International Journal of Innovative Research in Engineering
Volume 3, Issue 3 (May-June 2022), PP: 508-512.
www.theijire.com
ISSN No: 2582-8746

Employee Monitoring Software

Pranav Paigude[1], Sahil Gupta[2], Prof. Rupali Gavaraskar[3], Prof. Dnyaneshwar Kanade[4]
[1,2,3,4]Department of Electronics Engineering, Vishwakarma Institute of Technology Pune, Maharashtra, India.

How to cite this paper:
Pranav Paigude[1], Sahil Gupta[2], Prof. Rupali Gavaraskar[3], Prof. Dnyaneshwar Kanade[4] "Employee Monitoring Software", IJIRE-V3I03-508-512.

Copyright © 2022 by author(s) and5th Dimension Research Publication.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/

Abstract: *In recent years information technology (IT) has transformed workplaces tremendously. Nowadays organizations all over the world are relying on technology to monitor the activities performed by their employees during work hours and to increase the productive tasks done by them. This was because, a lot of studies concluded that if employees are not supervised, they have the tendency to waste time online. Monitoring employee helps organizations to keep a track on activities performed by them. This allows them to improve employee productivity. To effectively observe and monitor the employees during working hours there should be a proper check, so there is need of an intelligent, automated and efficient software. Hence the idea of developing a program that will monitor the screens of the employees of an organization was conceptualized. The text from the screenshot will be extracted and scanned for certain keywords. If the work related keywords are present in the text, the screenshot will be discarded. Else the screenshot will be blurred and saved on the employee's profile on a database, which would be accessible to the manager or the employer. The employer can see the count of such images on each employee's profile and take any action if necessary. Other information such as login time of employee, last seen time, total hours of work of a particular day would also be displayed in the database. This project will use python as the programming language and also use Tesseract-OCR for detecting the images and the text present in them.*

Key Word: *Employee monitoring; Python; Screenshot*

I.INTRODUCTION

As remote mode of work became normal during the pandemic, there was rising need for employers to monitor their employees who were not in their direct sight. Now, with remote work strategies still in place, and office re-openings being pushed back, the use of employee monitoring tools continues to grow. There has always been an element of workplace monitoring that involved managers being able to look over their office and their workers. The use of digital monitoring technologies if done with proper care can be beneficial for employers as well as employees. There can be several reasons why employers might want to track employee activities, like protecting sensitive data, preventing the employees from indulging in prohibited activities and increase work efficiency. Few of the already existing methods of remote monitoring include monitoring of emails, collaboration tools, web browsing, as well as the use of video tracking, attention tracking via webcam and key-logging. The above techniques do not safeguard the privacy of the employees and hence are opposed by them. This paper presents a better monitoring approach that will safeguard user privacy. The screenshots taken in this approach will be blurred after extracting the text from them. So the screenshots visible to the employer will be blurred but will give an idea about the window being open. The employer can thus understand if the window open was related to professional software or any unrelated software.

II. LITERATURE SURVEY.

Some previously developed applications [8], provide employee attendance marking services, along with the daily calculation of total work hours using a mobile application. However it does not provide for any type of employee monitoring application. One method of monitoring that has become popular is keystroke monitoring. Keystroke monitors can be used to generate and analyze two different kinds of information: the keystrokes data and the keystroke dynamics. The keystrokes data can be used to reconstruct the actual information typed by the user such as the commands or programs run, or the messages typed etc. Free-text detection of keystroke dynamics is a challenging and a not so common area, which is why very few works have been published so far. A new technique for free-text detection of keystroke that can be used for passive and dynamic user monitoring has been presented [9]. By allowing dynamic and passive user monitoring, the proposed technology can be used to track continuously legitimate and illegitimate users throughout computing sessions. By collecting only user keystroke dynamics instead of actual keystrokes data, this technique limits the amount of personal information gathered. Compared to other monitoring methods, this method somewhat protects the privacy of the person being monitored. While employee surveillance has become necessary, most of the technologies used till now do not take into consideration the privacy of the employees. Existing surveillance technologies record sensitive information such as telecommunications records, logs of web usages, email messages, etc. Such sensitive information is stored in organization servers exposing them to possible theft by hackers and identity fraudsters. Several critics and privacy advocates feel that the employee monitoring tools used so far infringe on employee privacy, which is a fundamental personal right. There is a complex situation, where it is necessary to protect the organization's data from insider threats, while it is equally important to safeguard employee privacy rights.

The technique for employee monitoring proposed in our paper helps monitor employees without violating their privacy. Our technique neither monitors keystrokes, nor does it collect email data. It just takes screenshots after random time intervals and applies a blur effect on the screenshot before uploading it on the cloud. The moderate blur effect makes it impossible to read the text in the screenshot, but gives an idea of which window was open and which software was being used. Since the text from the image is blurred, text messages, chats, passwords, etc. cannot get exposed. This safeguards employee privacy.

III. METHODOLOGY

The aim of the project is to propose a technology for a work center which will track and report the activities which their employees are doing at all times by taking screenshots of their screens and reading the text in it.

A. Text Extraction using OCR

OCR, or Optical Character Recognition, is a process of recognizing text inside images and converting it into an electronic form. These images could be of handwritten text, printed text like documents, receipts, name cards, etc., or even a natural scene photograph.

Pytesseract or Python-tesseract is an Optical Character Recognition (OCR) tool for python. It will read and recognize the text in images. We used it to extract text from the captured screenshots and further match the keywords in order to report any activity to admin.

i. Image Acquisition - In this module, the model will capture the images of employees screen and image digitization takes place.

ii. Pre-processing - It is basically the operations that are applied to a digital image. The output of the pre-processing stage will be a normalized bitmap image.

iii. Segmentation - The process of dividing pre-processing images into meaningful sub-images, units, such as words, sentences, or topics. The techniques applied for image segmentation is the thresholding method, edge detection-based techniques etc.

iv. Feature Extraction - It is the process of extracting the important characteristics of an image. In the feature extraction stage, each character is represented as a feature vector, which becomes its identity.

v. Post-processing - The module will provide the recognized character in a structured text form and it involved data cleaning steps for documents.

B. Firebase Cloud Storage

Cloud Storage is built for application developers who need to store and serve user-generated content, usually big files like photos or videos. It is mostly used or developed for photos and videos, but we might use this for other things like text files. It is a cloud computing model which stores data on the internet through a cloud computing provider who manages and operates data storage as a service. It is delivered on-demand with just-in-time costs and capacity. It eliminates buying and managing our own data storage infrastructure. It provides us agility, durability, global scale with "anywhere, anytime" data access.

In our case we are using two build functions of firebase:

1. **Firebase Storage:-** Cloud Storage is just for storing binary data using paths that look like file system paths. It is not exactly a database, and you cannot really query it like other databases. It is typically used for storing files like pictures, videos, PDFs, backups/exports, and other raw data which can be very large in size.

In our model we are using cloud storage to store the screenshots of employee’s screen in which it is found that the employee was using any other website or tool. According to our model we are matching the words present in the captured screenshots with given input keywords and on that basis the program determines whether the screenshot is to be saved or discarded.

2. **Firebase Real time Database**:-Fire store is a database used for querying data and is almost never used for storing binary data. We use it to store actual values that you intend to query, such as names, times, and other metadata. Since a Fire store document is limited to 1MB in size that also drastically cripples its ability to hold very large amounts of data like Cloud Storage. In this project we have used it to store employee’s details like login time, total working hours, current applications being used, etc. so that the data is synchronized in real-time and can be viewed by the manager or employer.

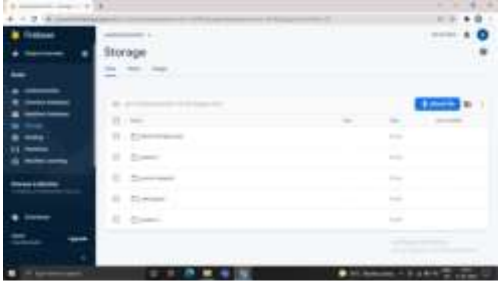
In this project, we have created an application which will start automatically on starting the laptop in which the application is installed. The application will establish a connection with the firebase database using its credentials like API key, database URL, firebase project id, domain etc. The application will capture the PC details and upload the start time of the application as the login time of the employee in the employee profile of that particular day in the database. The application creates a list of current running programs on the PC and the work related programs (Coding software, MS Excel, etc.) are saved in the 'Current Application' field of the database. This enables the manager or employer to view the applications open on the employee's laptop. Then the program takes a screenshot of the screen and uses Tesseract OCR for extracting the text from the screenshot. A text file is generated which stores the text in it. The program has a list of keywords which are related to the employee's work or domain in which he works. Usually these keywords have to be present on the screen when the employee is working. The text file generated is scanned for these keywords and if a keyword is present it means the employee is performing some work related tasks. Hence the screenshot is discarded. But if keywords are not present, then the screenshot will be uploaded on the Firebase Storage folder of the employee. The employer can view this screenshot and take any action if he finds the employee not doing any work related task. The program then uploads the current time in the 'Last Seen' field in the database of that employee. This program will be repeated after random intervals of time and the screenshots will be taken repeatedly. Also the last seen of the employee will keep getting updated. The total time for which the employee is working will also be calculated and updated in the database.

The employers or managers will daily, get an overview of when his employees started working and how long each one worked. The number of screenshots in an employee’s database will give an idea about how much time he spent on other tools or websites as compared to the other employees. The managers can use this data to compare employees, understand employee behaviour and take further decisions like assigning certain projects to certain employees, salary hikes, promotions, etc.

IV. RESULTS

Firstly, as the application starts, a unique profile of each employee gets created in the database as seen in Fig 1.

Fig1:Employee profiles



We have stored those screen shots of employees’ screen which are not relevant to the task given to employee in firebase storage. This can be seen in Fig 2.

Fig2: Screenshots of tasks

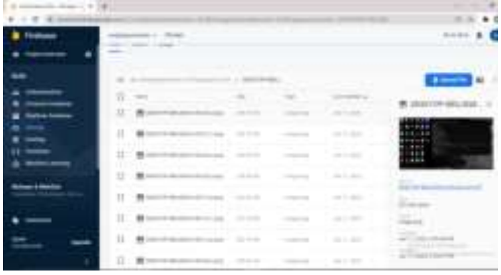


Fig 3 shows the output of the application on the terminal window. This window will not be displayed on the employees’ screens but has been shown for demonstration only.

Fig3: Output on terminal

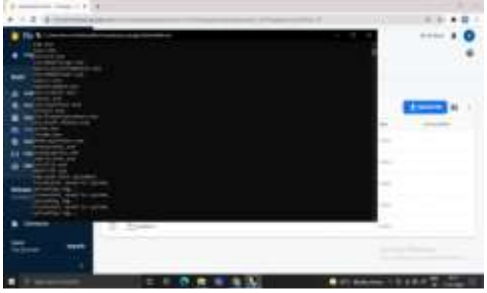


Fig 4 shows the various fields in an employee’s data. These are the parameters we used to monitor an employee and stored their data under their own profile. Also we have kept record of data for each day as seen in Fig 5. Fig 6 shows a screenshot saved on the cloud in which it was observed that the employee was using social media during work hours.

Fig4: Employee Data

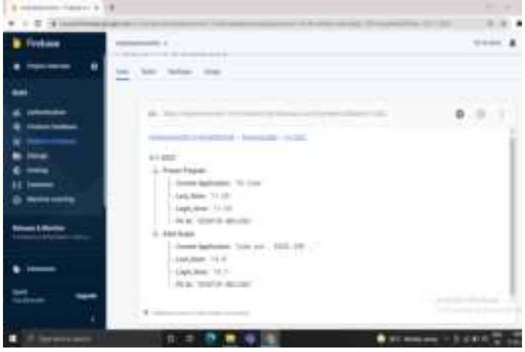


Fig5: Data of different dates stored under different tables

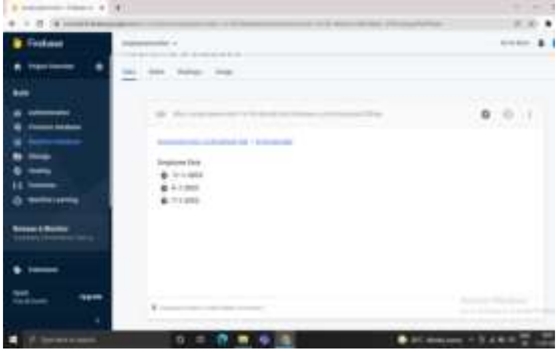
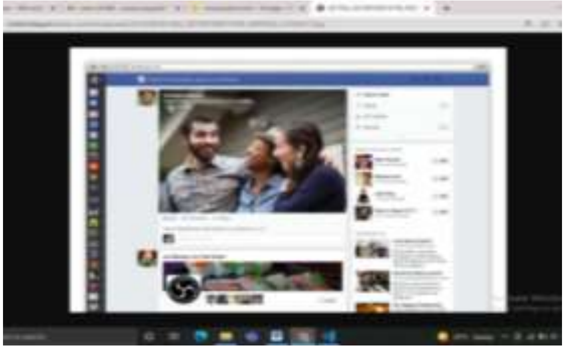


Fig6: Blurred screenshot saved on cloud storage



The program runs fluently. The employer can get an overview of various details of an employee in real-time and can also view them later. The interface of the database is not complex and is easy to understand for employers from various domains. On opening the database tab, data from various days is available. The manager has to expand the table of any desired day to get the detailed data. Similarly in the firebase storage tab, the required folder can be opened to view the screenshots. Depending on the number of screenshots under a particular employee's field, on a particular day the manager can interpret employee behavior and take any action if required. If the screenshot shows the employee using a tool which is not in the interest of the organization, managers can question the employee regarding such actions.

V. CONCLUSION

This paper discussed about the Model of Employee Monitoring System. The primary objective of this paper is to address the issue of automatic and inhuman employee monitoring in the computer-mediated work environment. The application requires a stable internet connection to connect to the cloud storage. The application is also a lightweight and less complex one, requiring lower memory. Hence it can easily run in the background while other tasks are being performed on the PC and this makes it an efficient employee activity monitoring application.

VI.FUTURE SCOPE

We can further add a feature of live screen viewing where employers can view the live screen of any employee.
We can use AI and ML and use the employee data to calculate the work efficiency of the employees. This can be further expanded to manage employee stress and remind the employee of taking a break after regular intervals. It can aid in reducing employee's mental fatigue and can be beneficial for the company as well as the employee.

References

[1]. *Yerby, J. (2013) Legal and ethical issues of employee monitoring. In Online Journal of Applied Knowledge Management A Publication of the International Institute for Applied Knowledge Management (Vol. 1).*

[2]. *2. Chang, S. E., Liu, A. Y., & Lin, S. (2015). Exploring privacy and trust for employee monitoring. Industrial Management and Data Systems, 115(1), 88–106.*

[3]. *3. Papini, J.S. (2007). Big Brother: The Effect of Electronic Employee Monitoring on Electronic Misbehavior, Job Satisfaction and Organiozational Commitment. Ann Arbor, MI: ProQuest Information and Learning Company*

[4]. *4. Stanton, J.M. & Stam, K.R. (2006) The Visible Employee: Using Workplace Monitoring and Surveillance to Protect Information Assets-Without Compromising Employee Privacy or Trust. New Jersey: Information Today, Inc.*

[5]. *5. Weckert, J. (2005). Electronic Monitoring in the Workplace: Controversies and Solutions. Heyshey, PA: Idea Group Publishing*

[6]. *6. Employee Monitoring and Tracking System. Sumit Kuldharan, Shubham Bhabad, Saurabh Shirsath, Supriya Borhade*

[7]. *7. Ferdianto, L. S. Sanjaya and Titan, "Design of Employee Management Application for Small Medium Enterprise," 2021 International Conference on Information Management and Technology (ICIMTech), 2021, pp. 696-701, doi: 10.1109/ICIMTech53080.2021.9534990.*

[8]. *8. Rishabh Bajpayi, Prof. M L Sharma and K C Tripathi, "Employee Management System", International Journal for Modern Trends in Science and Technology, 6(12): 225-234, 2020.*

[9]. *9. Ahmed, Ahmed Awad E.. "Employee Surveillance Based on Free Text Detection of Keystroke Dynamics." (2009).*