

Detecting Malicious Social Bots Based on Clickstream Sequences

Selvarani S¹, Sahana B.R²

¹M-Tech Student, CSE, Maharaja Institute of Technology

²Assistant Professor, Department of CSE, Maharaja Institute of Technology

ABSTRACT

The crucial part development in amount, pace, and range of user data (such Information created by users) web-based social networks has led to attempts to develop new techniques for acquiring and reviewing such massive data. For instance, social bots till now used to offer users higher-quality customer service while carrying out automated analytical tasks. Harmful social bots, such as fake news, have circulated false information, which has had an effect on the real world. Consequently, it is crucial to identify and get rid of risky social bots from online social networks. Based on their characteristics, the bulk of currently employed detection methods for risky social robots focus concerning the numeral aspects. Low analytical accuracy is caused by the ease with which social bots can mimic these characteristics. This article provides a word of art feature method based to the shifting likelihood of clickstream sequences and semi-supervised clustering for the detection of harmful social bots. This method considers both the transition probability of stream of click by the user analysis and the temporal component of activity. The detection accuracy for various types of malicious social bots based on transition probabilities of clickstreams increases by an average of 12.8%, according to the results of our tests on actual platforms for online social networks. This is in contrast to detection methods based on examination of user behavior by employing stats.

INTRODUCTION

Online social accounts known as "social bots" are reduced by computer set of commands that may do particular tasks in accordance with a bunch of criteria. With the rising use of mobile devices (such as Android and iOS smartphones), both the frequency and the kind of user participation on social networks grew. This is demonstrated by the large volume, velocity, and variety of data that the sizable online social network user base generates. Social bots have been widely employed to increase the efficacy and accuracy of data collection and analysis from social network services. For instance, the social bot SF QuakeBot can provide earthquake alerts for the San Francisco Bay area by scanning social media for real-time information about earthquakes. However, social networks are viewed by the public as potentially being exploited for bad or malicious ends, and they gather enormous amounts of user data. Automatic social bots are frequently perceived as bad because they cannot effectively reflect the genuine intents and desires of everyday users on online social networks.

SOCIAL BOTS DETECTION

On wired and wireless networks, botnets are becoming more and more prevalent. Particularly in a botnet, bots can cooperate to achieve a single evil purpose. Social bots have become very popular

recently since they can imitate human behavior in social networks. They are also taught how to work together to complete the tasks that have been given to them. Some people use a variety of technology for malicious or bad purposes, including sophisticated strategies and tools that may be linked to nation governments and state-sponsored actors as well as social bots. For example, social bots may 'crawl' for words and images from online social networks to fill out false user profiles and carry out other jobs in order to accurately mimic the characteristics of genuine users. Social networks have reportedly witnessed the birth of highly complex social bots that display characteristics of both human and social bot behavior. Between social bots and humans, these semi-social bots exist. Usually, humans start the automated process for a semi-social bot, whereas social bots handle the subsequent duties automatically.

EXISTING SYSTEM:

- Morstatter et al. recommended a supervised Boost OR model of the heuristic kind with a rising recall rate for the purpose of identifying harmful bots. This model takes into account the ratio of tweeted messages to those that have been posted on Twitter, the typical tweet length, the URL, and the gap between forwarded tweets.
- Wang et al. created a semi-supervised clickstream similarity graph user behavior model to find suspicious accounts in Renren. A supervised machine learning approach was proposed to identify social bots based on age, location, and other static features of active, passive, and inactive users in Twitter, as well as interacting person, interacting content, interacting theme, and some dynamic characteristics, in order to identify the active, passive, and inactive users based on social interactions between users of the platform.

DISADVANTAGES OF EXISTING SYSTEM:

- Massive volumes of data must be annotated and trained on in order to use supervised learning. In general, as it requires time and money, tagging data is inappropriate for the big data social networking context. To put it another way, a strategy like this is often inappropriate for real-time detection of risky social bots on social networking sites.
- Unsupervised learning, however, does not require manual data labeling. On the other hand, unsupervised learning techniques are sensitive to initial values and can only distinguish unique outcomes. It is impossible to discern between normal and abnormal clusters.

PROPOSED SYSTEM:

- We design an algorithm for detecting malicious social bots based on spatiotemporal features in this paper to detect malicious social bots on social network platforms in real-time. The algorithm for detecting malicious social bots is based on the spatiotemporal features we propose in this paper as the transition probability features between user clickstreams.
- We look at user behavior data to determine transition probability features between user clickstreams in order to more accurately detect hazardous social bots in online social networks. The detection of social bots using a semi-supervised space-time feature-based approach is proposed. This approach is based on time interval characteristics and transition probability features.

ADVANTAGES OF PROPOSED SYSTEM:

- We then evaluate and characterize situational awareness user activities in social networks by employing the semisupervised clustering detection approach we've proposed. This allows us to swiftly identify harmful social bots with the aid of just a small number of tagged individuals.
- We examine how users behave in social scenarios on online social networks in order to quickly identify potentially dangerous social bots. We evaluate user behavior components and select the user behavior transition probability based on general behavioral attributes. We then evaluate and characterize situational awareness user activities in social networks by employing the semisupervised clustering detection approach we've proposed. This allows us to swiftly identify harmful social bots with the aid of just a small number of tagged individuals.

METHODOLOGY

Developers must pay special attention to the input design process because it is a critical stage in the life cycle of software development. The goal of the input design is to give the program the most precise data possible. So that feeding errors are kept to a minimal, inputs must be properly prepared. The input forms or screens, according to Software Engineering Concepts, are designed to permit validation control over the input limit, range, and other necessary validations.

This system has input screens for almost all of its modules. Error messages are intended to alert users when they commit errors and direct them appropriately to prevent entering incorrect data. Under module design, let's look at this in more detail.

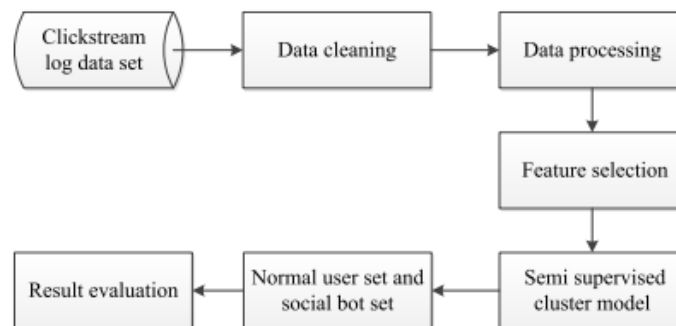


FIG 1.: EXPERIMENT PROCEDURE.

The user's clickstream data set was acquired, and then several operations were completed, including data set cleaning and screening, feature processing, data classification, and data classification. The detailed steps are shown in Figure 2.

1) Data cleaning: In order to remove erroneous data, calculate acceptable transition probabilities between clickstreams, and minimize transition probability mistakes caused by fewer data, less data must first be cleaned.

2) Data processing: From a pool of typical users, some data are randomly selected, and social bots are programmed with the label. The label for the normal user account is "1," whereas the label for the social bots account is "1." Under the category of clusters are users of seeds.

3) Feature selection: based on the core capabilities of the CyVOD platform, we select the transition probability properties connected to the playback function in the spatial dimension: Using P(play,play), P(play,like), P(play,feedback), P(play,comment), P(play,share), and P(play,more), we can determine the

inter-arrival times (IATs) in the time dimension. Because extremely high data sizes and sparse matrices could make data detection more difficult if all transition probability matrices of user activity are constructed. In the first step of the semi-supervised clustering process, labeled seed users determine the initial centers of two clusters. The clustering findings are then iterated over and improved using unlabeled data.

5) Obtain the set of typical users and the set of social bots: Detection can be utilized to ultimately obtain the set of typical users and the set of social bots.

6) Analysis of the findings: Precision, Recall, and F1 Score—the harmonic average of Precision and Recall, which equals $F1 = 2 \frac{Precision \times Recall}{Precision + Recall}$ —are the three distinct metrics we use to evaluate the results. While we wait, we assess the approach's performance using Accuracy and compare it to the SVM technique. Accuracy is defined as the percentage of samples that the classifier correctly identifies out of all samples.

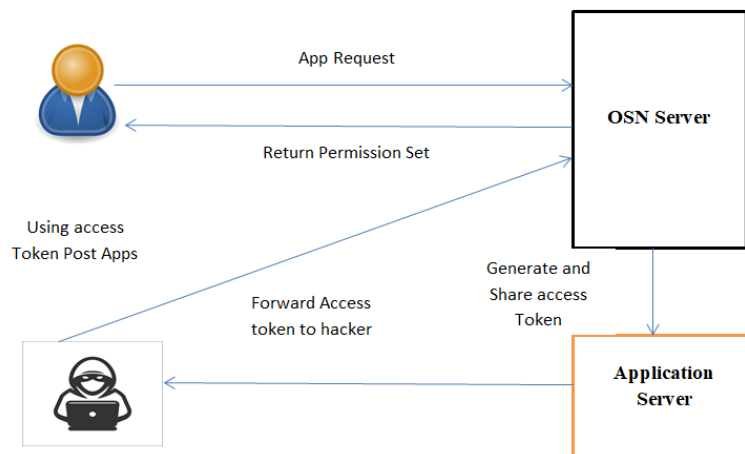


FIG 2:SYSTEM ARCHITECTURE

1. THE COLLECTION OF DATA

The cyvoD platform is made up of the website platform, Android, and iOS applications. In order to analyze user clickstream activity on CyVOD, a data burial point is used, and server-side data gathering takes place. However, you must connect directly with other websites or use their relevant API (if one is available) in order to collect the required data from other websites. For your own website, you can employ hidden technologies to obtain the required data in a realistic context.

2. RESEARCH DESIGN

Wicked social bots that work together to complete tasks, malicious social bots that handle numerous tasks, and social bots that focus on a single task. For example, a user can perform numerous things in the actions like "liking," "commenting," "sharing," and so forth. The social bot's P(play, like) value (transition probability of "the current click event is and the next click event is liking") would be high for detrimental likes and low or nonexistent for other transition probability features.

MODEL PHASES

The waterfall model is a sequential software development process where progress is seen as continuously flowing downward (like a waterfall) through the processes of requirement generation, analysis, design, implementation, testing, and maintenance.

REQUIREMENT ANALYSIS: Gathering the requirements for the system is part of the requirement analysis stage. This process includes processes like document creation and requirement analysis.

SYSTEM DESIGN: Keeping the needs in mind, the system specifications are transformed into a software representation. In this stage, the designer spends a lot of emphasis on things like software architecture, algorithms, and data structures.

CODING: At this point, the programmer starts to write code to build a finished product sketch. To put it another way, system requirements are solely used to generate machine-readable compute code.

IMPLEMENTATION: During the implementation stage, the software itself is programmed or coded. Software libraries, executables, user manuals, and other documentation are frequently produced as a result of this stage.

TESTING: All programs (models) are integrated and tested at this step to make sure the system as a whole conforms with the software specifications. A major focus of testing is verification and validation.

MAINTENANCE: The most prolonged step, maintenance include upgrading the program to satisfy shifting client needs, adapt to environmental changes, correct errors and oversights missed during testing, and boost the software's effectiveness.

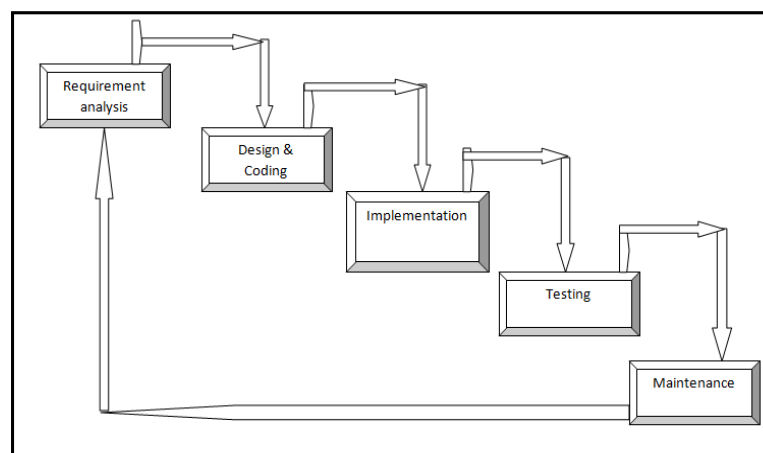


FIG 3:- WATERFALL MODEL

RESULTS AND DISCUSSION

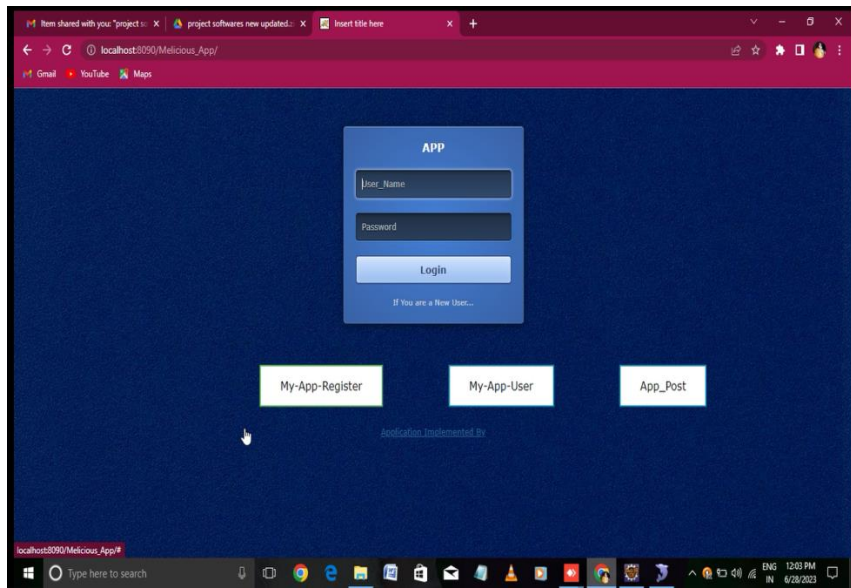


FIG 8: LANDING PAGE

First step is Landing into the landing page by signing in to the localhost:8090. If the user is the new user then he/she should click on the the “If you are a New User” as shown in fig 9 and we have to enter the appropriate datafor creating an account.

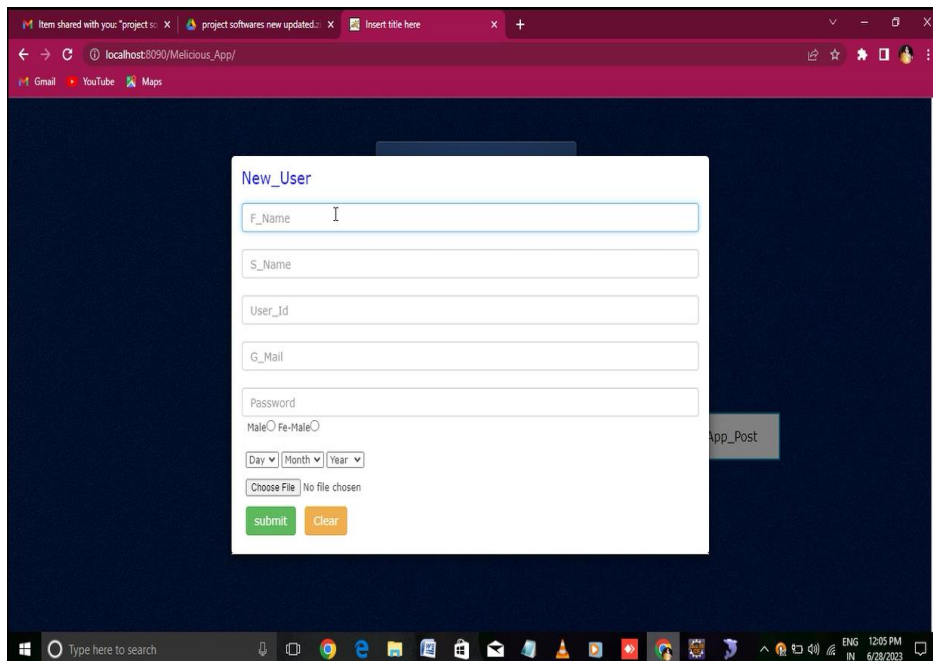


FIG 9: PAGE TO CREATE AN ACCOUNT FOR NEW USER

The below Fig 10 shows the page to login after creating a new account with the new user credentials(by giving his/her username and password).

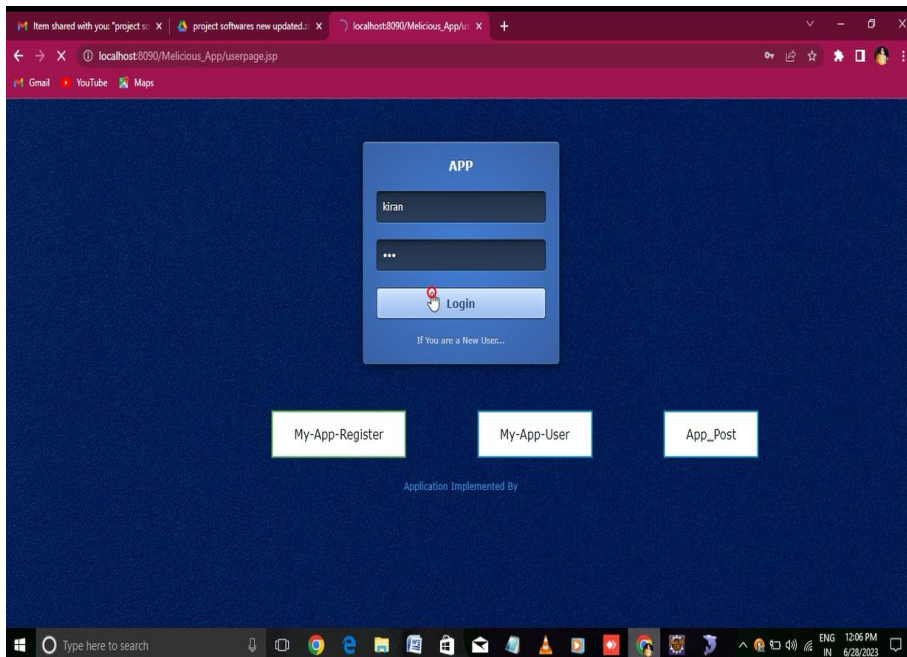


FIG 10: LOGIN PAGE

The below FIG 11 shows the landing page of a new user. The page is consisting of profile picture and name of the new user, along with that it is having some buttons like Find Friends, View Friend Request, Upload image, Friend List, My Tweet, My New Apps and logout.

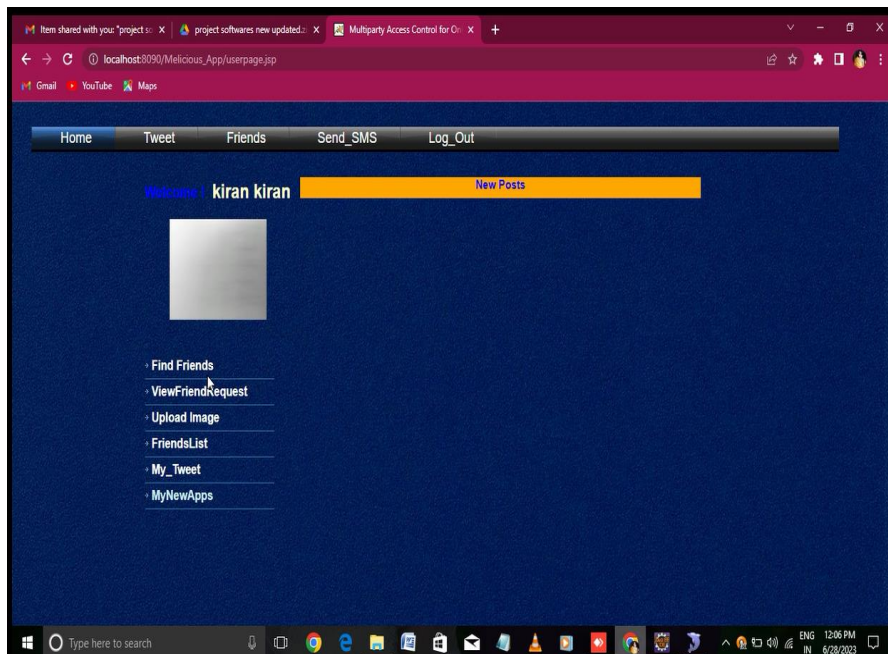


FIG 11: SHOWS THE LANDING PAGE OF A NEW USER

The below Fig 12 shows the user after sending a friend request to his friend. After sending a friend request it will show like "Waiting for Approve". He can give request to one or more friends.

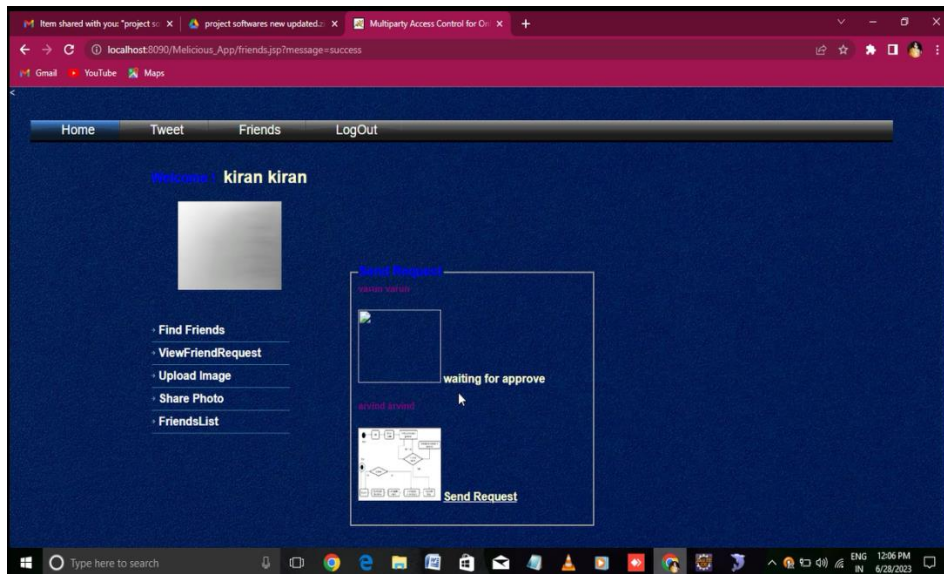


FIG 12: 12 SHOWS THE USER AFTER SENDING A FRIEND REQUEST TO HIS FRIEND

The below Fig 13 shows another person logging in with his credentials.

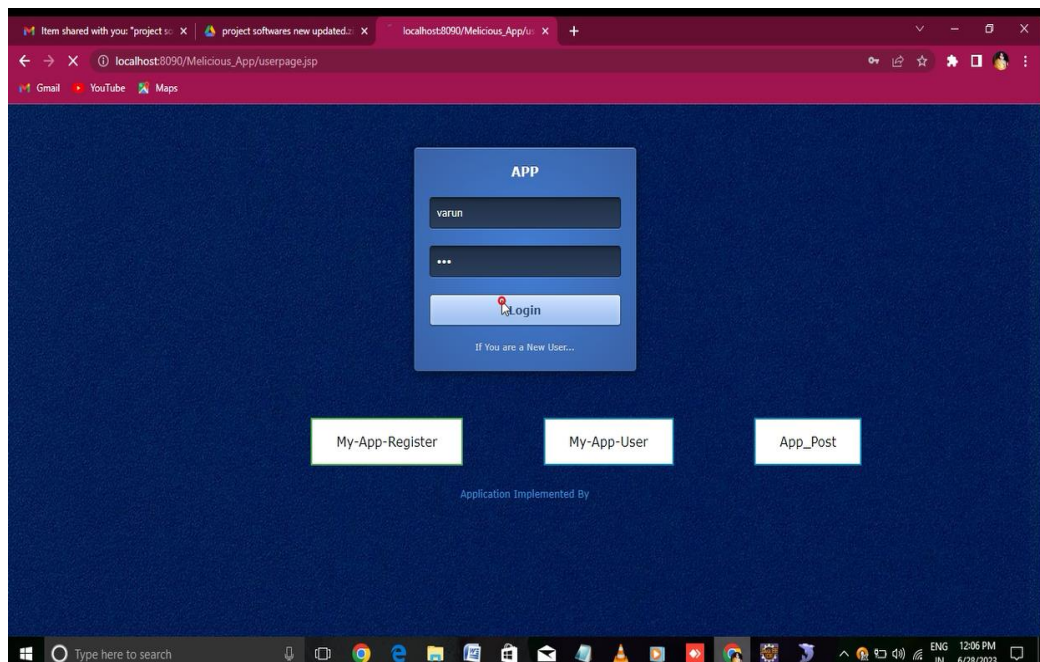


FIG 13: ANOTHER PERSON LOGGING IN TO HIS ACCOUNT

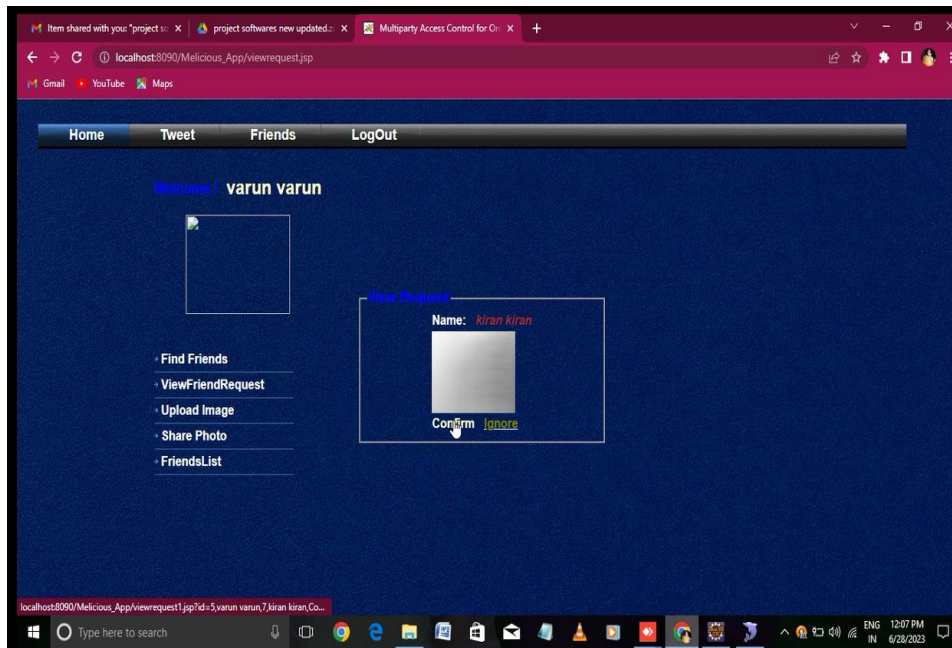


FIG 14: CONFIRMING THE FRIEND REQUEST

In fig 14, the new user's friend accepting the request . In fig 15 , when the new user sees his notification he got the confirmation of his request.

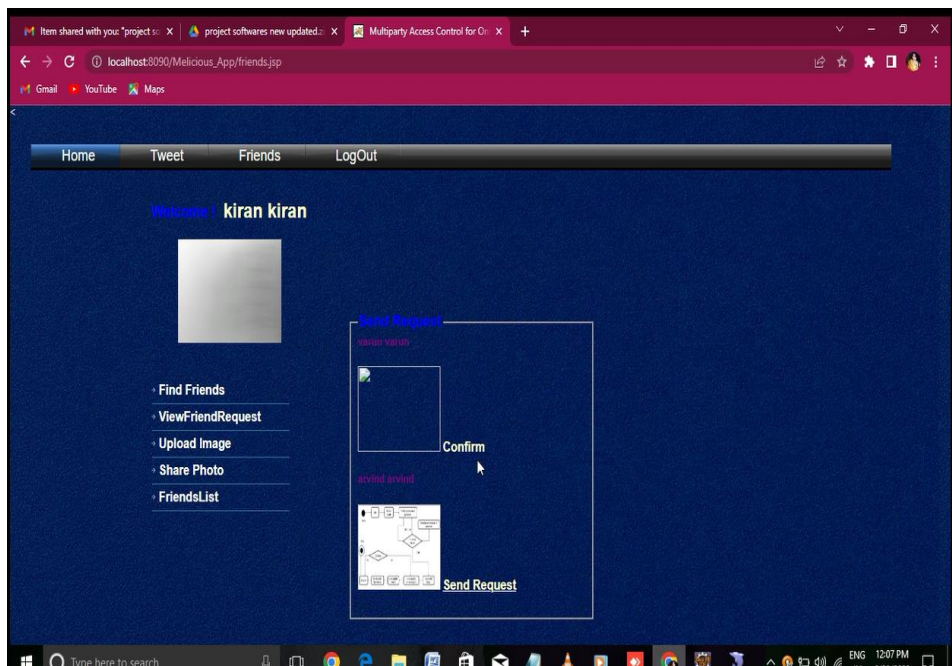


FIG 15: SHOWING THAT THE FRIEND REQUEST HAS BEEN ACCEPTED BY HIS FRIEND

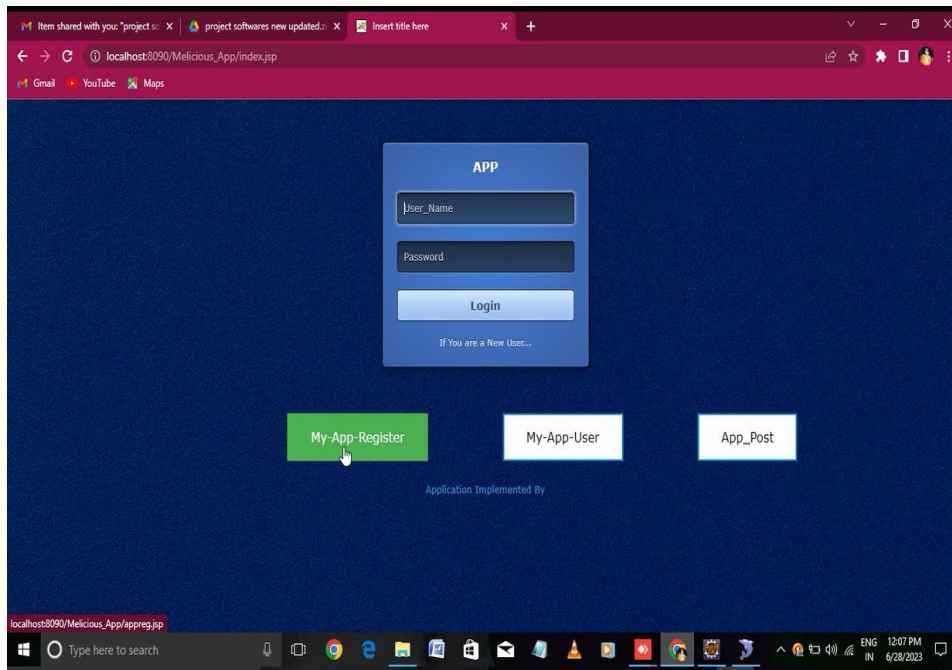


FIG 16: REGISTERING IN MY APP REGISTER

In fig 16, the developer who wants to publish his app will go on with My App Register and in fig 17, the landing page of member register can be found.

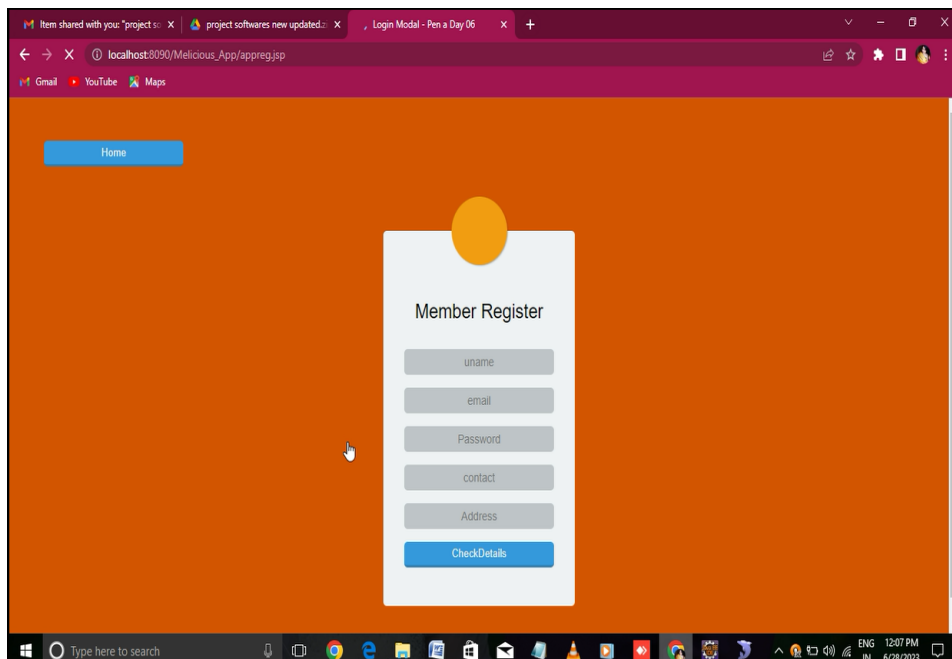


FIG 17: REGISTER LANDING PAGE

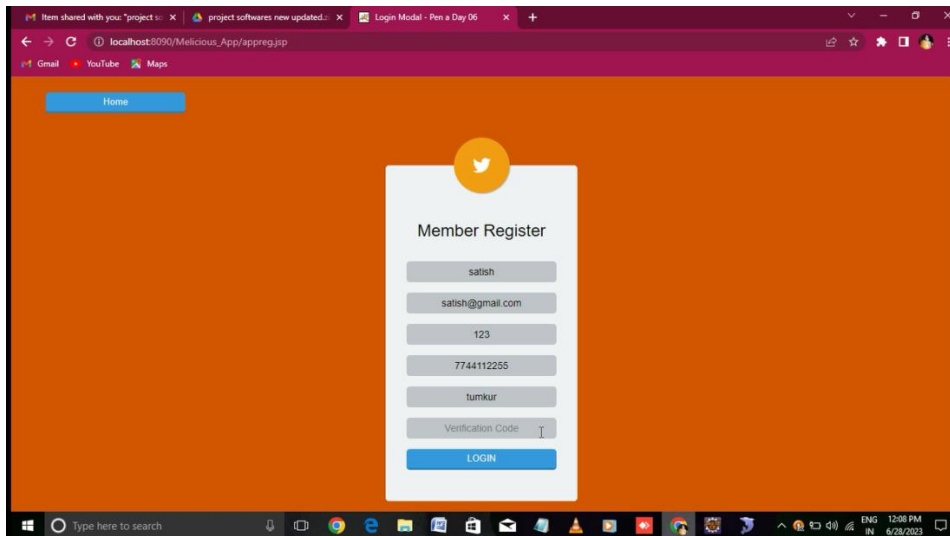


FIG 18: ASKING FOR VERIFICATION CODE

In fig 18 , its asking for verification code to register to My App User and In fig 19 it shows getting verification code in Tomcate . If giving the verification only, the developer can register into My App user app.

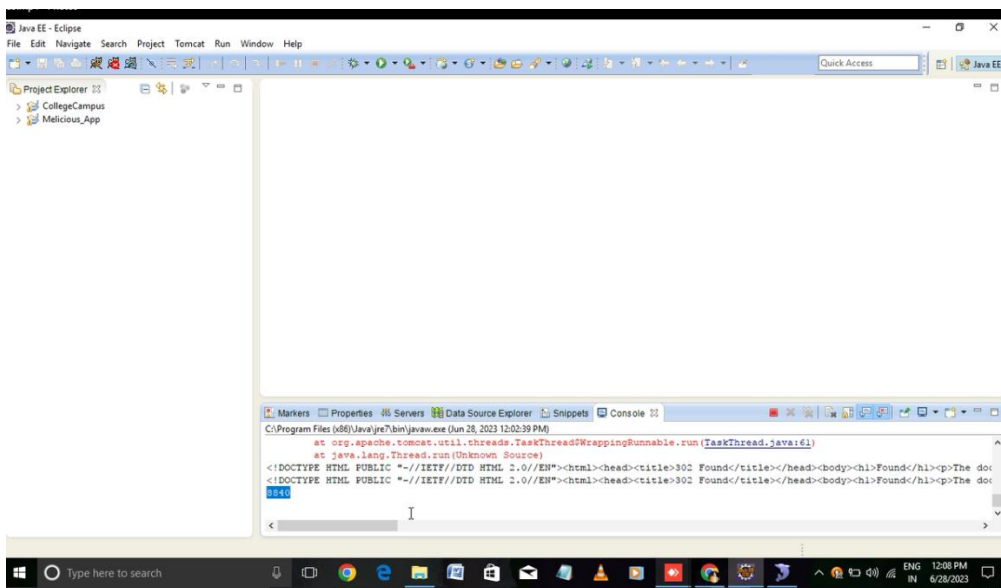


FIG 19: GETTING VERIFICATION CODE IN TOMCAT

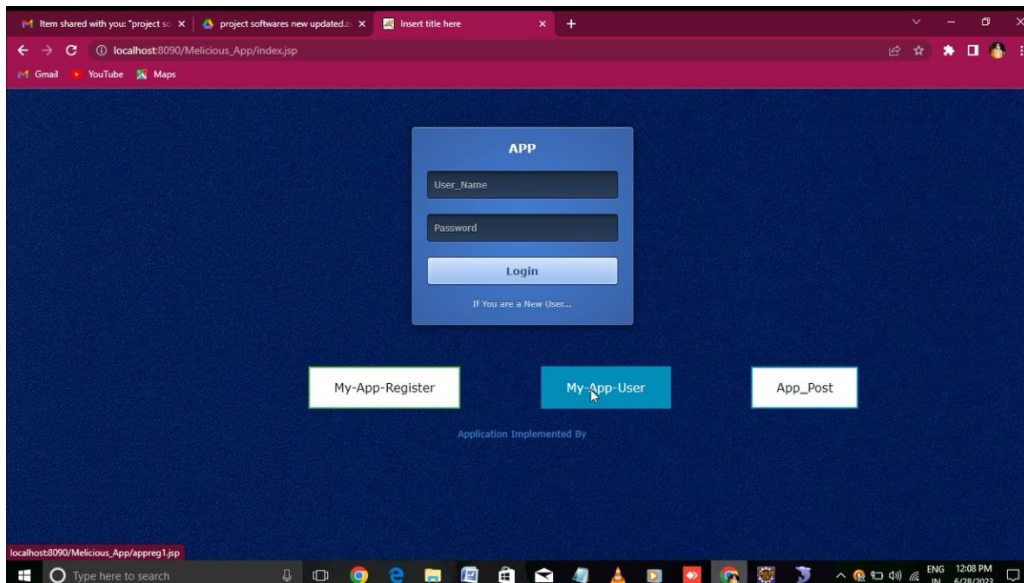


FIG 20: CLICKING ON “MY APP USER” TO LOGIN BY DEVELOPER

After completing registration, developer logging in to my app user in fig 20. In fig 21 it shows the landing page of my app user. Here the developer can login with his login credentials.

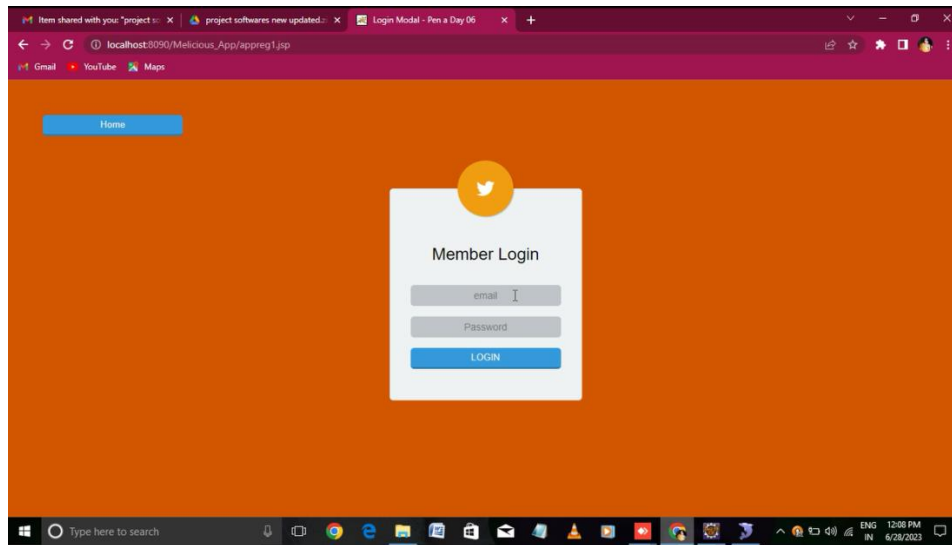


FIG 21: LOGIN PAGE OF MY APP USERS

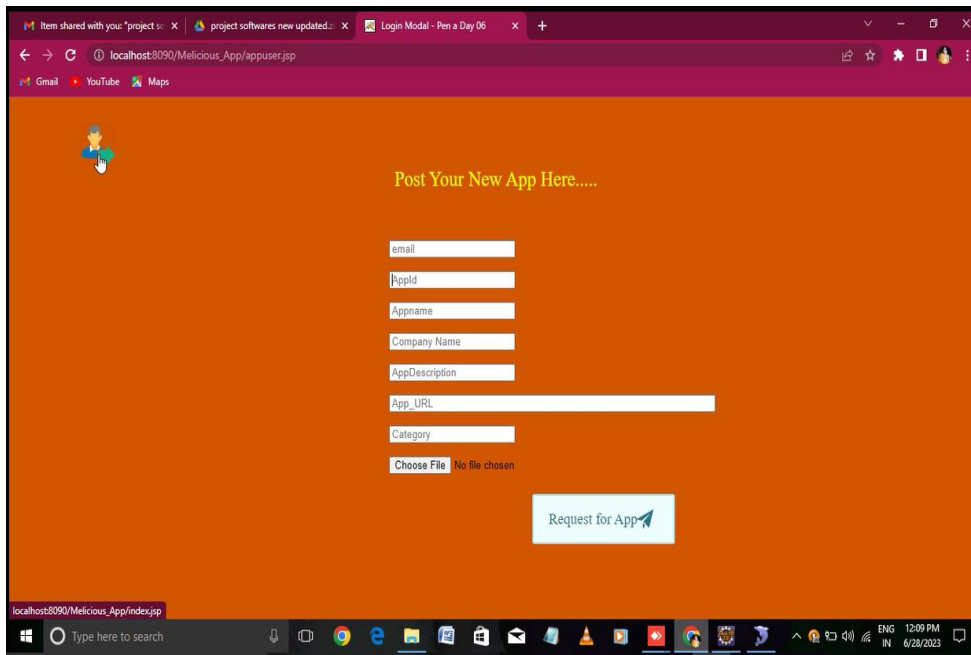


FIG 22: MY APP USER APP NEEDS APP ID TO SHARE THE APP

In landing page of My App User, it needs AppID. With out that developer can't share his App. To fetch the AppID, the admin logs to his account by giving his login credentials.

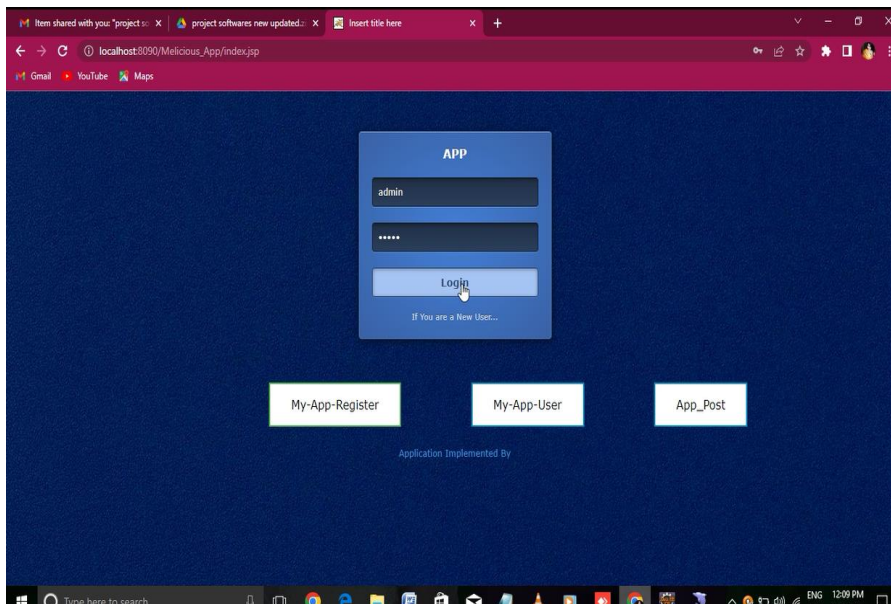
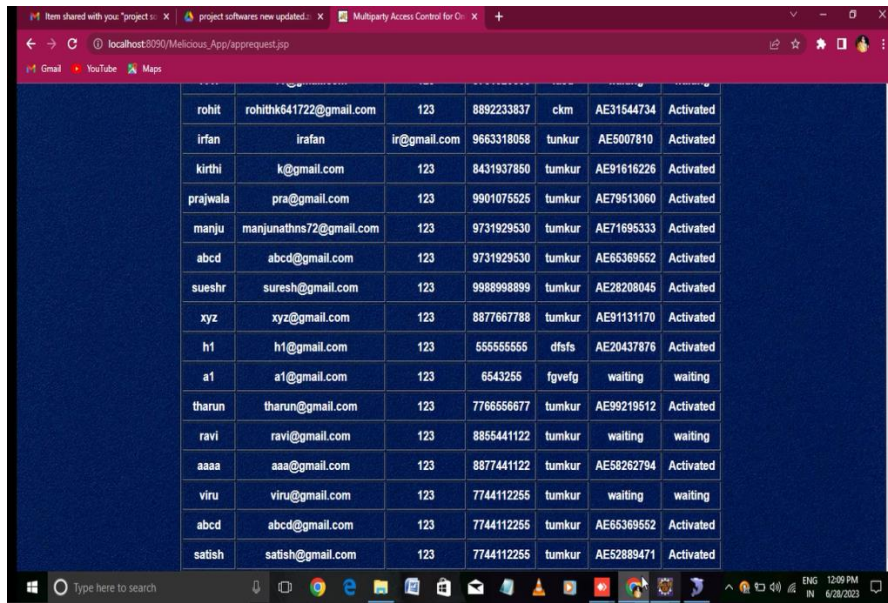


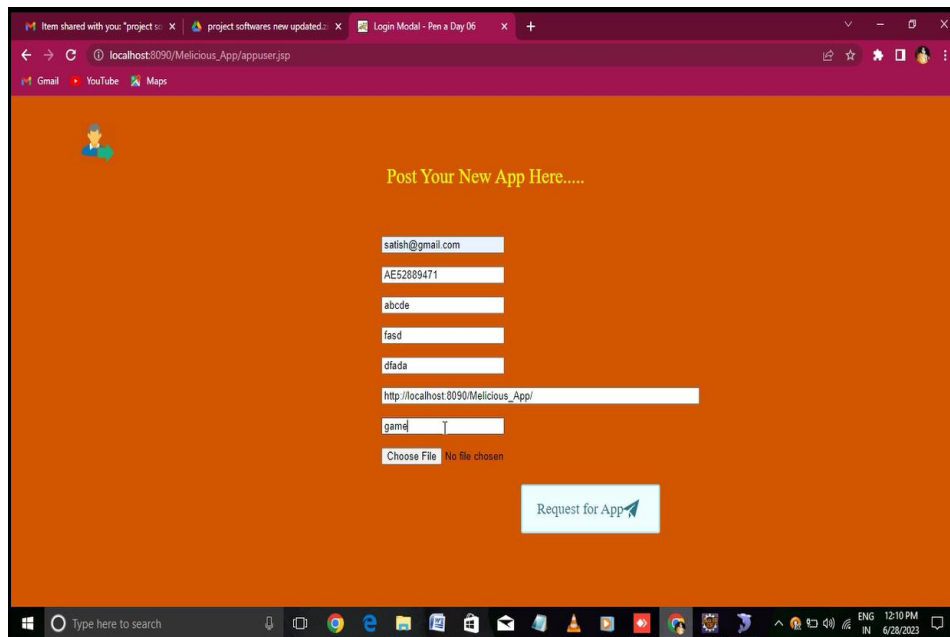
FIG 23: ADMIN LOGIN BY USING HIS CREDENTIALS TO GET THE APP ID



rohit	rohitk641722@gmail.com	123	8892233837	ckm	AE31544734	Activated
irfan	irafan	ir@gmail.com	9663318058	tunkur	AE5007810	Activated
kirthi	k@gmail.com	123	8431937850	tumkur	AE91616226	Activated
prajwala	pra@gmail.com	123	9901075525	tumkur	AE79513060	Activated
manju	manjunathms72@gmail.com	123	9731929530	tumkur	AE71695333	Activated
abcd	abcd@gmail.com	123	9731929530	tumkur	AE65369552	Activated
sueshr	suresh@gmail.com	123	9988998899	tumkur	AE28208045	Activated
xyz	xyz@gmail.com	123	8877667788	tumkur	AE91131170	Activated
h1	h1@gmail.com	123	5555555555	dfsf	AE20437876	Activated
a1	a1@gmail.com	123	6543255	fgvfg	waiting	waiting
tharun	tharun@gmail.com	123	7766556677	tumkur	AE99219512	Activated
ravi	ravi@gmail.com	123	8855441122	tumkur	waiting	waiting
aaaa	aaa@gmail.com	123	8877441122	tumkur	AE58262794	Activated
viru	viru@gmail.com	123	7744112255	tumkur	waiting	waiting
abcd	abcd@gmail.com	123	7744112255	tumkur	AE65369552	Activated
satish	satish@gmail.com	123	7744112255	tumkur	AE52889471	Activated

FIG 24: ADMIN FETCHES THE APPIDS IN REQUEST APP

The admin the having the access to fetch the AppID’s of requested Apps , that is shown in fig 24. And in fig 25, we can see that the developer including his APPID to share the app.



Post Your New App Here.....

No file chosen

FIG 25: DEVELOPER INCLUDING HIS APPID TO SHARE THE APP

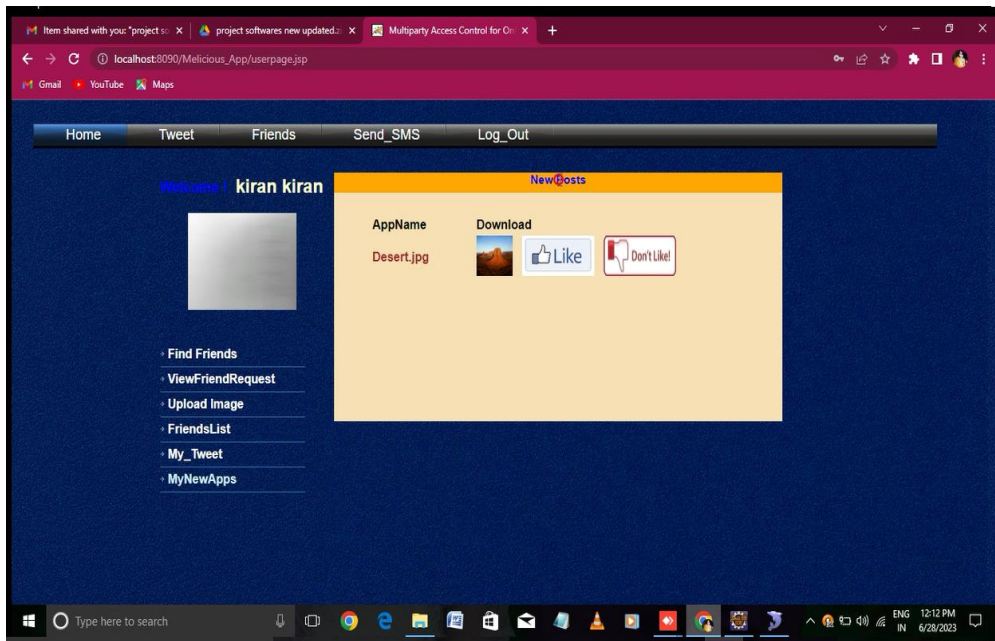


FIG 26: THE USER GETTING NEW POST POP UP MESSAGE OF NEW APP

In fig 27, the user is getting a new post pop up message of a new app. In this, the user can like or dislike the post and can also download the post. In fig 27, it shows that when the user likes the post.

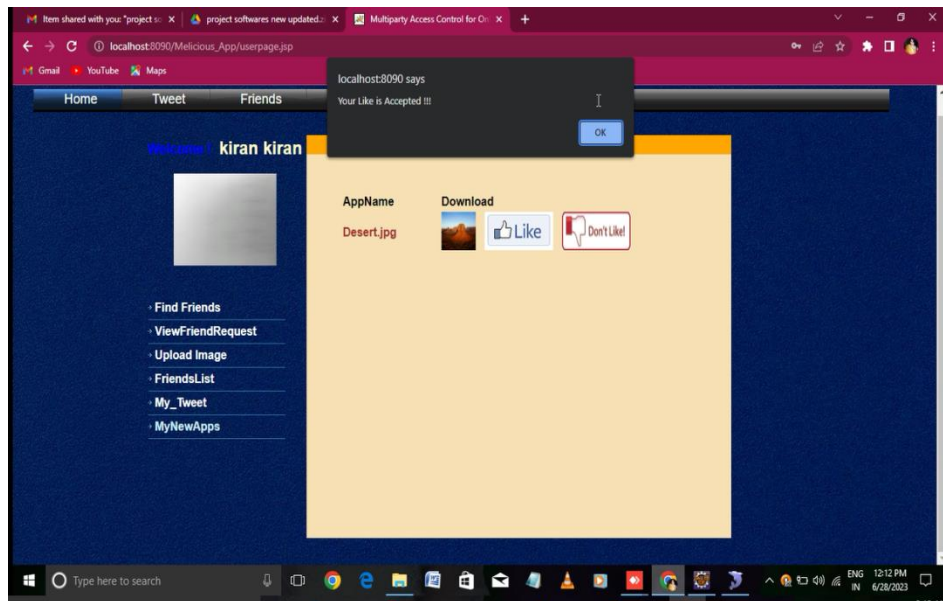


FIG 27: POP UP MESSAGE WHEN USER LIKES THE POST POSTED BY DEVELOPER

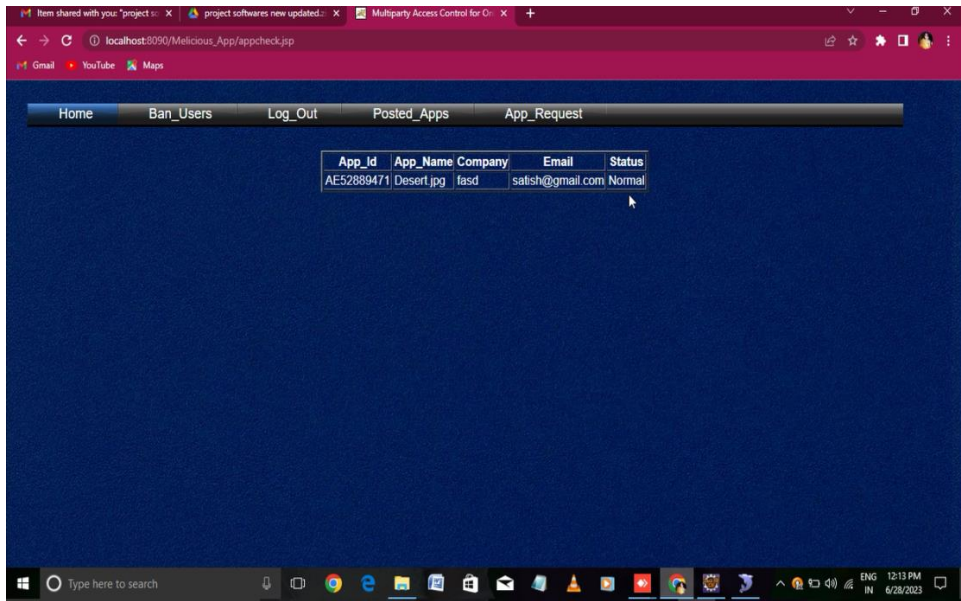


FIG 28: THE ADMIN FINDS THAT POST IS NORMAL

The admin can check whether the posted app is malicious, suspicious or normal one. In fig 28 the admin finds that the post is normal. And in fig 29, the hacker posts the app with an unregistered app ID.

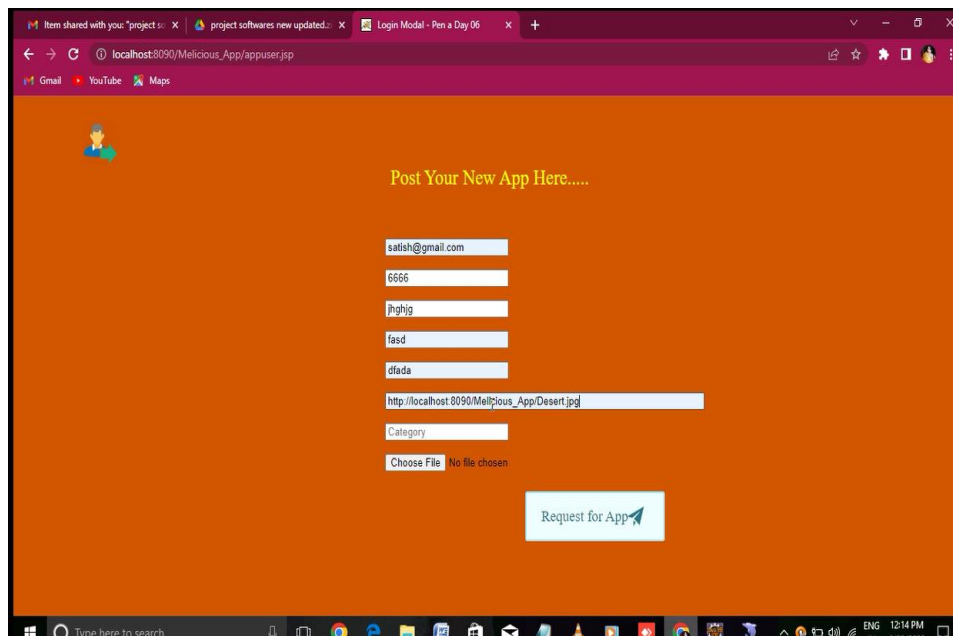


FIG 29: POSTING A POST WITH UNREGISTER APPID

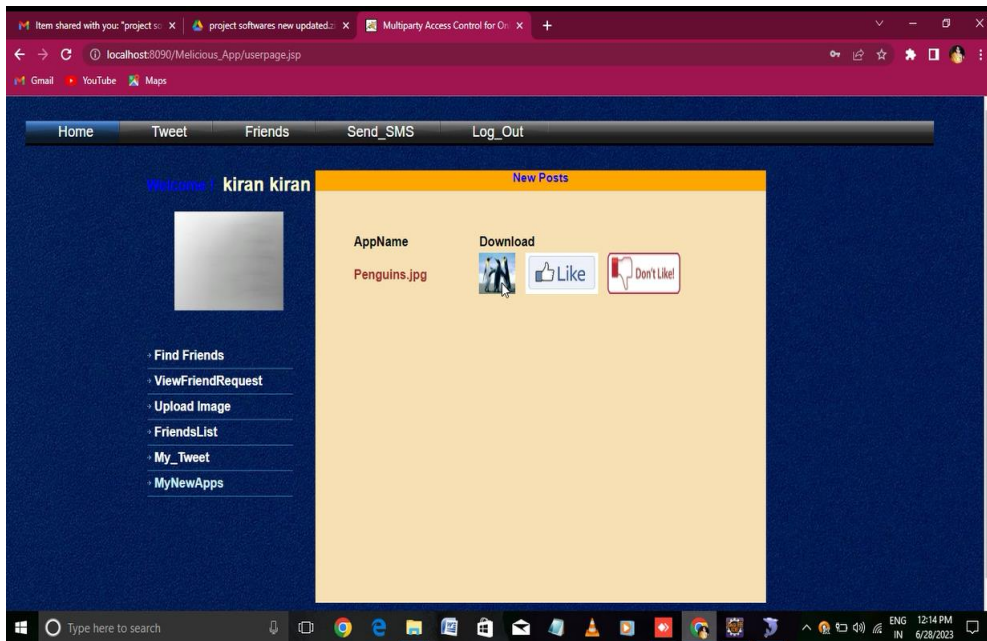


FIG 30: GETTING POST NOTIFICATION ON USER PROFILE

In fig 30, getting post notification on user profile and getting warn pop up message when user click on download in fig 31.

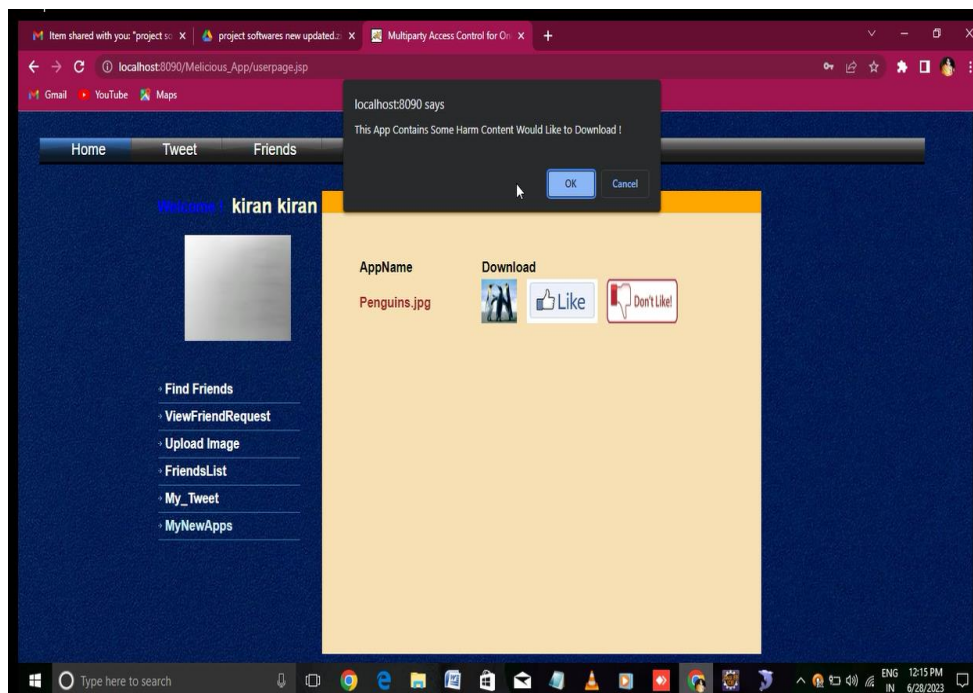


FIG 31: GETTING WARN MESSAGE WHEN USER CLICK ON DOWNLOAD

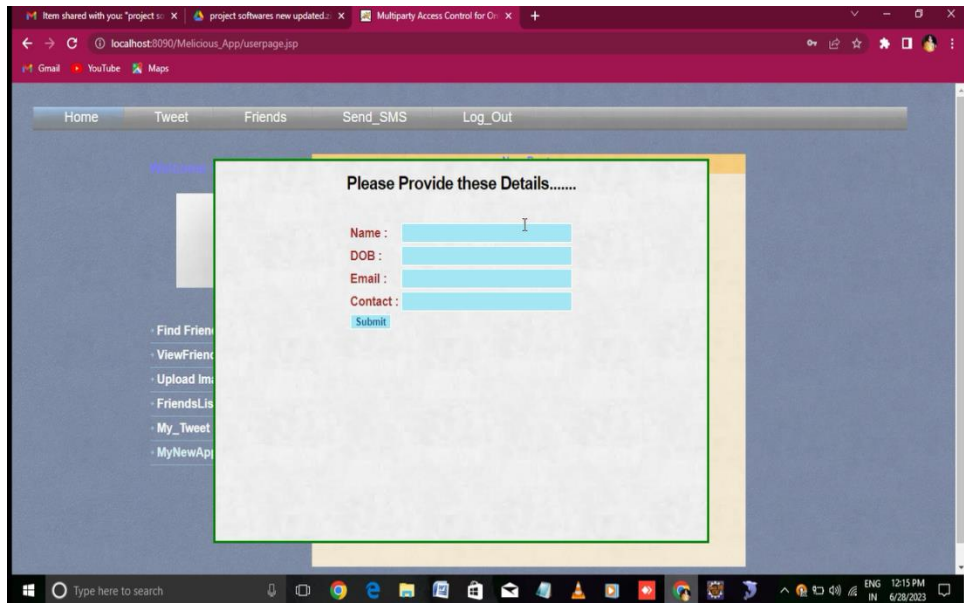


FIG 32: ASK FOR FUTHER DETAILS

In fig 32, after downloading suspicious app the page takes to another page to give futher details. The admin found thst post is suspicious as shown in fig 33.

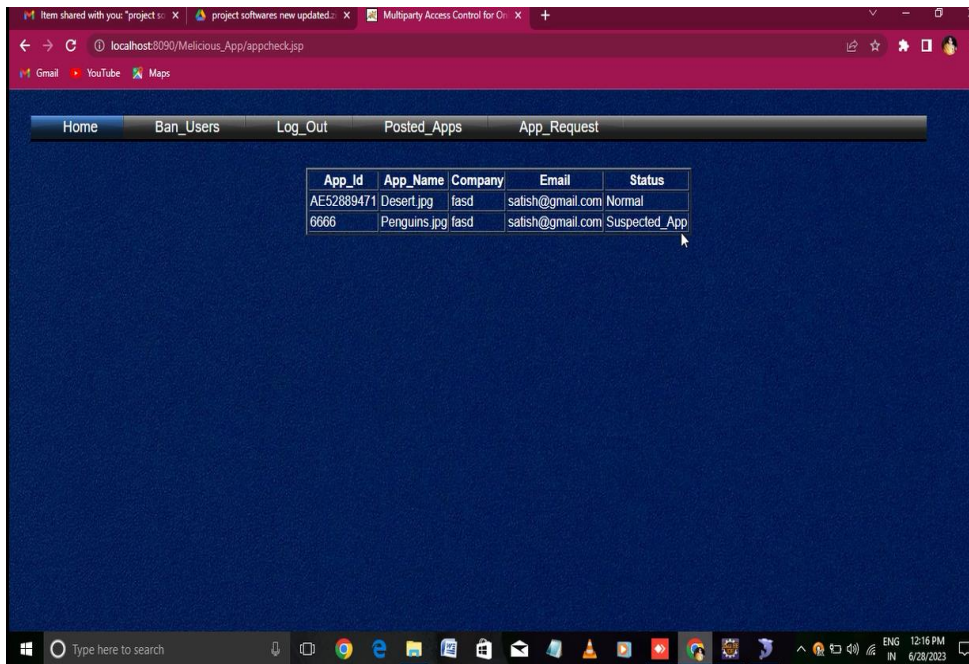


FIG 33: THE ADMIN FINDS THAT POST IS SUSPICIOUS APP.

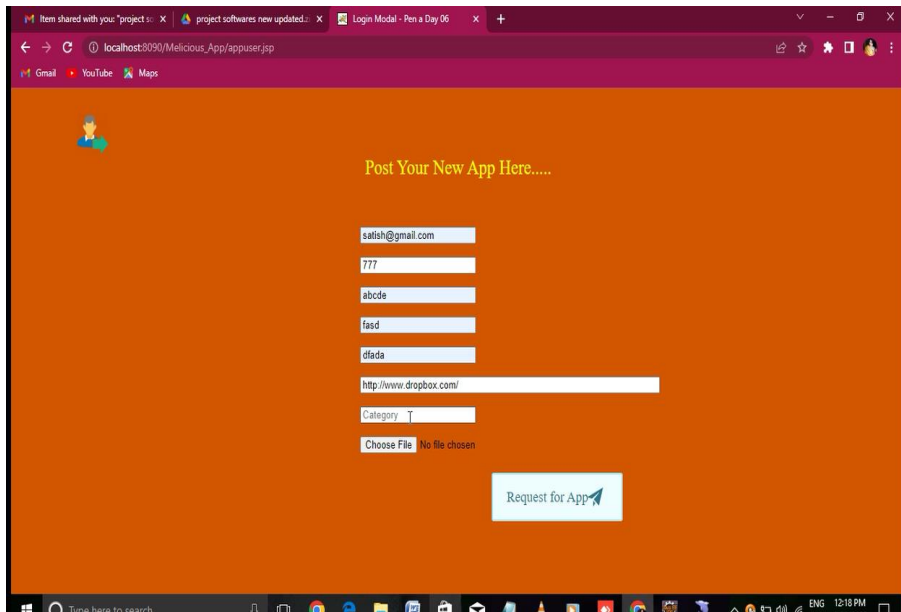


FIG 34: POSTING A POST WITH UNREGISTER APPID

The hacker post an app with unregistered AppID as shown in fig 34 and we can also see that hacker performing fishing in fig 35.

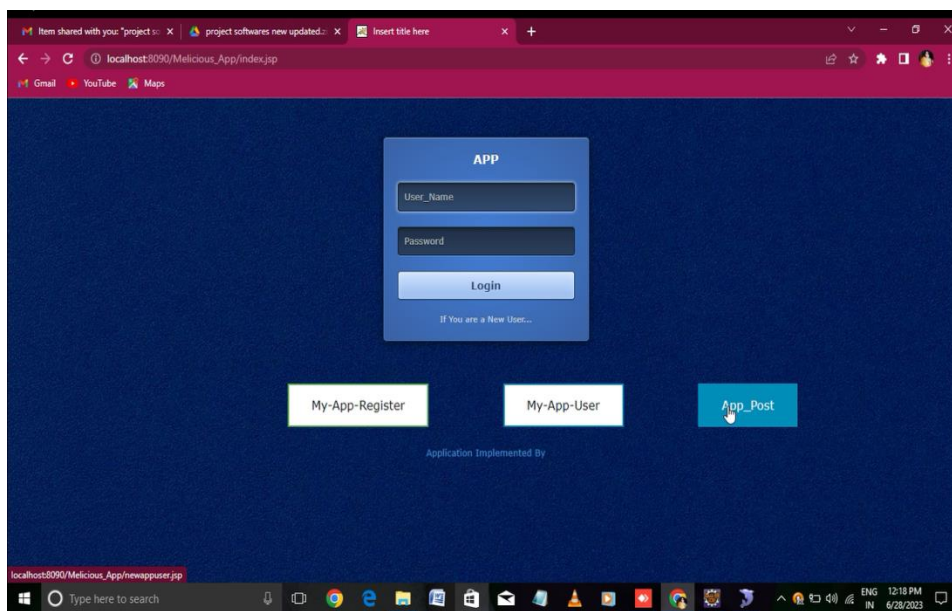


FIG 35: HACKER PERFORMING PHISHING BY USING APP POST

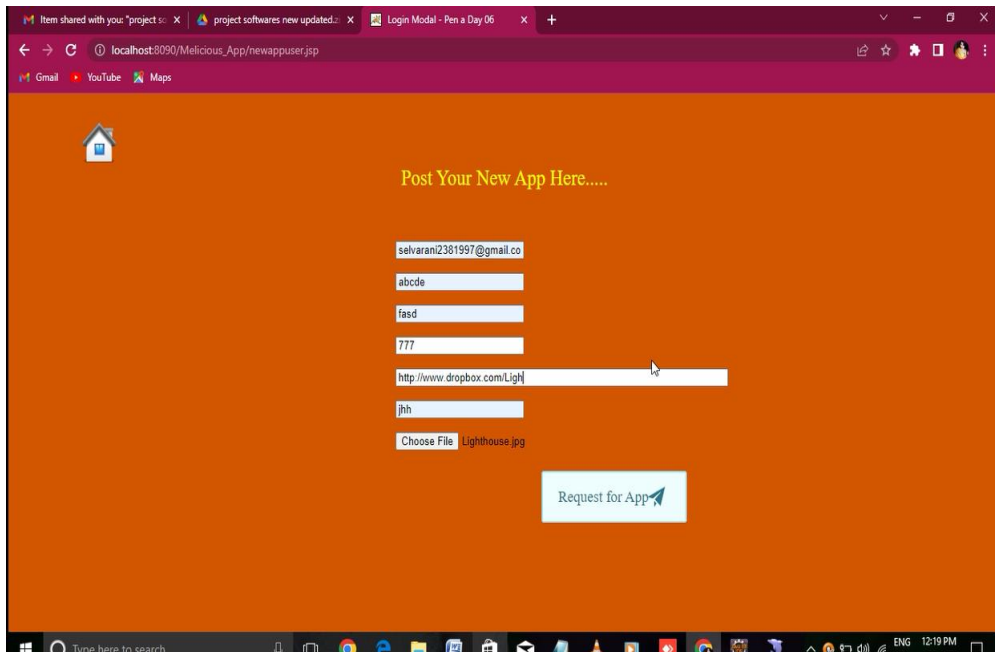


FIG 36: HACKER PERFORMING PHISHING BY GIVING UNREGISTERED APPID OF PREVIOUS POST.

In fig 36, hacker performing phishing by giving unregistered appid of previous post and in fig 37, gettign an error message when user attempt to click on first app request

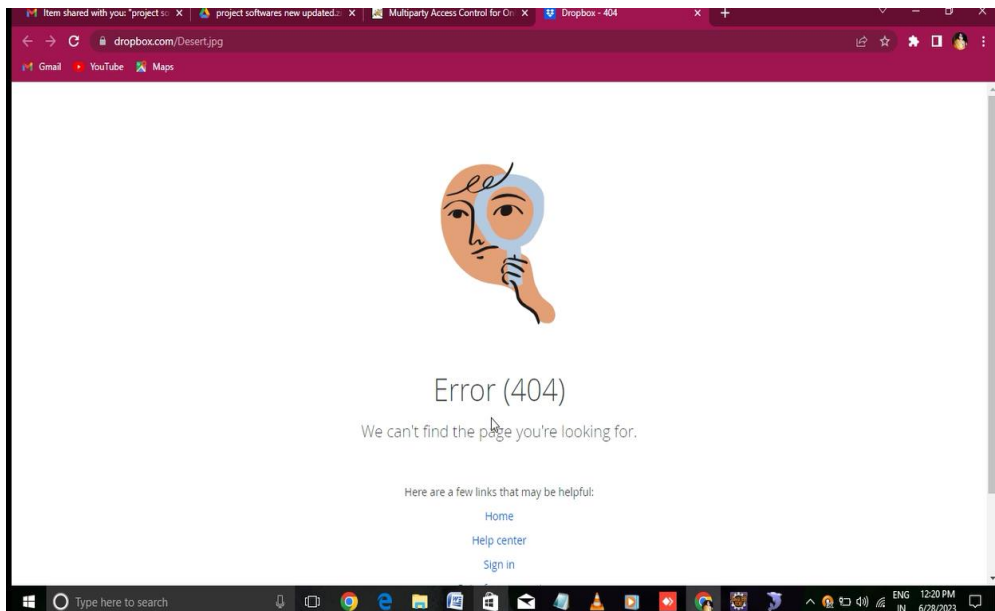


FIG 37: GETTING AN ERROR MESSAGE WHEN USER ATTEMPT TO CLICK ON FIRST APP REQUEST

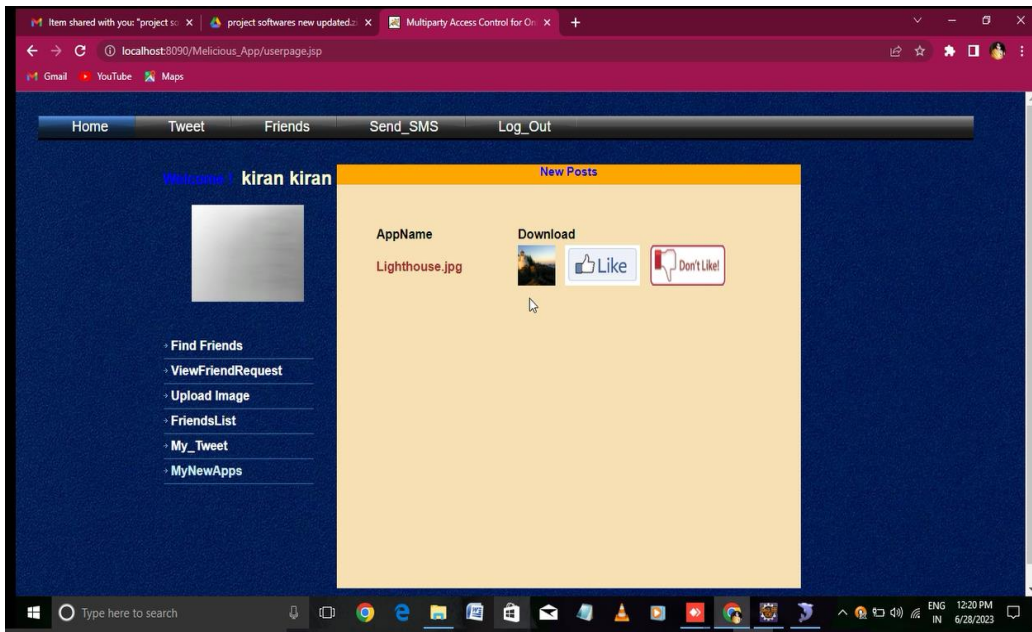


FIG 38: GETTING ONE MORE POP UP POST AFTER GETTING ERROR PAGE

Getting one more pop up post after getting error page in fig 38 and admin found that apps are suspicious and malicious as shown in the fig 39.

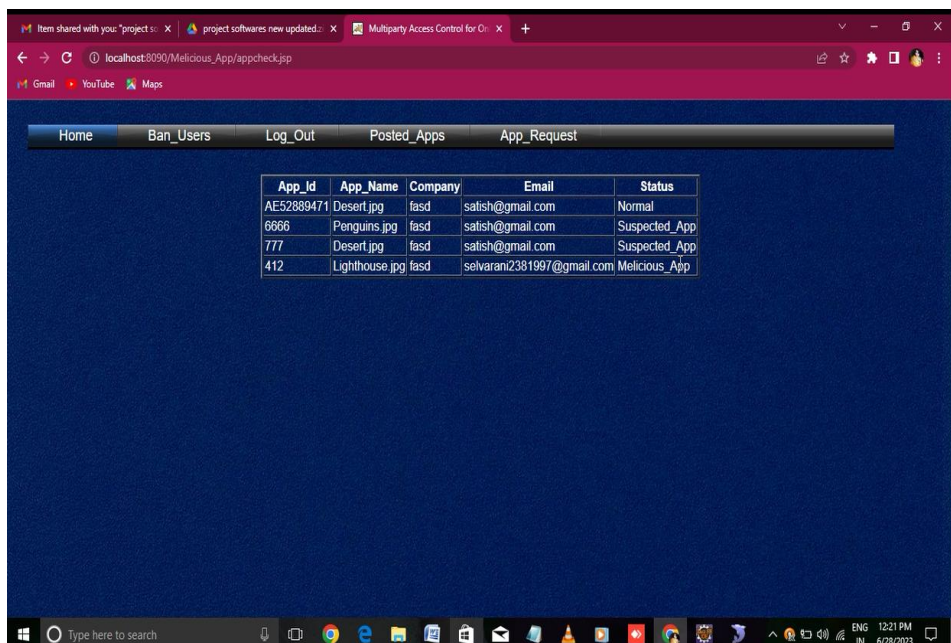


FIG 39: ADMIN FOUND THAT APPS ARE SUSPICIOUS AND MALICIOUS

FEATURE ENHANCEMENT

The existing system recognizes bad actors and blocks their access to a certain website (such as Twitter, Facebook, etc.). Because of how the suggested fix functions, the malicious user cannot access any websites now and cannot access any websites in the future.

CONCLUSION

Unit testing, a method to confirm that a particular module of the source code is operating properly, is one of the testing approaches covered in this chapter. Another name for it is module testing. Also briefly mentioned are the many forms of integration testing, which combine and test various software parts individually. In addition to these two main types of testing, the preparation of test data, validation testing, output testing, user acceptability testing, and many more types of testing are also discussed in this article. We released a state-of-the-art method for precisely locating malicious social bots in online social networks. According to studies, it is possible to precisely identify dangerous social bots on online social networks by using social scenario analytics based on the possibility of a transition occurring between user clickstreams. Future research will examine more risky social bot activities, and the suggested detection method will be improved and expanded to reveal the precise intentions and objectives of a larger range of hostile social bots.

REFERENCES

1. F. Morstatter, L. Wu, T. H. Nazer, K. M. Carley, and H. Liu, "A new approach to bot detection: Striking the balance between precision and recall," in Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining, San Francisco, CA, USA, Aug. 2016, pp. 533_540.
2. C. A. De Lima Salge and N. Berente, "Is that social bot behaving unethically?" Commun. ACM, vol. 60, no. 9, pp. 29_31, Sep. 2017.
3. M. Sahlabadi, R. C. Muniyandi, and Z. Shukur, "Detecting abnormal behavior in social network Websites by using a process mining technique," J. Comput. Sci., vol. 10, no. 3, pp. 393_402, 2014.
4. F. Brito, I. Petiz, P. Salvador, A. Nogueira, and E. Rocha, "Detecting social network bots based on multiscale behavioral analysis," in Proc. 7th Int. Conf. Emerg. Secur. Inf., Syst. Technol. (SECURWARE), Barcelona, Spain, 2013, pp. 81_85.
5. T.-K. Huang, M. S. Rahman, H. V. Madhyastha, M. Faloutsos, and B. Ribeiro, "An analysis of software cascades in online social networks," in Proc. 22nd Int. Conf. World Wide Web, Rio de Janeiro, Brazil, 2013, pp. 619_630.
6. H. Gao et al., "Spam ain't as diverse as it seems: Throttling OSN spam with templates underneath," in Proc. 30th ACSAC, New Orleans, LA, USA, 2014, pp. 76_85.