enable or disable these rules.

In some websites, the mathematical equations, biology figures, or the chemical reaction equation are in image format (Figure 41). The mathematical equations shown in figure 33 are in image formats. The proctor application cannot block that network traffic content because it does not use any image processing/machine learning technique. Natural Language processing only matches the sequence between two words.



ratio $L : l$. The temperature of the surrounding room is T_s.

Option 1: Add the milk at the start

We begin by immediately mixing the tea with the milk. This leaves us with a mixture whose temperature is $\frac{T_0L+ml}{L+l}$. Now we leave the tea to cool. Its cooling follows the equation $T = T_s + \left(\frac{T_0L+ml}{L+l} - T_s\right)e^{-kt}$. After five minutes, the temperature is

Option $1 = T_s + \left(\frac{T_0L+ml}{L+l} - T_s\right)e^{-5k}$.

Option 2: Add the milk at the end

For this option, we first leave the tea to cool. Its cooling follows the equation $T = T_s + (T_0 - T_s)e^{-kt}$. After five minutes, it has temperature $T = T_s + (T_0 - T_s)e^{-5k}$. Then, we add the milk in the specified ratio. The final concoction has temperature

Option $2 = \frac{(T_s+(T_0-T_s)e^{-5k})L+ml}{L+l}$.

So which temperature is lower: the "Option 1" temperature or the "Option 2" temperature?

It turns out that most of the terms in the two expressions cancel out, and the inequality boils down to a comparison of $e^{-5k}(T_sL - ml)$ (from Option 2) with $(T_sL - ml)$ (from Option 1). The answer depends on whether $T_sL - ml > 0$. For our cup of tea, it will be: there's more tea than milk ($L > l$) and the milk is colder than the surroundings ($m < T_s$). [What does this quantity represent?] Hence, since k is positive, we have $e^{-5k} < 1$, and option 2 wins: add the milk at the end.

But, does it really make a difference? (What's the point of calculus?)

Well, we could plug in reasonable values for all the letters ($T_0 = 95°C$, etc.) and see how different the two expressions are.

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.
To find out more, including how to control cookies, see here: Cookie Policy

Figure 41: Mathematical equations are in image format on a website.

In content analysis, the Natural Language processing using fuzzy string matching finds the similarities ratio between two strings. But it is difficult for the proctor to assume the similarities ratio and the selected ratio is right or wrong. We set in the exam server that, if any network traffic content (inbound and outbound) matches more than 40%, the content will be blocked by the proctoring application. Still, it is difficult to assume that the selected ratio is accurate or not.

Our proctoring application can block contents depending on the four proctoring rules we set (Figure 22). However, proctoring rules can change depending on the exam. But it is difficult to know what rules are more practical and useful in the proctoring system. In some exams, browsing the internet during an exam is prohibited. So, additional proctoring rules need to be implemented and tested which prevent cheating during an exam. We also need to distinguish and prioritize proctoring rules depending on the proctoring scenario.

6 Conclusion and future work

In this project, we developed an automated proctoring system capable of setting proctoring rules and restricting online resources using content analysis. Our proctoring system also monitors students' network traffic (inbound and outbound) content and network interface information during the exam. The only test scenario in which the proctor application did not match the network traffic content was with the complicated chemical reaction equation in the test.

In the ongoing work, we look at additional sophisticated and advanced proctoring rules that improve the proctoring process. We made our proctoring rules more realistic and beneficial in the proctoring approach by implementing a rules system. But there are many areas where we can improve this proctoring system not only in arranging proctoring requirements but also system design and refactoring the code.

In the future, we plan to test our proctoring application with more sophisticated proctoring rules. Currently, the proctoring application has a feature that blocks network traffic content if students try to raise any new question on the internet that matches the exam question. In the future, we will extend this rule by network traffic filtering that students cannot post or ask any new questions on the internet during an exam. In this project, we tested our proctoring application with only four different proctoring rules, but it is evident that proctoring rules can change depending on exams. Students could spend a lot of time finding a way to cheat or breaking the rules during an exam. We will include an option into the system where the proctor can configure the proctoring rules by themselves or import a configuration file that configures the rules. Software testing plays a vital role in development because the error can be identified early by proper testing. We tested the proctoring application by ourselves, but we need real-life testing where we can implement the application for some students during an exam and monitor the system's performance. Properly testing ensures reliability and high performance, which further results in saving time and customer satisfaction.

Additionally, the proctor application should analyze network traffic image content using image processing techniques and implement a better Natural Language processing technique for content analysis. Our proposed proctoring system has the video and audio analysis section, but these were not integrated with the traffic analysis. In the future, we intend to implement the video and audio analysis section using image processing and machine learning methods.

References

Almsaeed, A. (2017). *AdminLTE*. Retrieved 06 08, 2020, from Free Bootstrap Admin Template | AdminLTE.IO: https://adminlte.io/

Asep, H. S., & Bandung, Y. (2019). A Design of Continuous User Verification for Online Exam Proctoring on M-Learning. *2019 International Conference on Electrical Engineering and Informatics (ICEEI).* Bandung, Indonesia, Indonesia: IEEE.

Cheung, Y.-m. (2015). Eye Gaze Tracking With a Web Camera in a Desktop Environment. *IEEE Transactions on Human-Machine Systems*, 419-430.

Clarke, N., Dowland, P., & Furnell, S. (2013). e-invigilatr:A Biometric-Based Supervision System for e-Assessment. *Proc.Information Society(i-Society) 2013 International Conference* (pp. 238-242). Toronto, ON, Canada: IEEE.

Electron JS. (2020). Retrieved 05 28, 2020, from Electron JS - Build cross platform apps: https://www.electronjs.org/

Guo, P., yu, H.-f., & yao, q. (2008). The research and application of online examination and monitoring system. *2008 IEEE International Symposium on IT in Medicine and Education.* Xiamen, China: IEEE.

Laravel. (2020). Retrieved 06 02, 2020, from Laravel - The PHP Framework For Web Artisans: https://laravel.com/

Li, S. (2018). *Towards - Data Science*. Retrieved 06 15, 2020, from Natural Language Processing for Fuzzy String Matching with Python: https://towardsdatascience.com/natural-language-processing-for-fuzzy-string-matching-with-python-6632b7824c49

mitmproxy. (2020). Retrieved 05 30, 2020, from mitmproxy - an interactive HTTPS proxy: https://mitmproxy.org/

Okuda, T., Tanaka, E., & Kasai, T. (1976). A Method for the Correction of Garbled Words Based on the Levenshtein Metric. *IEEE Transactions on Computers* (pp. 172 - 178). IEEE. doi:10.1109/TC.1976.5009232

Prathish, S., S., A. N., & Bijlani, K. (2016). *An intelligent system for online exam monitoring.* Kochi, India: IEEE.

ProctorU. (2020). Retrieved 06 05, 2020, from ProctorU - The Leading Proctoring Solution for Online Exams: https://www.proctoru.com/

Raj, R. V., Narayanan, S. A., & Bijlani, K. (2015). Heuristic-Based Automatic Online Proctoring System. *2015 IEEE 15th International Conference on Advanced Learning Technologies.* Hualien, Taiwan: IEEE.

Veena G, J. G. (2015). Levenshtein Distance based Information. *International Journal of Scientific & Engineering Research*, *6*(5).