Department of Information Technology
NBA Accredited

A.P. Shah Institute of Technology

G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615

UNIVERSITY OF MUMBAI

Academic Year 2021-2022

A Project Presentation on

Web Content Sanitization

Submitted in partial fulfilment of the degree of
Bachelor of Engineering(Sem-6)
in

INFORMATION TECHNOLOGY

By

Parth Bhoir(19104050)
Pranav Mayekar(19104040)
Anjali Singh(20204006)

Under the Guidance of
Prof. Nahid Shaikh

1.Project Conception and Initiation

So many companies while starting there new projects or products find it difficult to gather data & spend huge capital to test their products. So coin planet provides a platform for new entrepreneurs & new business establishments to test their product on this platform.

On this platform through coin planet they can gather unbiased opinions & information through surveys & develop their resources accordingly.

So basically we will be gathering huge loads of data from the customers who are sharing their thoughts and their ideas about that particular product and some users might use some inappropriate text or data while sharing those feedback.

So to avoid that type of text or data or to exclude that type of responses from the other feedbacks web content sanitization comes in to the frame by detecting an inappropriate text/data we can protect the space of users who are daily engaging.

1.1 Objectives

- To create a menace free space for users.
- To detect any inappropriate content.
- To detect any inappropriate image/video.

1.2 Literature Review

- J Weinberger and P Saxena said Filtering or sanitization is the predominant mechanism in today’s applications to defend against cross-site scripting (XSS) attacks
- We study several emerging web application frameworks including those presently used for development of commercial web applications. We evaluate how effective these frameworks are in guarding against the common pitfalls of sanitization (UCB/EECS-2011-11)
- We find that while some web frameworks safeguard against the empirically relevant use cases, most do not. In addition, some of the security features in present web frameworks provide a false sense of security.

- Davide Balzarotti and Marco Cova proposed to validate sanitization in web application by composing static and dynamic analysis in 2008 IEEE symposium on security and privacy.

- Researchers have proposed many novel defenses, ranging from purely server-side to browser-based or both. Sanitization or filtering, the practice of encoding or eliminating dangerous constructs in untrusted data, remains the predominant defense strategy in existing applications.

1.3 Problem Definition

- The content we are receiving for our website should not contain any inappropriate text/content because of that our website engagement will be less so it is necessary for us to protect the other users views and to provide a user secured user space.
- So the Input sanitization describes cleansing and scrubbing user input to prevent it from jumping the fence and exploiting security holes.

1.4 Scope

- It will reduce online hatred.
- It provides menace free platform.
- It will help reduce toxic content from being brodcasted.

1.5 Technology stack

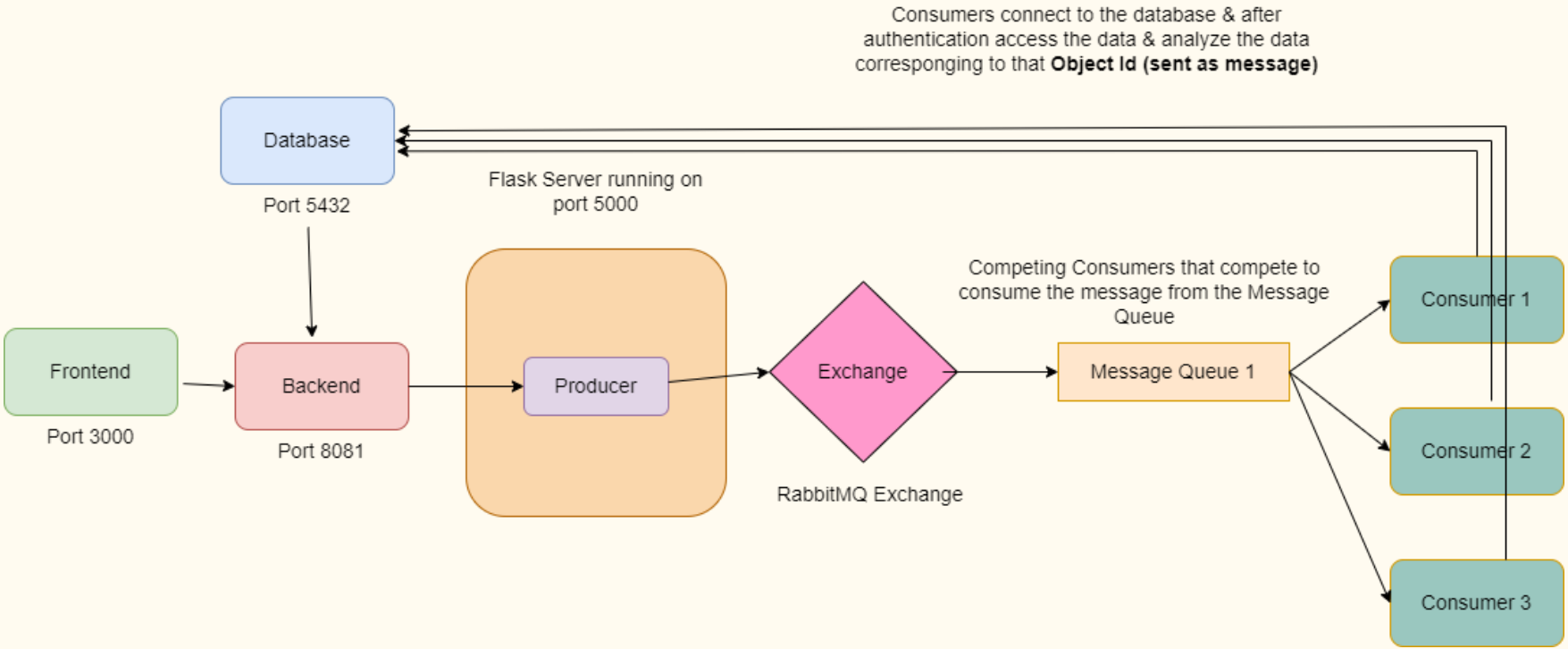
- React for frontend
- Node for backend
- PostgreSQL – we will be using PostgreSQL for our website database
- RabbitMQ: RabbitMQ is lightweight and easy to deploy on premises and in the cloud. It supports multiple messaging protocols.
- Python: Flask to create a standalone server for receiving machine learning request.

2. Project Design

2.1 Proposed System

- Greater efficiency
- To detect inappropriate content for other users
- To protect others user space.
- Proper analysing of text and data content

2.2 Design and flow of modules



3. Implementation

```
C:\Program Files\rabbitmq-server-windows-3.9.14\rabbitmq_server-3.9.14\sbin>rabbitmq-server.bat
2022-04-24 11:37:06.983000+05:30 [info] <0.228.0> Feature flags: list of feature flags found:
2022-04-24 11:37:06.998000+05:30 [info] <0.228.0> Feature flags:   [x] implicit_default_bindings
2022-04-24 11:37:06.998000+05:30 [info] <0.228.0> Feature flags:   [x] maintenance_mode_status
2022-04-24 11:37:06.998000+05:30 [info] <0.228.0> Feature flags:   [x] quorum_queue
2022-04-24 11:37:06.998000+05:30 [info] <0.228.0> Feature flags:   [x] stream_queue
2022-04-24 11:37:06.998000+05:30 [info] <0.228.0> Feature flags:   [x] user_limits
2022-04-24 11:37:06.998000+05:30 [info] <0.228.0> Feature flags:   [x] virtual_host_metadata
2022-04-24 11:37:06.998000+05:30 [info] <0.228.0> Feature flags: feature flag states written to disk: yes
2022-04-24 11:37:08.092000+05:30 [noti] <0.44.0> Application syslog exited with reason: stopped
2022-04-24 11:37:08.092000+05:30 [noti] <0.228.0> Logging: switching to configured handler(s); following messages may no
t be visible in this log output

  ##  ##      RabbitMQ 3.9.14
  ##  ##
  ##########  Copyright (c) 2007-2022 VMware, Inc. or its affiliates.
  ######  ##
  ##########  Licensed under the MPL 2.0. Website: https://rabbitmq.com

  Erlang:      24.3.2 [jit]
  TLS Library: OpenSSL - OpenSSL 1.1.1d  10 Sep 2019

  Doc guides:  https://rabbitmq.com/documentation.html
  Support:     https://rabbitmq.com/contact.html
  Tutorials:   https://rabbitmq.com/getstarted.html
  Monitoring:  https://rabbitmq.com/monitoring.html

  Logs: <stdout>
        c:/Users/Parth979/AppData/Roaming/RabbitMQ/log/rabbit@LAPTOP-M1PF0PK8.log
        c:/Users/Parth979/AppData/Roaming/RabbitMQ/log/rabbit@LAPTOP-M1PF0PK8_upgrade.log

  Config file(s): (none)
```


Running the RabbitMQ Server

```
More? python run_consumer.py CONSUMER_1
Connecting to the PostgreSQL database...
The surrent postgresql version is RealDictRow([('version', 'PostgreSQL 13.3, compiled by Visual C++ buil
d 1914, 64-bit')])
============== Consumer 'CONSUMER_1' is up & running ==============
```

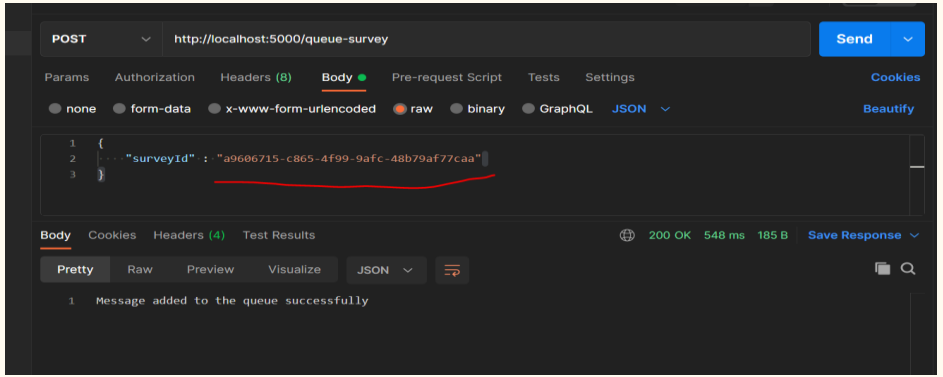
Running the consumer

```
More? python run_consumer.py CONSUMER_2
Connecting to the PostgreSQL database...
The surrent postgresql version is RealDictRow([('version', 'PostgreSQL 13.3, compiled by Visual C++ buil
d 1914, 64-bit')])
============== Consumer 'CONSUMER_2' is up & running ==============
```

Running Another Consumer

```
More? python run_server.py
 * Serving Flask app 'server' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 152-569-870
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Running the Flask Server



Send Request To Flask Server

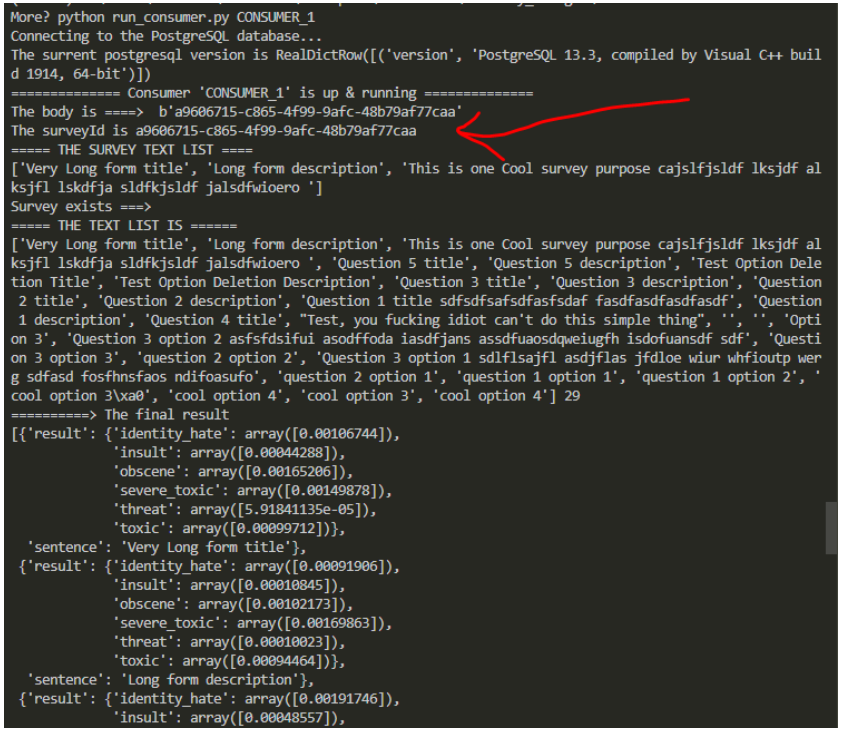
```
SELECT * FROM public."Questions" WHERE survey_id=CAST('a9606715-c865-4f99-9afc-48b79af77caa' as uuid);
```

Data Output | Explain | Messages | Notifications

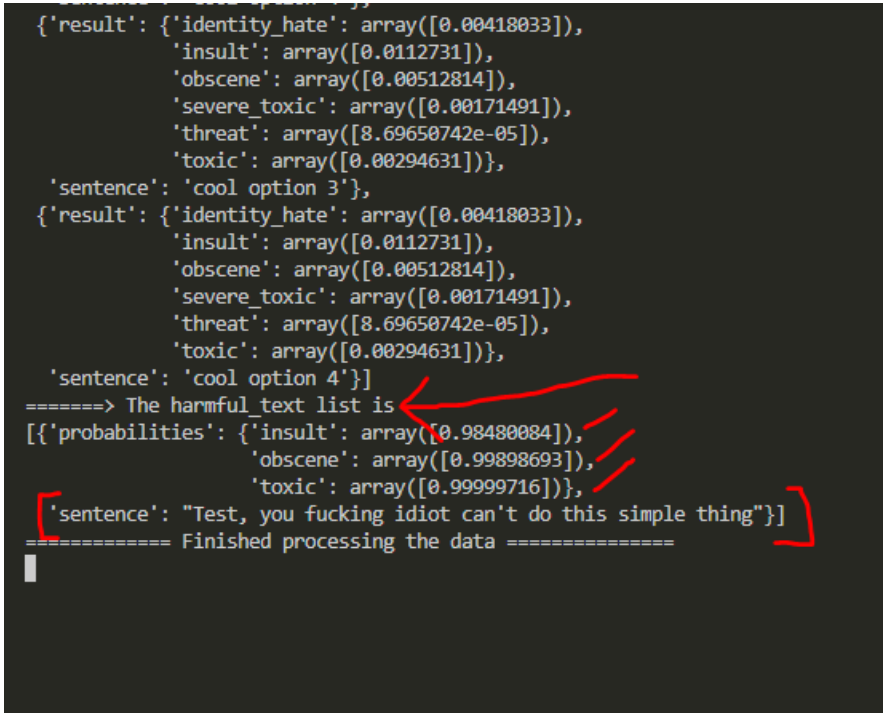
	id [PK] uuid	title character varying (255)	description character varying (300)	ques_img character varying (400)	ques_video character varying (400)	ques_audio character varying
1	1b0fede5-b1...	Question 4 title	Test, you fucking idiot can't do...			
2	13fa4dcd-b5...	Question 5 title	Question 5 description			
3	7950a4b1-17...	Test Option Deletion Title	Test Option Deletion Descripti...			
4	f0ec6990-a7...	Question 3 title	Question 3 description			
5	4a6d9b57-3b...	Question 2 title	Question 2 description			
6	2840ba65-a5...	Question 1 title sdfsdfsafsdfa...	Question 1 description			

Data in the database using pgadmin

5. Result



Request is received by the consumer & it starts processing



After processing we get the result which sentence is toxic & what is the probability

6. Conclusion and Future Scope

Conclusion and Future scope

- During the course of this project, we were able to achieve text content sanitization & exclude inappropriate text & data for our website environment by using logistic regression. So, we hope that in the future this will help to reduce toxic & inappropriate content for text on English Language for now.
- We hope that in future this will help to reduce toxic and inappropriate content for images and videos as well.

References

•[1] Robertson C, Mele C, Tavernise S. 11 Killed in Synagogue Massacre; Suspect Charged
•With 29 Counts. 2018;.

•[2] Hate Speech ABA Legal Fact Check-American Bar Association;. Available
•from: https://abalegalfactcheck.com/articles/hate-speech.html.

•[3] Ross B, Rist M, Carbonell G, Cabrera B, Kurowsky N, Wojatzki M. Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis. In: The 3rd Workshop on Natural Language Processing for Computer-Mediated Communication @ Conference on Natural Language Processing; 2016

•[4] Pedregosa F, Varoquaux G, Gramfort A. Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. JMLR. 2011:12:2825-2830.

•[S] Opitz D, Maclin R. Popular ensemble methods: An empirical study. Journal of artificial intelligence research. 1999;11:169-198.

Thank You