

Harrisburg University of Science and Technology

## Digital Commons at Harrisburg University

---

Other Works

---

4-7-2021

### Second Version on A Centralized Approach to Reducing Burnouts in the IT industry Using Work Pattern Monitoring Using Artificial Intelligence using MongoDB Atlas and Python

Sasibhushan Rao Chanthati

*Harrisburg University of Science and Technology, sasichanthati@gmail.com*

Follow this and additional works at: <https://digitalcommons.harrisburgu.edu/other-works>



Part of the [Business Intelligence Commons](#), [Computer and Systems Architecture Commons](#), [Data Storage Systems Commons](#), [Digital Communications and Networking Commons](#), [Leadership Commons](#), [Other Computer Engineering Commons](#), and the [Robotics Commons](#)

---

#### Recommended Citation

Chanthati, Sasibhushan Rao. (2021). Second Version on A Centralized Approach to Reducing Burnouts in the IT industry Using Work Pattern Monitoring Using Artificial Intelligence using MongoDB Atlas and Python. 10.13140/RG.2.2.12232.74249.

This Article is brought to you for free and open access by Digital Commons at Harrisburg University. It has been accepted for inclusion in Other Works by an authorized administrator of Digital Commons at Harrisburg University. For more information, please contact [library@harrisburgu.edu](mailto:library@harrisburgu.edu).

## **Second Version on A Centralized Approach to Reducing Burnouts in the IT industry Using Work Pattern Monitoring Using Artificial Intelligence using MongoDB Atlas and Python**

**Author: Sasibhushan Rao Chanthati**

**Research Paper and System Interface**

Email(s):

[sasichanthati@gmail.com](mailto:sasichanthati@gmail.com)

[sasibhushan.chanthati@gmail.com](mailto:sasibhushan.chanthati@gmail.com)

[SChanthati@alumni.harrisburgu.edu](mailto:SChanthati@alumni.harrisburgu.edu)

[sasibhushanchanthati@ieee.org](mailto:sasibhushanchanthati@ieee.org)

Professional Profile Link(s):

<https://www.linkedin.com/in/sasibhushanchanthati/>

<https://orcid.org/0000-0001-5778-8140>

<https://www.researchgate.net/profile/Sasibhushan-Rao-Chanthati>

<https://scholar.google.com/citations?user=t6JwIkoAAAAJ&hl=en>

[DOI: 10.13140/RG.2.2.12232.74249](https://doi.org/10.13140/RG.2.2.12232.74249)

<https://rgdoi.net/10.13140/RG.2.2.12232.74249>

Second Version on A Centralized Approach to Reducing Burnouts in the Industry Using Work Pattern Monitoring Using Artificial Intelligence © 2021 by Sasibhushan Rao Chanthati is licensed under CC BY-NC

US Copyrights Registration Number: TXu 2-247-181(April 07, 2021)

The document is Non-Affiliated, Individual research paper by Author: Sasibhushan Rao Chanthati, no organization is associated with this research paper.

## Abstract

Industry burnout is interlinked with cultural, individual, physical, or emotional exhaustion, and social factors, the resolution of which requires the technology-driven trends in the workplace and the technologies such as work pattern monitoring and Artificial Intelligence that can deal with large amounts of data. Industries face a gigantic problem i.e., employee burnout which can charge a firm loss in numerous hours and thousands of dollars every year. The more advanced companies use work pattern monitoring using Artificial Intelligence to make their employees work more professionally. In this research my attempts to understand the development and leadership, on the effects of work pattern monitoring using Artificial Intelligence technology on information technology organizations (Sasibhushan Rao Chanthati, 2022).

In this updated second version, the data of the employees will be stored on **a cloud server with governance & compliances**. The study discussed the development of methods which are configured as **two different system interfaces**, which are of minimum valuable product (MVP) and the results obtained from the two approaches. The system will provide work pattern monitoring via the **‘Real-Time Database – MongoDB Atlas’** which will synchronize the employee burnout data to improve the employee experience. This research also illustrates the advantages and disadvantages of the proposed solutions. “Burnout Detection Mechanism” that will help Industry management and Human Resource Management to manage the emotional state of the employees, understanding their real state. The study conducted a self-survey, and the outputs of the surveys are explained in this paper. The sample data we are using is mainly focused on information technology employment perception.

**Keywords:** Burnouts, Work Pattern Monitoring, Python, Large Language, Vector Search, Artificial Intelligence, MongoDB Atlas.

## 1. Introduction

The **IT (Information Technology) industry** is usually characterized by constant technological innovation and highly competitive demands, which usually contribute to the pressure experienced by people working in the industry. Stress, a condition whereby an employee experiences physical, emotional, and mental drain, due to stress, has been prevalent in this sector (Ajayi FA, Udeh CA, 2024). A range of demands that challenge IT practitioners include time constraints and overtime working in addition to the pressure to learn new technologies, which boosts stress and burnout (Gandhi TK, Classen D, Sinsky CA, Rhew DC, Vande Garde N, Roberts A, Federico F, 2023). Combating burnout is important because it has a direct effect on the health and satisfaction of the employees, as well as influencing organizational outcomes. This results in low job satisfaction, high rates of absenteeism, high turnover rates and consequently, damages business organizations (Leitao J, Pereira D, Gonçalves A, 2021). The research questions formulated for this study relate to the shortcomings of the existing approaches for assessing and mitigating burnout in the IT sector.

The one-time mandatory surveys and the constant monitoring of the employees by the HR departments do not help identify the employees' current emotional or mental status (Rožman M, Oreški D, Tominc P, 2023). These are also known to be very passive methods because they deal with burnout when it has already affected the employees in a big way. Unfortunately, due to the absence of proper tracking of the listed indicators, burnout remains invisible until it results in adverse outcomes for workers and the overall efficiency of the company (Narayanan LK, Subbiah P, Muralidharan RR, Baskaran AP, Srinivasan V, Baskaran AP, Victor P, Ramachandran H, 2023). This study aims at creating a centralized solution that monitors work patterns of IT professionals in real-time and identify early signs of burnout

The study seeks to design a model that uses AI and machine learning to process the gathered data on employees' work patterns and other related information to predict burnout signs and suggest the necessary actions to take. When applied to the monitoring process, AI can provide

a proactive approach and help identify the necessary actions to promote the health of staff members more consistently and effectively (Gordon S, Adler H, 2022).

**The scope of this research includes the development and implementation of two system interfaces: Vector embedding.py to handle all the information and reviews about the employees and Vector search.py for searching and analyzing the data of the employees.** This research will target IT professionals due to the high levels of burnout commonly exhibited in the respective field. However, the findings of this research and the developed system can be applied to other sectors that are experiencing similar challenges. The limitations of this research can include the necessity to use a substantial amount of data for the training of AI models; and its challenges in implementing new forms of monitoring among employees or organizations.

This study is important because it offers the theoretical and practical framework for enhancing the well-being of employees in organizations. The approach described in this paper allows organizations to scale and monitor burnout in real-time, which will save employees' health and increase their satisfaction and retention. The use of AI in monitoring work patterns is one of the major developments in the application of human resource management that provides a data-driven perspective of the burnout issue. This work does not only seek to add to the accumulation of knowledge in the workplace but also seeks to be of practical utility in enhancing the organizational well-being at the workplace, thereby improving organizational productivity.

## **2. Literature Review**

### ***1.1 The IT industry and professional burnout***

Despite being a long-standing issue, burnout has become an increasingly significant problem in the IT industry, with various researchers conducting numerous studies to explore its frequency and impact on the workforce and organizations (De Stefano V, 2019). The working conditions in the IT field involving many hours of work, pressure, deadlines, and requirement of getting updated with new technologies increases stress levels among employees (Blankenhagel KJ, Theilig MM, Koch H, Witte AK, Zarnekow R, 2019). The study notes that burnout cannot be viewed as a personal issue but rather an organizational one with its source in culture and requirements of the organization (Kresge L, 2020). Precarious work conditions linked to burnout are often mentioned as primary causes of burnout in IT personnel, including work overload, lack of work control, low rewards, and absence of community and fairness at work (Zaza S, Riemenschneider C, Armstrong DJ, 2022).

The consequences of burnout process are significant, they affect the individual and organizational results. Burnout results in reduced job satisfaction, lower commitment to work, and higher levels of absenteeism. Burnout among workers leads to the manifestation of poor thinking resulting in poor decisions, less creativity as well as slower solving problems (Khogali HO, Mekid S, 2023). For organizations, the outcome is not less dire either. Low workforce engagement and burnout leads to high turnover, high recruitment, and training costs, and reduced organizational productivity (Koh DM, Papanikolaou N, Bick U, Illing R, Kahn Jr CE, Kalpathi-Cramer J, Matos C, Martí-Bonmatí L, Miles A, Mun SK, Napel S, 2022). Including loss of employee experience means that knowledge loss is likely to occur and can negatively affect the morale of the rest of the team besides reducing productivity (Djokic A, 2024).

### ***1.2 Technological Solutions to Burnout***

Due to increasing concern about burnout, many technological interventions have been designed to identify or reduce burnout (Ajayi FA, Udeh CA, 2024). These technologies include

small applications that aim to improve the health of employees, and large systems that collect data from different sources to draw a full picture of the health of the workforce (Durodolu OO, Marutha NS, 2024). Smartwatches and other wearable devices that record physical signs such as heart rate and sleep have also been employed to detect stress levels and impending burnout (Chang A, 2023). Moreover, teleworking requires applications that allow employees for constant feedback and pulse surveys to determine their satisfaction levels and signs of burnout. However, one cannot ignore the fact that these technologies come with certain constraints (Borgstadt JT, Kalpas EA, Pond HM, 2022). Wearable devices, however, particularly create privacy issues in the workplace and can be seen as invasive by the employees. Furthermore, the usefulness of these tools is usually dictated by the quality of the obtained data and the possibility of its further processing and analysis. Assessment surveys are informative as they can be used as reactive measures to address a problem after it has been reported by employees (Narayanan LK, Subbiah P, Muralidharan RR, Baskaran AP, Srinivasan V, Baskaran AP, Victor P, Ramachandran H, 2023). This delayed response can also mean a lost chance for early management of the condition.

### ***1.3 AI in Work Pattern Monitoring***

AI (Artificial Intelligence) and ML (Machine Learning) have provided new opportunities to track work patterns and find out the signs of burnout. AI systems are capable of scanning data from emails, project management tools and even time tracking applications looking for signals that suggest burnout (Kuadey NA, Ankora C, Tahiru F, Bensah L, Agbesi CC, Bolatimi SO, 2024). For example, AI can access email data to determine who is working beyond normal working hours or on weekends, which are typical precursors of burnout (Faqihi A, Miah SJ, 2023). There have been documented cases of work pattern monitoring using AI. One good example of the application of AI is in managing and anticipating employee turnover and fatigue at IBM. The AI system used data from the Human Resources department, self-reported data, and work patterns to track likely burnout cases and recommend solutions specific to these cases. Another example is the MyAnalytics tool by Microsoft and it is also an AI tool that aims at enlightening employees on their work patterns; meetings' time, focused work time and hours worked past dark. These insights

assist employees to balance their working and personal lives hence reducing the possibility of burnouts (Koutsimani, P.; Montgomery, A.; Masoura, E.; Panagopoulou, E, 2021).

#### ***1.4 Gaps in Existing Research***

Although tremendous progress has been made in applying technology to screen and control burnout, there are still some issues that need to be addressed in current literature. Another drastic need is the lack of sophisticated AI applications that allow organizations to monitor the mental health of their staff in real-time (Adler DA, Tseng E, Moon KC, Young JQ, Kane JM, Moss E, Mohr DC, Choudhury T, 2022). Most current solutions are either remedial or provide only snapshot updates, which may not be helpful for timely prevention. In addition to that, the problem of merging different types of data is still unsolved. Combination of the various data types like physiological data, work patterns, and feedback must be integrated into a single system to meet the objective of the burnout detection. The fourth gap relates to the absence of studies on the long-term efficacy of AI-assisted interventions (De Stefano VM, 2020). However, knowledge of the long-term effects of such interventions on employee health and organizational performance remains limited, despite clear short-term gains. Furthermore, issues of privacy and security should also be given equal importance to encourage more and more people to embrace these technologies. Employees must be assured that their information is being used appropriately and that other individuals cannot access it (Ozakca NS, Bulus A, Cetin A, 2024).

This research seeks to fill these gaps by creating a more elaborate AI-based system that can monitor work patterns in real-time as a way of identifying and preventing burnout. Through the act of consolidating data feeds and offering ongoing information, the proposed system aims to provide a preventative approach to burnout management, which would benefit the wellbeing of workers and increase productivity in the process.



## 2. Methods

### *2.1 Self-Survey conducted for this research*

Our online survey was conducted for the below Questionnaires I online platform for submitting our questionnaire and getting results from professionals from different industries background.

#### **Questionnaire on Technology Changing Work and Organization:**

- How has technology impacted your day-to-day work tasks?
- Have you received adequate training to adapt to new technologies introduced in the workplace?
- Do you feel that technology has improved efficiency and productivity within the organization?
- Are there any specific areas where you think technology could be better utilized to streamline processes or improve communication?
- How comfortable are you with the pace of technological change within the organization?

### *2.2 System Design and Architecture*

#### **A Centralized Real-Time Database - Mongo DB Atlas Cloud**

A log of a regular basis engagement scores of the employees are tracked over time and thus the data of the employees will be stored on a cloud server with governance compliances. The 'Real-Time Database' will synchronize the employee burnout data and the log and provides a real-time view interface for the IT specialists to view the state of the employees to improve their experience (Luz A, Olaoye G, 2024).

#### **System Design Cloud Server Burnout Detection Mechanism**

1. **Employee Surveys:** Companies often conduct regular surveys to assess employee well-being, job satisfaction, and stress levels. These surveys may include specific questions related to burnout

symptoms, such as exhaustion, cynicism, and reduced productivity. The survey responses can provide insights into the prevalence of burnout within the organization.

**2. Development Programs:** Employee development programs increase productivity, decrease errors or missed deadlines, or a significant decrease in employee engagement can be indicators of burnout.

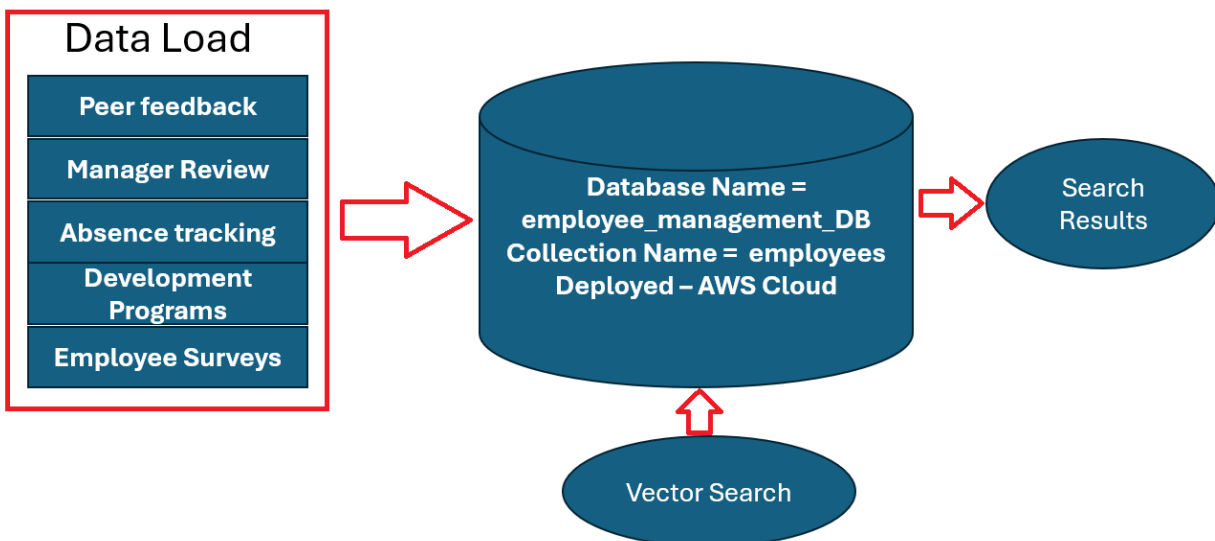
**3. Absence tracking:** Frequent absence can be a red flag for burnout. Tracking these metrics can help identify patterns and individuals who may be experiencing burnout.

**4. Manager Review:** Managers who work closely with employees can often detect signs of burnout. They may notice changes in behavior, such as increased irritability, withdrawal from team activities, or decreased enthusiasm for work. Encouraging open communication and fostering a supportive work environment can help facilitate such observations.

**5. Peer feedback:** Peer feedback mechanisms allow employees to share their experiences and raise concerns related to burnout.

Figure 1: Mongo DB Atlas Cloud

## Mongo DB Atlas Cloud



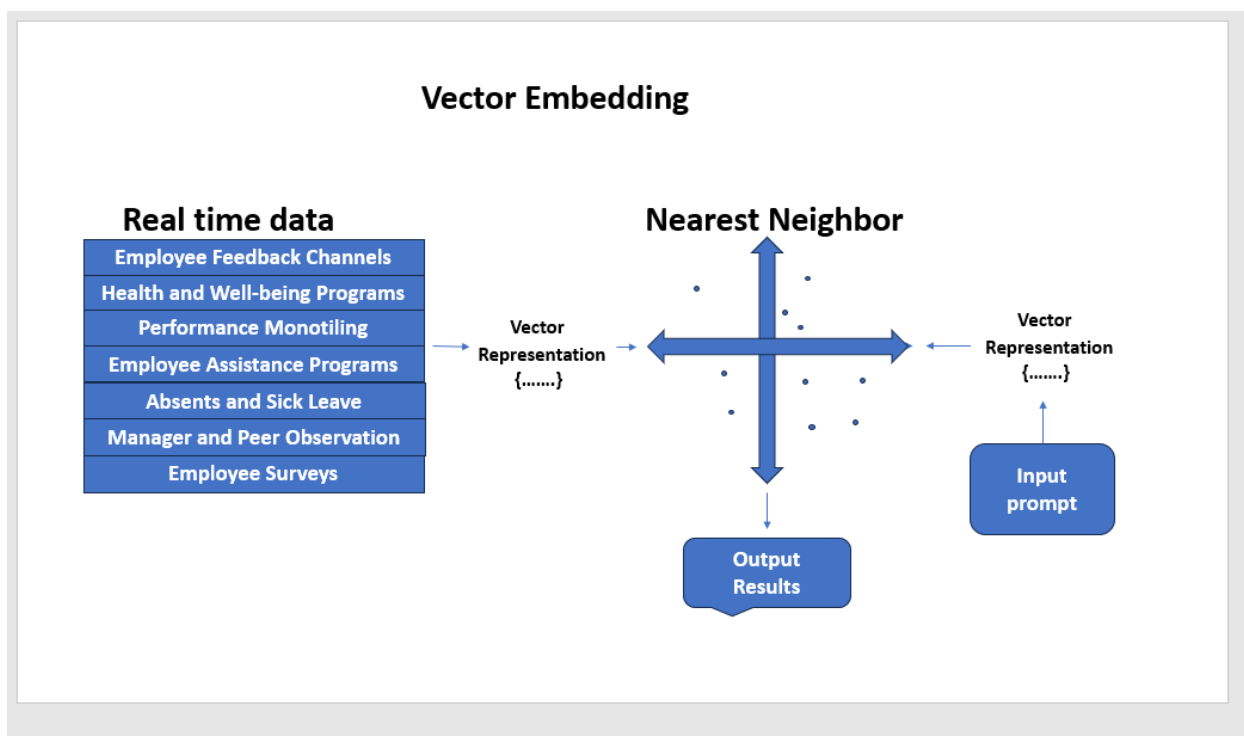
### ***2.3 Vector Search***

HR Management can **use vector search method for understanding the feedback to their questionnaire**. The vector search should understand HR Management queries through work pattern monitoring and natural language processing and gives bright responses to them (De Stefano V, 2024). It will simulate instant connections with users automatically and can yield extremely effective results based on a set of predefined triggers, conditions, or events thus preventing problems from arising in the future. It will eliminate the unnecessary amount of work and take on a substantial load of employees (Algabri HK, 2024). The vector search is based on meaning, this technique helps the machine learning models to transform the text into high dimensional vectors, which refer to embeddings are essentially high dimensional array of numbers that captures the semantic meaning of a word, phrase, or a sentence. Vectors can search through to find similar content based on multiple vectors being near on a high dimensional space. Vector search allows us to find relevant results even when the exact key word is not known (Kavitha M, Roobini S, Prasanth A, Sujaritha M, 2023)**Error! Reference source not found.** The main advantage is semantic understanding, even if the query words are not present in the index but the meaning of phrases are similar, they will be considered as match.

**Vector Embedding - Convert real time data (words and sentences) into numbers that capture their meaning and relationships. These numerical representations help machines understand and process this data more effectively.**

This process helps our ML algorithms find patterns in data and perform our semantic search and give recommendations as output.

Figure 2: Vector Embedding



## ***2.4 System Interfaces and Development***

### ***2.4.1 System Interface: Vectorembdding.py***

The system interface vectorembdding.py is used for loading data into database manually, Large Language Model and Vector Embedding. **I have developed the system interface by using Python that provides an interface for managing employee information and reviews stored in a MongoDB atlas database. I have used different libraries to develop the system. I have used Sentence Transformer Hugging face Interface for encoding text fields into numerical vectors. I have used MongoClient from pymongo for interacting with MongoDB Atlas. Then I used Model Initialization to initialize a pre-trained model from the Sentence Transformers library.**

**Results:** The system results are on employee management, seamlessly integrating with MongoDB. It streamlines various tasks including creating new employee profiles, updating existing employee reviews, and showcasing employee details. By utilizing a pre-trained model for text encoding, the system ensures efficient data handling within the database. Its user-friendly command-line interface empowers users to effortlessly navigate functionalities such as viewing employee information, adding new entries, updating reviews, and gracefully exiting the program. In essence, the system delivers a robust solution for optimizing employee management workflows.

### ***2.4.2 System interface: VectorSearch.py***

This system interface is a vector search or HR search interface that allows querying employee data stored in a MongoDB atlas database. **I have used different Imports and necessary libraries, SentenceTransformer and util from sentence\_transformers for encoding text fields and utility functions. MongoClient from pymongo for interacting with MongoDB atlas.**

The Model Initialization - It initializes a pre-trained model from Sentence Transformers for encoding text fields into numerical embeddings. I have created different functions for the vector search process and receive output.

Prompts (Questions) the user can input are focused on HR-related questions and provides corresponding answers based on the queried data. This system provides a flexible interface for HR professionals to query and analyze employee data stored in MongoDB atlas, facilitating tasks such as talent management, skill assessment, and performance evaluation.

**Results:** The HR search interface or vector search system allows for a wide range of queries on employee data stored in a MongoDB database, providing valuable insights for HR professionals. Utilizing this system interface, I can target a specific set of job titles, be it software engineers, for subsequent communication or assessment of the roles within the organization. Total figures for employment broken down by machine learning and AI expertise reveal insights concerning employees' skills and the possibility of establishing training within the company. Helps to identify the employee with the best performance review enables them to appreciate excellence and keep up the good performance among the workers.

As with the previous technique, using AI/ML-related keywords when evaluating the state of development programs or self-survey questions allows for proper assessment of the organization's preparedness for leveraging the technology and the further potential need for skills training programs. Managers with high overall management scores: Employees with high managerial ratings and frequent AI/ML references can be valuable traits or promotions for managers or specialists in diverse industries. The findings yielded from the application of the proposed HR search interface aid in decision making, organizational training, development, and future planning.

## ***2.5 Minimum Viable Product (MVP) and Results***

### *2.5.1 MVP and Results for System Interface: Vectoreembedding.py*

**Introduction to MongoDB Atlas**, which runs on a cloud platform, which provides scalable storage and computing resources. The cluster is deployed in a specific region for optimized performance and latency. MongoDB Atlas automatically configures a replica set to ensure data redundancy and high availability. Access to the cluster is secured using authentication mechanisms. MongoDB Atlas is a fully managed cloud database service provided by MongoDB. It offers numerous advantages over traditional on-premises or self-managed database solutions.

### *2.5.2 To create a MongoDB Atlas database, we need to follow these general steps:*

**Sign up or log in to MongoDB Atlas free account:** If you have not created your MongoDB Atlas, then you will have to create one before proceeding to the next process. If you already have an associated account, then you should sign in with that account.

**Create a New Project:** Projects are partition in Atlas and refer to your database topology and deployment. Start a new project by clicking on the “New Project” icon at the top of the page and configure the project parameters as required.

**Create a Cluster:** In the context of your project, it is necessary to draw a cluster. A cluster is a group of MongoDB nodes responsible for data storage and management. Decide which cloud provider, region, and configuration to use for your Kubernetes cluster. It is possible to first build with the help of a free plan or choose paid in accordance with needs.

**Configure Cluster Settings:** Add more details to configure your cluster like instance size, disk size, you can also configure backup, auto scaling, and encryption options. Leverage these settings depending on the results you want to achieve and the risks you can tolerate.

**Create a Database User:** MySQL Create a database user with the access privileges necessary to access the created database. When fine-tuning additional parameters, you’re usually asked to

provide the username and password for the user and to define the roles (Read/ Write) regarding the possibility to access your database.

**Whitelist IP Addresses:** Start the Network Access service to enable the configuration of which IP addresses are permitted contact with your cluster (Unver A, 2018). It is also important to whitelist the current IP or provide a range of IPs that will be able to connect to the server.

**Connect to Your Cluster:** Once you have created your cluster and your user, MongoDB Atlas automatically generates a connection string that can be used in your application code to connect to your cluster (Kumar W, Noul D, 2024)**Error! Reference source not found..** It contains the log-in details and any other necessary information for establishing a connection with the database.

**Create a Database and Collection:** When you are connected to your cluster, you can design a brand-new database in MongoDB Atlas. To manage your data, you should know that you can create collections into your database. Determine about the data model of your application the fields for your collections.

**Insert Data:** It is time for you to put the data into the collection of the database you have created using MongoDB's CRUD: Create, Read, Update, Delete.

Remember to follow best practices for database design and security when creating your MongoDB Atlas database. It's essential to plan your schema carefully, optimize queries for performance, and ensure proper authentication and authorization mechanisms are in place to protect your data.

## ***2.6 Created my own Database in the MongoDB Atlas***

### ***2.6.1 Database Name:***

The database is named "**employee\_management\_DB**". This is the database where employee-related data is stored.



### 2.6.2 Collections:

The database contains one collection named "**employees**". This collection holds documents representing individual employees. Each document contains various fields describing employee details and reviews.

## 4. Results

### 4.1 MVP and Results for System Interface: *Vectorembdding.py*

#### **Functional Analysis:**

#### **First step is establishing connection to MongoDB Atlas:**

The system Interface which I developed established a connection to the MongoDB Atlas database. It returns a reference to the collection named "employees" within the "**employee\_management\_DB**" database.

#### **Second step is to retrieve employee details by ID:**

The System Interface retrieves employee details based on the provided employee ID. It searches the "employees" collection for a document with a matching **employee\_id** field.

#### **Third step is to create a new employee:**

The System Interface allows the user to input details for a new employee (such as employee ID, first name, last name, and job title). It encodes the text fields into embeddings using the SentenceTransformer model and inserts a new document into the "employees" collection with the provided details.

#### **Fourth step is to update employee review:**

The System Interface prompts the user to enter new review information for a specified employee ID. It retrieves the employee's current details and allows the user to input new review

data, including management review, self-survey, peer feedback, development programs, and absence. It then updates the corresponding fields in the employee document.

### **Main Vector Search Interface:**

The system interface serves as the main user interface for interacting with the system. It displays a menu of options for the user to choose from the options below.

Option 1: Display Employee Information by entering the employee ID.

Option 2: Create a New Employee by providing necessary details.

Option 3: Update Employee Review by entering the employee ID and providing new review information.

Option 4: Exit the program.

**Execution:** This provides a user-friendly interface for managing employee data and reviews in a MongoDB database, offering functionalities for adding new employees, updating reviews, and retrieving employee information. The system interface is a basic interface for managing employee information and reviews in a MongoDB database atlas.

### **Technical Analysis:**

I used Python script to utilize the **sentence\_transformers** library to perform vector embeddings for employee data. In this interacts the Mongo DB database is used for storing the data and, I can retrieve employee details along with their reviews.

I used the **sentence\_transformers library (Open Source from Hugging Face)** which is used for generating vector embeddings from text data. The pymongo library is used for interacting with MongoDB.

I used the functions below **connect\_to\_mongodb()**. This function is used to establish a connection to the MongoDB database and returns the collection object for further operations.

**get\_employee\_details\_by\_id(collection, employee\_id)** is used to retrieve employee details from the MongoDB collection based on the provided employee ID.

**create\_new\_employee(collection)** function allows user to input details of a new employee, encodes textual fields using a pre-trained model, and stores the data along with embeddings in the MongoDB collection.

**update\_employee\_review(collection, employee\_id)** is used to enable the user to update the review details for a specific employee identified by their ID.

**vector\_search\_interface()** Main interface function providing options to display employee information, create a new employee, update employee reviews, or exit the program. It acts as the main menu for interacting with employee data.

**Sentence Transformer Model:** I used the `sentence-transformers/all-MiniLM-L6-v2` model. This model is based on the MiniLM architecture, which is a small version of the well-known language model BERT (Bidirectional Encoder Representations from Transformers). It is fine-tuned on various tasks, including semantic similarity, and capable of generating meaningful embeddings for sentences. The SentenceTransformer model from the `sentence_transformers` library is utilized to generate vector embeddings specifically for textual data (employee names and job titles).

**Generating Embeddings:** When creating a new employee record (**create\_new\_employee()**), the model is used to encode the textual fields (first name, last name, and job title) into numerical embeddings. These embeddings are then stored along with all other employee details in the MongoDB collection.

**MongoDB Interaction:** The script interacts with a MongoDB database to store and retrieve employee information. MongoDB is a NoSQL database that stores data in flexible, JSON-like documents, making it suitable for storing unstructured or semi-structured data like employee records.

**Database Connection:** The `connect_to_mongodb()` function establishes a connection to the MongoDB database hosted on the MongoDB Atlas cloud platform. It returns the collection object that refers to the "**employees**" collection within the "**employee\_management\_DB**" database.

**CRUD Create, Read, Update, Delete Operations:** Retrieve: The `get_employee_details_by_id()` function retrieves employee details by their unique employee ID from the MongoDB collection.

**Create:** New employee records are created using the `create_new_employee()` function, which inserts a new document into the MongoDB collection.

**Update:** Employee reviews can be updated using the `update_employee_review()` function, which modifies the existing document in the MongoDB collection.

**Display:** Employee information, including reviews, can be displayed using various functions within the interface.

## 4.2 Execution of code (Vectorembedding.py) to Load Data into collection and Perform Vector Embedding to Employees Data

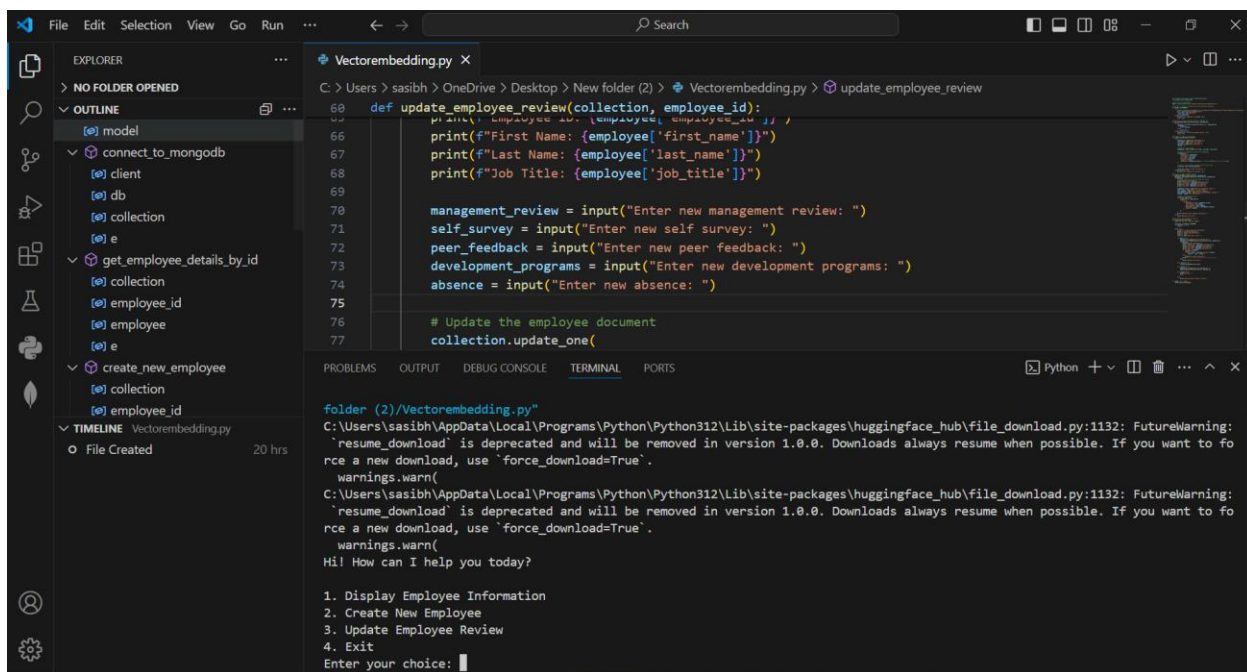
I used VS studio code editor for executing the code:

All the Code is provided in the - Appendices.

For Executed the above code you need to create your own Mongo DB Atlas database and collection.

Result - Please find the screenshot below for the output i received.

Figure 3: Employee ID



The screenshot shows the VS Code editor with the file 'Vectorembedding.py' open. The code in the editor includes a function 'update\_employee\_review' that prints employee details and prompts for management review, self-survey, peer feedback, development programs, and absence. Below the code, the terminal output shows the execution of the script, displaying a menu with four options: 1. Display Employee Information, 2. Create New Employee, 3. Update Employee Review, and 4. Exit. The prompt 'Enter your choice:' is visible at the bottom of the terminal.

```

68 def update_employee_review(collection, employee_id):
69     print(f"Employee ID: {employee_id}")
70     print(f"First Name: {employee['first_name']}")
71     print(f>Last Name: {employee['last_name']}")
72     print(f"Job Title: {employee['job_title']}")
73
74     management_review = input("Enter new management review: ")
75     self_survey = input("Enter new self survey: ")
76     peer_feedback = input("Enter new peer feedback: ")
77     development_programs = input("Enter new development programs: ")
78     absence = input("Enter new absence: ")
79
80     # Update the employee document
81     collection.update_one(

```

```

folder (2)/Vectorembedding.py
C:\Users\sasibh\AppData\Local\Programs\Python\Python312\Lib\site-packages\huggingface_hub\file_download.py:1132: FutureWarning:
'resume_download' is deprecated and will be removed in version 1.0.0. Downloads always resume when possible. If you want to fo
rce a new download, use 'force_download=True'.
  warnings.warn(
C:\Users\sasibh\AppData\Local\Programs\Python\Python312\Lib\site-packages\huggingface_hub\file_download.py:1132: FutureWarning:
'resume_download' is deprecated and will be removed in version 1.0.0. Downloads always resume when possible. If you want to fo
rce a new download, use 'force_download=True'.
  warnings.warn(
Hi! How can I help you today?

1. Display Employee Information
2. Create New Employee
3. Update Employee Review
4. Exit
Enter your choice:

```

**Note:** The output is to enter your choice, if option 1 is selected, the system will prompt to enter employee ID, when employee id is entered the system will display the employee information. When option 2 is selected by entering number 2 and hit enter. This option will help to create a new Employee and the new employee can be created with employee ID, First Name, Last Name, and job title. When option 3 is selected, the system will ask to enter the employee ID, when employee

id is entered, the system will ask to update the reviews of the employee id we entered, the reviews which includes (management\_review, self\_survey, peer\_feedback, development\_programs, absence)

**Now let me execute some tests and find out the results.**

**Test case to display existing employee information using Employee ID**

Once the code is executed the system will display four choices.

1. Display Employee Information
2. Create New Employee
3. Update Employee Review
4. Exit

**Enter your choice: 1**

I have entered 1 and the system displayed option to Enter the Employee ID:

I have entered the existing employee ID as EMP002.

The system will display the existing employee information.

Figure 4: Display the existing employee information.

The screenshot shows a Python IDE with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'Vectorembedding.py' with a sub-project 'Vectorembedding.py' containing files like 'model', 'connect\_to\_mongodb', 'client', 'db', 'collection', 'e', 'get\_employee\_details\_by\_id', 'collection', 'employee\_id', 'employee', 'e', 'create\_new\_employee', 'collection', and 'employee\_id'. The terminal output shows the execution of a menu-based program:

```

`resume_download` is deprecated and will be removed in version 1.0.0. Downloads always resume when possible. If you want to force a new download, use `force_download=True`.
warnings.warn(
Hi! How can I help you today?

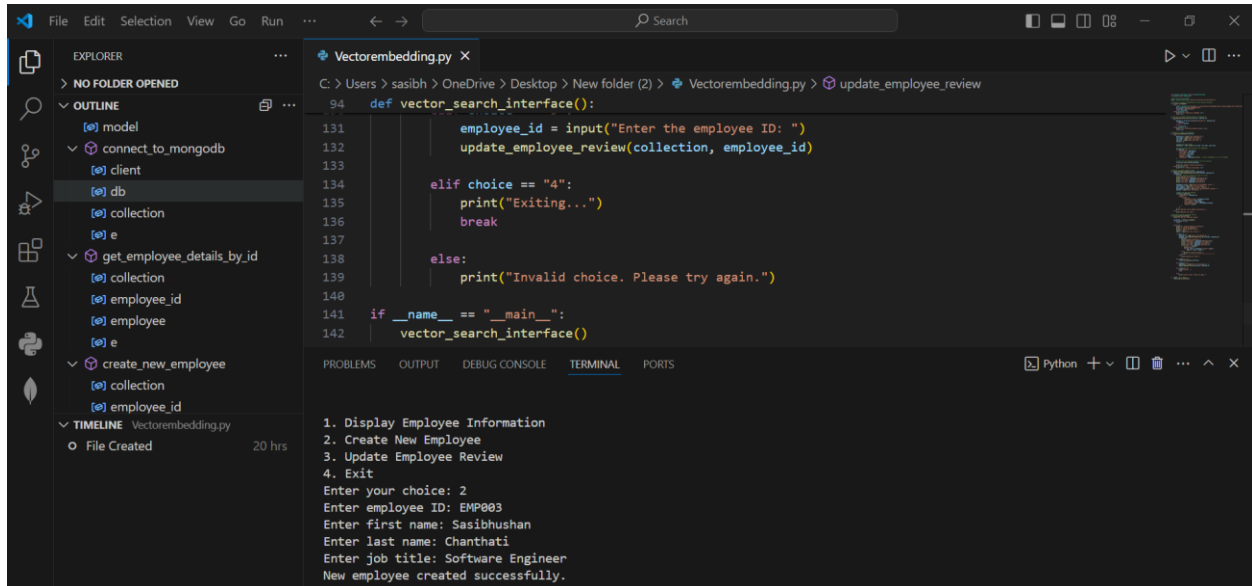
1. Display Employee Information
2. Create New Employee
3. Update Employee Review
4. Exit
Enter your choice: 1
Enter the employee ID: EMP002
Employee Information:
Employee ID: EMP002
First Name: Sam
Last Name: Sam
Job Title: SE

```

**Test case to create a new employee and add information to the new employee.**

Now I have entered the choice as 2 and filled in the necessary information like employee ID, Employee First Name, Employee Last name, and Job Title.

Figure 5: New employee and add information



The screenshot displays a code editor window titled "Vectorembedding.py" with the following Python code:

```
94 def vector_search_interface():
131     employee_id = input("Enter the employee ID: ")
132     update_employee_review(collection, employee_id)
133
134     elif choice == "4":
135         print("Exiting...")
136         break
137
138     else:
139         print("Invalid choice. Please try again.")
140
141 if __name__ == "__main__":
142     vector_search_interface()
```

The terminal output shows the execution of the program:

```
1. Display Employee Information
2. Create New Employee
3. Update Employee Review
4. Exit
Enter your choice: 2
Enter employee ID: EMP003
Enter first name: Sasibhushan
Enter last name: Chanthathi
Enter job title: Software Engineer
New employee created successfully.
```

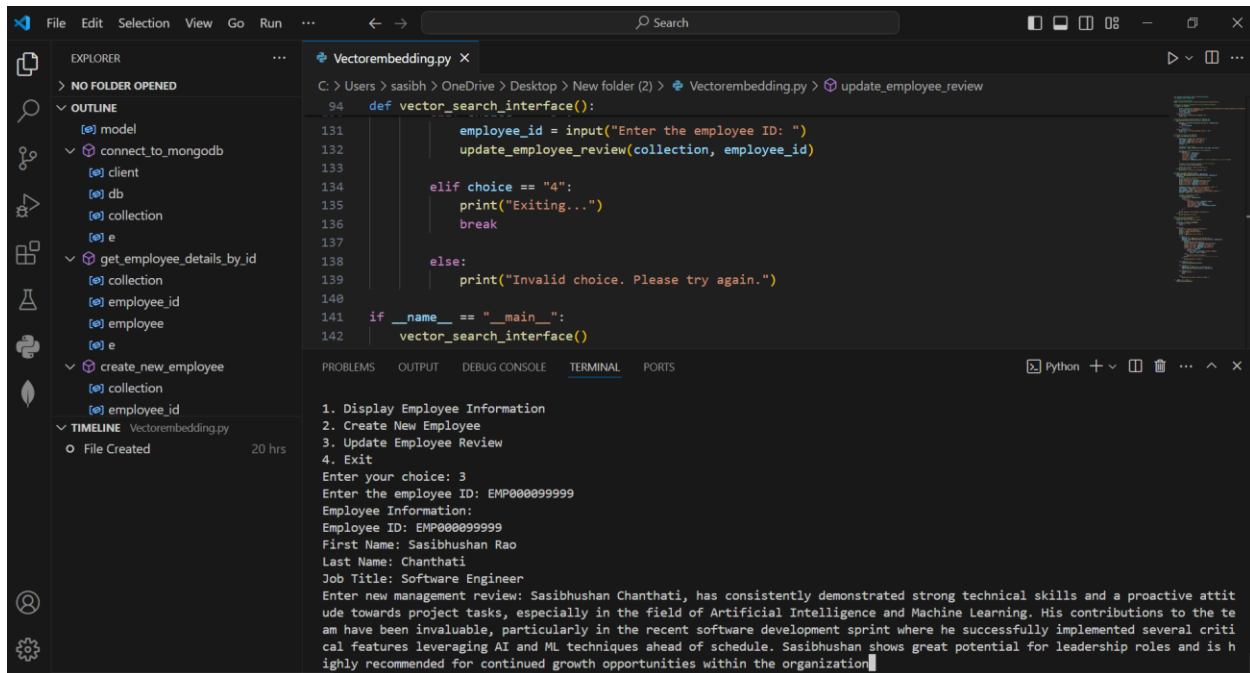
The Explorer pane on the left shows the project structure with folders for "connect\_to\_mongodb", "get\_employee\_details\_by\_id", "create\_new\_employee", and "Timeline". The Timeline pane shows a "File Created" event for "Vectorembedding.py" at 20 hrs.



## Test Case to update Reviews of existing employee.

It displayed all the employee information and gave a prompt to enter the employee's management review.

Figure 6: Update Reviews of existing employee



The screenshot shows a Python IDE with a file named `Vectorembedding.py`. The code defines a `vector_search_interface()` function that prompts the user to enter an employee ID and then updates the review for that employee. The terminal output shows the program running and displaying the following information:

```
1. Display Employee Information
2. Create New Employee
3. Update Employee Review
4. Exit
Enter your choice: 3
Enter the employee ID: EMP000099999
Employee Information:
Employee ID: EMP000099999
First Name: Sasibhushan Rao
Last Name: Chanthati
Job Title: Software Engineer
Enter new management review: Sasibhushan Chanthati, has consistently demonstrated strong technical skills and a proactive attitude towards project tasks, especially in the field of Artificial Intelligence and Machine Learning. His contributions to the team have been invaluable, particularly in the recent software development sprint where he successfully implemented several critical features leveraging AI and ML techniques ahead of schedule. Sasibhushan shows great potential for leadership roles and is highly recommended for continued growth opportunities within the organization.
```

I have Entered the Management Review. Entered the Self Survey, peer feedback, development program and absence.

Figure 7: API integration

The screenshot shows a code editor with a Python script named `vector_search_interface()` in `Vectorembedding.py`. The script includes a function to interact with a MongoDB database and a `__main__` block that calls the function. The terminal output shows the execution of the script, displaying a self-survey and peer feedback for an employee named Sasibhushan.

```

94 def vector_search_interface():
131     employee_id = input("Enter the employee ID: ")
132     update_employee_review(collection, employee_id)
133
134     elif choice == "4":
135         print("Exiting...")
136         break
137
138     else:
139         print("Invalid choice. Please try again.")
140
141 if __name__ == "__main__":
142     vector_search_interface()

```

Terminal Output:

```

ude towards project tasks, especially in the field of Artificial Intelligence and Machine Learning. His contributions to the te
am have been invaluable, particularly in the recent software development sprint where he successfully implemented several criti
cal features leveraging AI and ML techniques ahead of schedule. Sasibhushan shows great potential for leadership roles and is h
ighly recommended for continued growth opportunities within the organization
Enter new self survey: In this self-survey, I reflect on my performance over the past quarter. Overall, I am pleased with my pr
ogress and contributions to the team. I have actively engaged in learning new technologies, particularly in the areas of Artifi
cial Intelligence and Machine Learning, which have helped me enhance my skill set and contribute effectively to project deliver
ables. Additionally, I have collaborated well with team members, providing support and sharing knowledge whenever possible. Loo
king ahead, I aim to further deepen my expertise in AI and ML while continuing to collaborate effectively within the team to dr
ive success
Enter new peer feedback: Sasibhushan has been an exceptional team member over the past quarter. He consistently demonstrates a
strong understanding of complex technical concepts, especially in the field of Artificial Intelligence and Machine Learning. Sa
sibhushan's willingness to share his knowledge and assist others has greatly benefited the team, and his contributions to our p
rojects have been invaluable. He communicates effectively, collaborates well with peers, and always strives for excellence in h
is work. I have full confidence in Sasibhushan's abilities and believe he will continue to excel in his role.
Enter new development programs: As part of Sasibhushan's professional development plan, I recommend enrolling him in an advance

```

As discussed, we can manually enter and can create thousands of records. The second approach we can implement is loading data by using APIs from different data base. We are not using APIs and just loading the database manually. Organizations can afford API integration.

### *4.3 Embedding Arrays created in the database.*

**The same interface Vectorembedding.py will also create embeddings. The steps describe the process.**

Loading Pre-trained Model: The code loaded a pre-trained Sentence Transformer model using the SentenceTransformer class from the sentence\_transformers library. The specific model used in this code is '**sentence-transformers/all-MiniLM-L6-v2**'.

Encoding Text Fields: When we create new employee record (**create\_new\_employee function**), the text fields such as first name, last name, and job title are encoded using the pre-trained model's encode method. The **model.encode** function generates embeddings for the input text. The input consists of lists containing first name, last name, and job title.

Conversion to List: The embeddings generated by the encode method are initially in NumPy array format. MongoDB cannot directly store NumPy arrays. So before inserting the new employee document into the MongoDB collection, the embeddings are converted to lists using the **tolist()** method.

Storing Embeddings in MongoDB: The embeddings, along with all other employee details, are stored in a MongoDB document. The **new\_employee** dictionary contains all the employee information, including the embeddings as a list under the key 'embeddings'. This document is then inserted into the MongoDB collection using **collection.insert\_one(new\_employee)**.

#### *4.4 MVP and Results for Search Interface Vectorsearch.py*

##### **System Interface - Vectorsearch.py**

The HR Search Interface is a tool designed to provide HR-related insights and analytics by querying an employee database stored in MongoDB. It allows users to ask questions related to Employees and Output will be provided. The purpose of this tool is to assist HR professionals in extracting useful information from employee data stored in a MongoDB database. It aims to streamline HR-related inquiries and provide quick, informative responses.

##### **Technical Analysis:**

**connect\_to\_mongodb():** This function we are using to establishes a connection to the **MongoDB database**. **get\_software\_engineers(collection):** This function is to retrieve software engineers from the database. **count\_ml\_ai\_experts(collection):** This function is to count employees with machine learning/AI expertise. **find\_best\_performance(collection):** I am using this to find the employee with the best performance review.

**check\_development\_program\_ai\_ml(collection):** I am using to checks if any development program mentions AI/ML. **check\_self\_survey\_ai\_ml\_certifications(collection):** I am using to checks if any self-survey mentions AI/ML certifications.

**check\_excellent\_management\_review\_ai\_ml(collection):** I am using to check if any employee with an excellent management review mentions AI/ML.

**I have added other placeholder functions for future implementation.**

**get\_project\_managers(collection):** Retrieves project managers.

**count\_python\_proficiency(collection):** Counts employees proficient in Python.

**find\_highest\_absence(collection):** Finds employees with the highest number of absences.

**hr\_search\_interface():** Main function to interact with users, accepts HR-related questions, and provides relevant responses based on the implemented functions. Users interact with the interface by asking HR-related questions. The system responds with relevant information or

alerts if an error occurs. Implementation of placeholder functions for project managers, Python proficiency, and absences. Integration of additional HR-related queries based on organizational needs. Improved error handling and input validation for user queries. The HR Search Interface provides a user-friendly tool for HR professionals to extract valuable insights from employee data stored in MongoDB. By leveraging machine learning embeddings and MongoDB queries, it streamlines HR analytics and decision-making processes.

I have imported the necessary modules such as SentenceTransformer from the sentence\_transformers package and MongoClient from pymongo. **Execution of Code - Vectorsearch.py is attached in the appendix.**

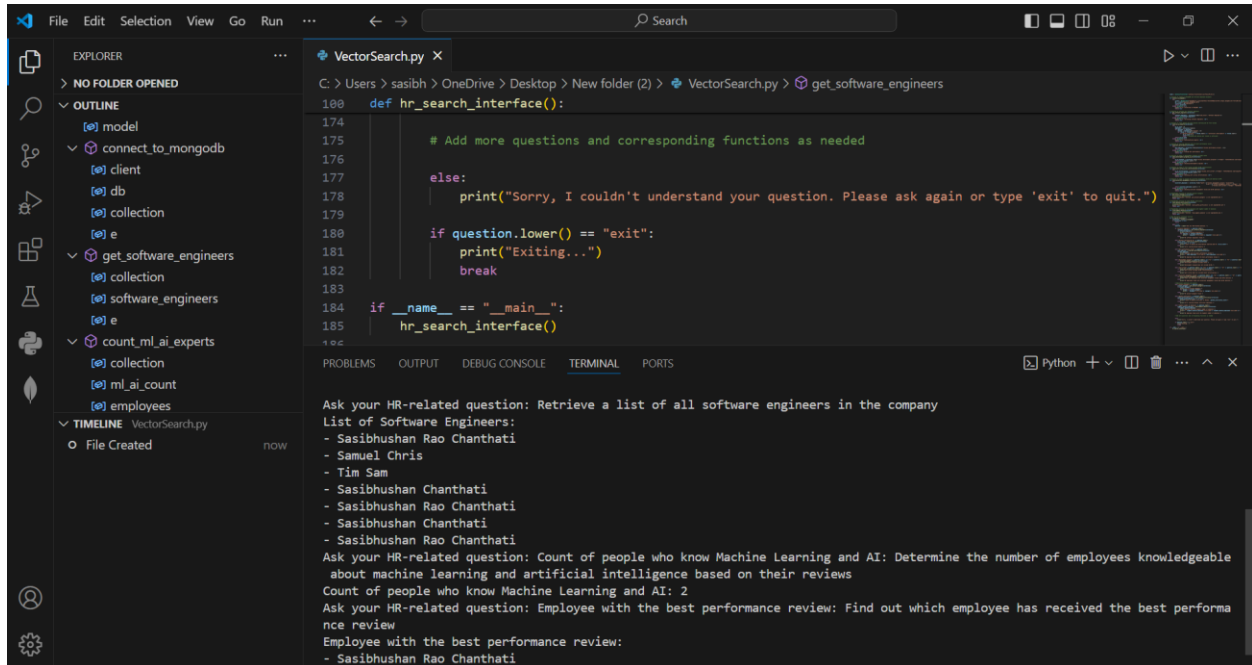
**Result:** I received the below results for the question I entered as an Inputs and interface response for the question which I have given as input and as a result I got answer.

Once the code is executed in VS Code, I entered the below question.

Question: “retrieve a list of all Data Analysts in the company”.

The result is displayed: The names of software engineers as a list. I have created multiple records with different employee IDs with my name. All results are displayed.

Figure 8: Different employee ID's



The screenshot shows a Python IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The code editor displays the following Python code:

```
100 def hr_search_interface():
174
175     # Add more questions and corresponding functions as needed
176
177     else:
178         print("Sorry, I couldn't understand your question. Please ask again or type 'exit' to quit.")
179
180     if question.lower() == "exit":
181         print("Exiting...")
182         break
183
184 if __name__ == "__main__":
185     hr_search_interface()
186
```

The terminal output shows the following interactions:

```
Ask your HR-related question: Retrieve a list of all software engineers in the company
List of Software Engineers:
- Sasibhushan Rao Chanthathi
- Samuel Chris
- Tim Sam
- Sasibhushan Chanthathi
- Sasibhushan Rao Chanthathi
- Sasibhushan Chanthathi
- Sasibhushan Rao Chanthathi

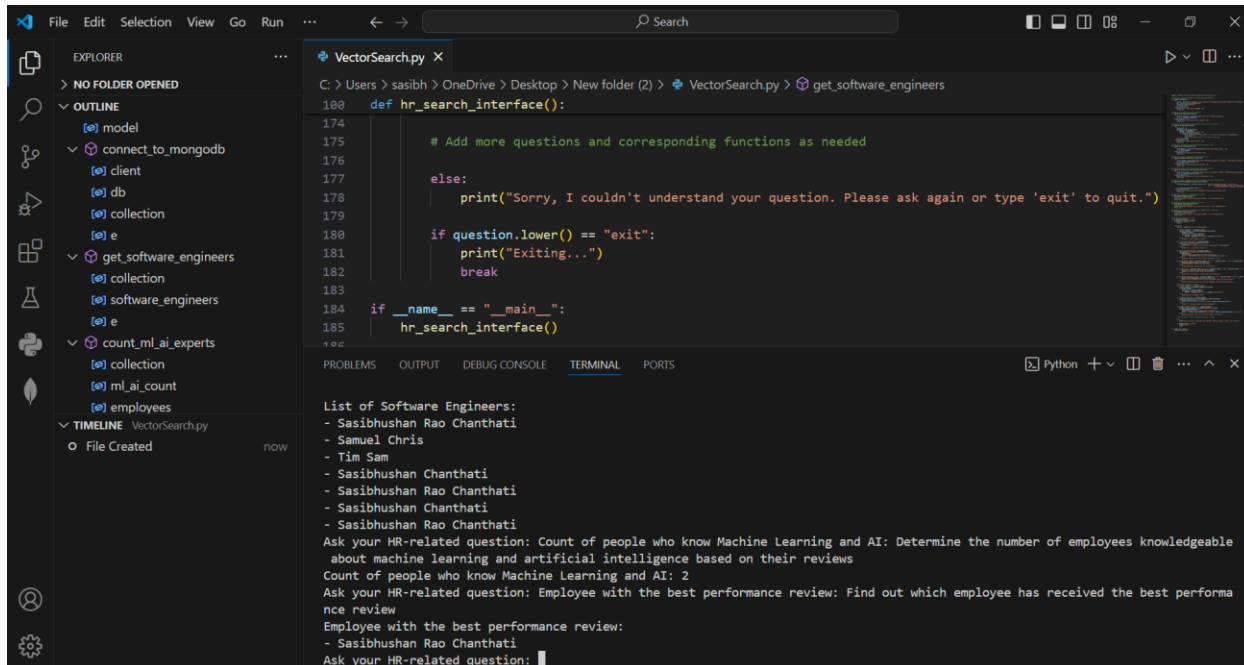
Ask your HR-related question: Count of people who know Machine Learning and AI: Determine the number of employees knowledgeable
about machine learning and artificial intelligence based on their reviews
Count of people who know Machine Learning and AI: 2

Ask your HR-related question: Employee with the best performance review: Find out which employee has received the best performan
ce review
Employee with the best performance review:
- Sasibhushan Rao Chanthathi
```

**Result:** The second question I was asking: Count of people who know Machine Learning and AI: Determine the number of employees knowledgeable about machine learning and artificial intelligence based on their reviews.

The result is displayed below: 2.

Figure 9: Count of people who know Machine Learning and AI

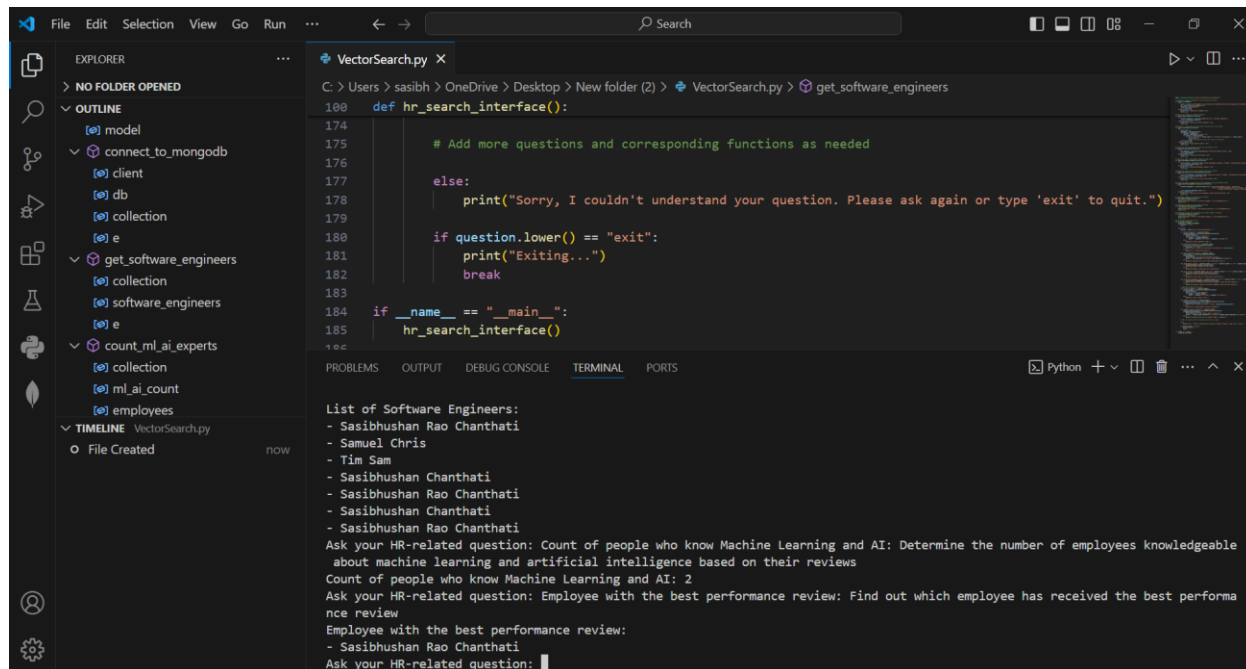


The screenshot shows a code editor with a Python file named `VectorSearch.py`. The code defines a function `hr_search_interface()` that interacts with a user. The terminal output shows the following:

```
List of Software Engineers:  
- Sasibhushan Rao Chanthati  
- Samuel Chris  
- Tim Sam  
- Sasibhushan Chanthati  
- Sasibhushan Rao Chanthati  
- Sasibhushan Chanthati  
- Sasibhushan Rao Chanthati  
Ask your HR-related question: Count of people who know Machine Learning and AI: Determine the number of employees knowledgeable  
about machine learning and artificial intelligence based on their reviews  
Count of people who know Machine Learning and AI: 2  
Ask your HR-related question: Employee with the best performance review: Find out which employee has received the best performa  
nce review  
Employee with the best performance review:  
- Sasibhushan Rao Chanthati  
Ask your HR-related question: 
```

**Result:** The third question I asked: Employee with the best performance review: Find out which employee has received the best performance review. The result is displayed below.

Figure 10: Employee with the best performance review



The screenshot shows a Python IDE with a file named `VectorSearch.py` open. The code defines an `hr_search_interface()` function that handles user input. The terminal output shows the following sequence of events:

```
List of Software Engineers:  
- Sasibhushan Rao Chanthati  
- Samuel Chris  
- Tim Sam  
- Sasibhushan Chanthati  
- Sasibhushan Rao Chanthati  
- Sasibhushan Chanthati  
- Sasibhushan Rao Chanthati  
Ask your HR-related question: Count of people who know Machine Learning and AI: Determine the number of employees knowledgeable about machine learning and artificial intelligence based on their reviews  
Count of people who know Machine Learning and AI: 2  
Ask your HR-related question: Employee with the best performance review: Find out which employee has received the best performance review  
Employee with the best performance review:  
- Sasibhushan Rao Chanthati  
Ask your HR-related question: 
```



**Result:** I am asking this question: List employees who have been promoted in the last six months. The answer is negative. “Sorry Could not understand the question”.

Figure 11: Sorry could not understand the question

The screenshot shows a Python IDE with two files: `VectorSearch.py` and `Vectorembedding.py`. The `VectorSearch.py` file contains the following code:

```

1 from sentence_transformers import SentenceTransformer, util
2 from pymongo import MongoClient
3
4 # Load the pre-trained model
5 model = SentenceTransformer('sentence-transformers/all-MiniLM-L6-v2')
6
7 # Function to connect to MongoDB and retrieve employee documents
8 def connect_to_mongodb():
9     try:
10        client = MongoClient("mongodb+srv://sasichanthati:Test12345@cluster0.1vkkqso.mongodb.net/?retryWri
11        db = client["employee_management_DB"]
12        collection = db["employees"]
13        return collection
14

```

The terminal output shows the following:

```

PS C:\Users\sasibh> & C:/Users/sasibh/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sasibh/OneDrive/Desktop/New
folder (2)/VectorSearch.py"
C:\Users\sasibh\AppData\Local\Programs\Python\Python312\Lib\site-packages\huggingface_hub\file_download.py:1132: FutureWarning:
'resume_download' is deprecated and will be removed in version 1.0.0. Downloads always resume when possible. If you want to fo
rce a new download, use 'force_download=True'.
  warnings.warn(
C:\Users\sasibh\AppData\Local\Programs\Python\Python312\Lib\site-packages\huggingface_hub\file_download.py:1132: FutureWarning:
'resume_download' is deprecated and will be removed in version 1.0.0. Downloads always resume when possible. If you want to fo
rce a new download, use 'force_download=True'.
  warnings.warn(
Ask your HR-related question: List employees who have been promoted in the last six months
Sorry, I couldn't understand your question. Please ask again or type 'exit' to quit.
Ask your HR-related question:

```

To modify an answer for this question “List employees who have been promoted in the last six months”. I have modified the code a little bit and created a second interface.

I modified the code and created a second interface which is `VectorSearchModify.py` I am add the full code so that it would be easy for you to execute. The code is modified to get an answer for the question: “List employees who have been promoted” or with similar meaning. Attached in appendix.

**Result:** Now enter the same question “List employees who have been promoted in the last six months” and verify the output. The result is changed to (No Resent Promoted employee found).

Figure 12: List employees who have been promoted

```

48 def hr_search_interface():
49
50     # Add other HR-related questions and corresponding functionalities here
51
52     while True:
53         question = input("Ask your HR-related question: ")
54
55         if question.lower() == "exit":
56             print("Exiting...")
57             break
58         else:
59             print("Sorry, I couldn't understand your question. Please ask again or type 'exit' to quit.")
60
61     if __name__ == "__main__":
62         hr_search_interface()
63
64
65
66
67

```

Terminal Output:

```

PS C:\Users\sasibh> & C:/Users/sasibh/AppData/Local/Programs/Python/Python312/python.exe "C:/Users/sasibh/OneDrive/Desktop/New
Folder (2)/test5122024.py"
C:\Users\sasibh\AppData\Local\Programs\Python\Python312\Lib\site-packages\huggingface_hub\file_download.py:1132: FutureWarning:
`resume_download` is deprecated and will be removed in version 1.0.0. Downloads always resume when possible. If you want to fo
rce a new download, use `force_download=True`.
  warnings.warn(
C:\Users\sasibh\AppData\Local\Programs\Python\Python312\Lib\site-packages\huggingface_hub\file_download.py:1132: FutureWarning:
`resume_download` is deprecated and will be removed in version 1.0.0. Downloads always resume when possible. If you want to fo
rce a new download, use `force_download=True`.
  warnings.warn(
Ask your HR-related question: List employees who have been promoted in the last six months
No recently promoted employees found.

```

We can Create N number of System Interfaces and merge to same User Interface.

#### 4.5 The Self Survey Results

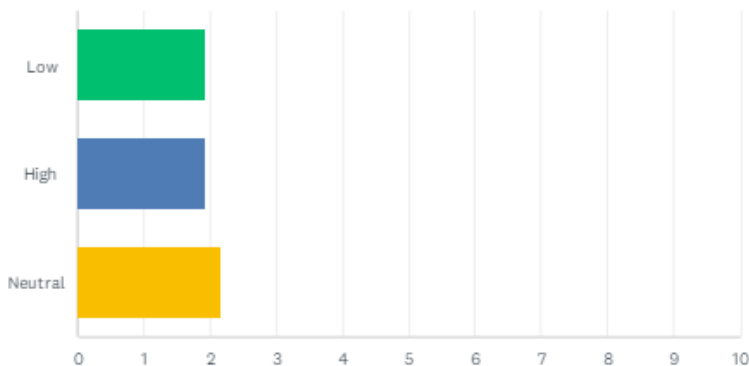
78 people responded to our survey. Below are the self-survey results.

Figure 13: Upgrading Technology

A Centralized Approach to Reducing Burnouts in the Industry Using Work Pattern Monitoring Using Artificial Intelligence

### Q1 Upgrading technology impacted your day-to-day work tasks?

Answered: 78 Skipped: 0



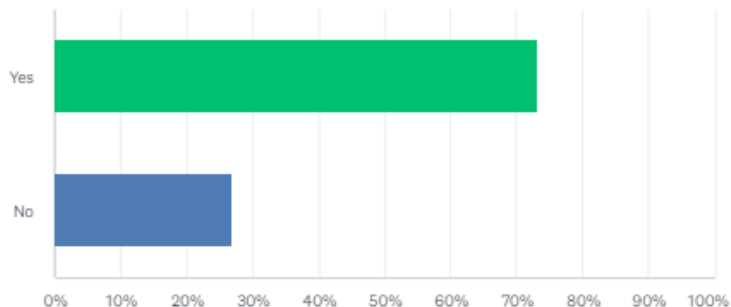
	1	2	3	TOTAL	SCORE
Low	21.79% 17	48.72% 38	29.49% 23	78	1.92
High	35.90% 28	20.51% 16	43.59% 34	78	1.92
Neutral	42.31% 33	30.77% 24	26.92% 21	78	2.15

Figure 14: Adequate training to new technologies

A Centralized Approach to Reducing Burnouts in the Industry Using Work Pattern Monitoring Using Artificial Intelligence

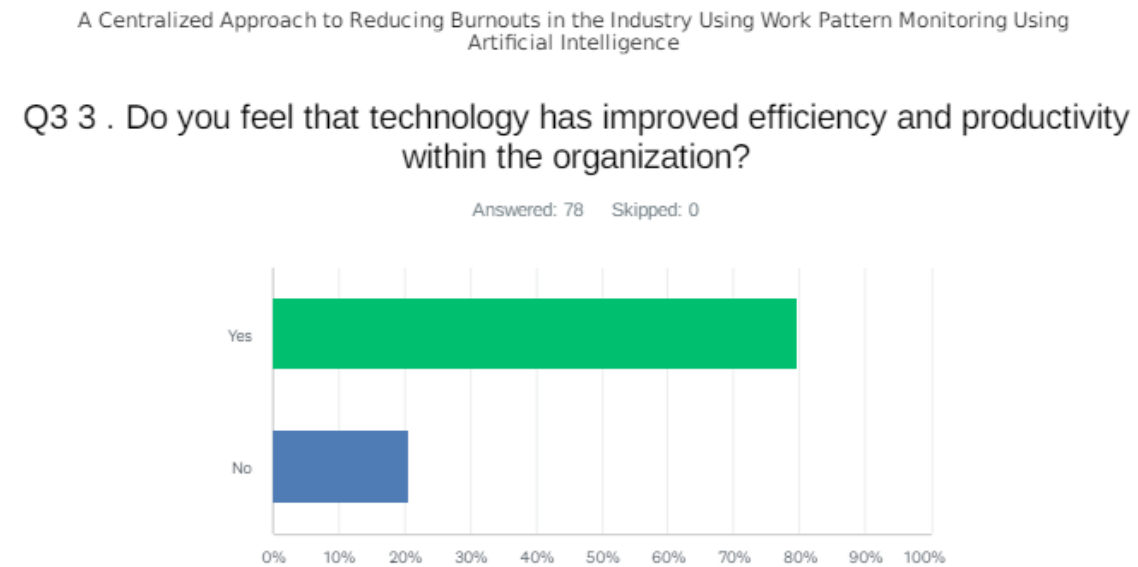
Q2 Have you received adequate training to adapt to new technologies introduced in the workplace?

Answered: 78 Skipped: 0



ANSWER CHOICES	RESPONSES	
Yes	73.08%	57
No	26.92%	21
TOTAL		78

Figure 15: Technology has improved efficiency and productivity within the organization



**Many answered No and Yes for the Q4, and some answers are shown below.**

Figure 16: Technology could be better utilized to improved processes

**Q4**

Are there any specific areas where you think technology could be better utilized to streamline processes or improve communication?

Things that can be automated

A Centralized Approach to Reducing Burnouts in the Industry Using Work Pattern Monitoring Using Artificial Intelligence

Q4 Are there any specific areas where you think technology could be better utilized to streamline processes or improve communication?

Answered: 78 Skipped: 0

---

Q4

Are there any specific areas where you think technology could be better utilized to streamline processes or improve communication?

Things that can be automated

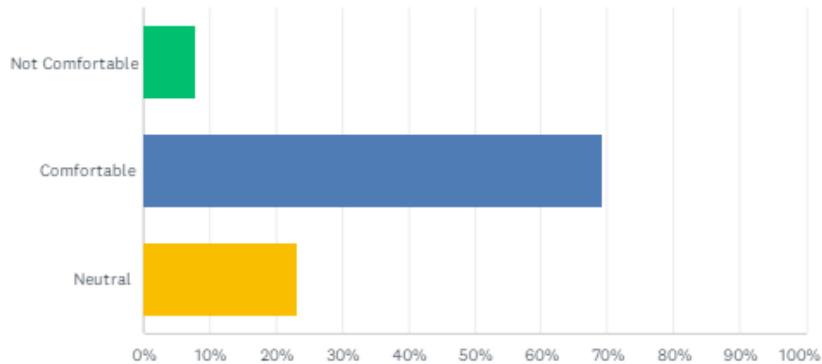
---

Figure 17 Technological change within the organization

A Centralized Approach to Reducing Burnouts in the Industry Using Work Pattern Monitoring Using Artificial Intelligence

Q5 How comfortable are you with the pace of technological change within the organization?

Answered: 78 Skipped: 0



ANSWER CHOICES	RESPONSES	
Not Comfortable	7.69%	6
Comfortable	69.23%	54
Neutral	23.08%	18
TOTAL		78

Figure 18: Gender

A Centralized Approach to Reducing Burnouts in the Industry Using Work Pattern Monitoring Using Artificial Intelligence

### Q1 Gender

Answered: 78 Skipped: 0

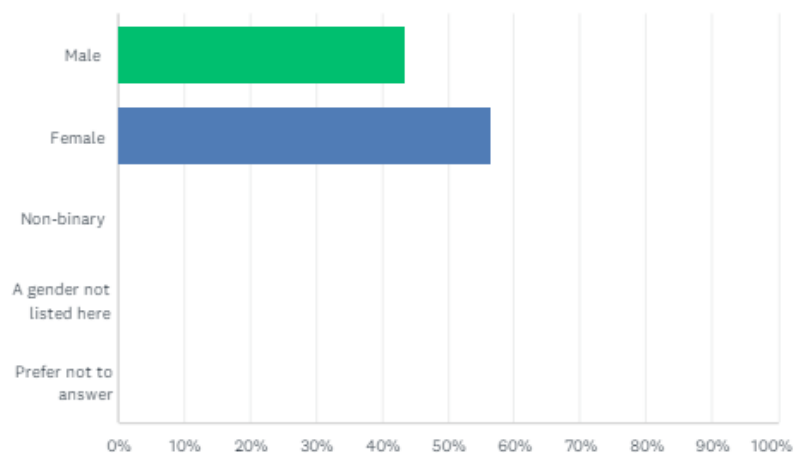


Figure 19: Responses

ANSWER CHOICES	RESPONSES	
Male	43.59%	34
Female	56.41%	44
Non-binary	0.00%	0
A gender not listed here	0.00%	0
Prefer not to answer	0.00%	0
<b>TOTAL</b>		<b>78</b>

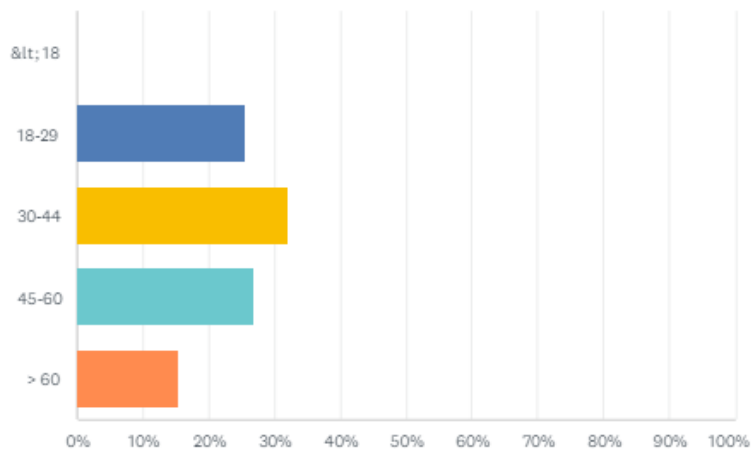


Figure 20: Age

A Centralized Approach to Reducing Burnouts in the Industry Using Work Pattern Monitoring Using Artificial Intelligence

### Q2 Age

Answered: 78 Skipped: 0



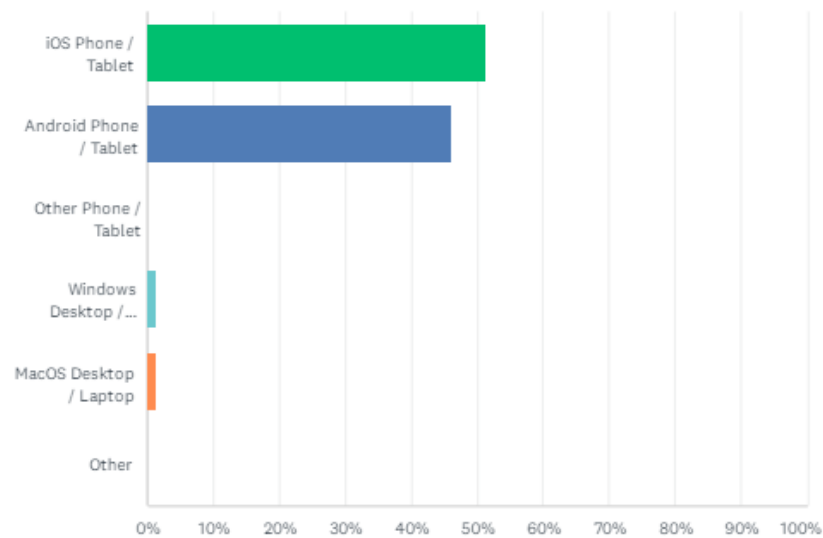
ANSWER CHOICES	RESPONSES	
< 18	0.00%	0
18-29	25.64%	20
30-44	32.05%	25
45-60	26.92%	21
> 60	15.38%	12
<b>TOTAL</b>		<b>78</b>

Figure 21: Device Type

A Centralized Approach to Reducing Burnouts in the Industry Using Work Pattern Monitoring Using Artificial Intelligence

### Q3 Device Type

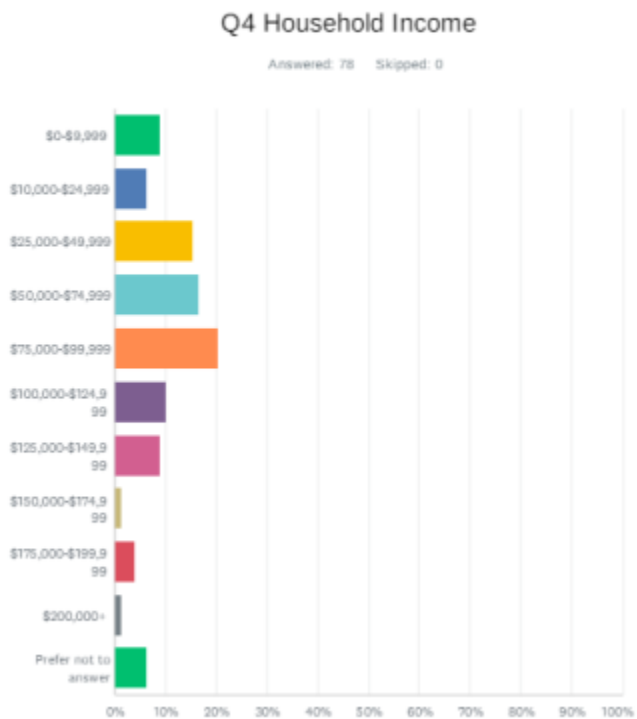
Answered: 78 Skipped: 0



ANSWER CHOICES	RESPONSES	
iOS Phone / Tablet	51.28%	40
Android Phone / Tablet	46.15%	36
Other Phone / Tablet	0.00%	0
Windows Desktop / Laptop	1.28%	1
MacOS Desktop / Laptop	1.28%	1
Other	0.00%	0
<b>TOTAL</b>		<b>78</b>

Figure 22: Household Income

A Centralized Approach to Reducing Burnouts in the Industry Using Work Pattern Monitoring Using Artificial Intelligence



A Centralized Approach to Reducing Burnouts in the Industry Using Work Pattern Monitoring Using Artificial Intelligence

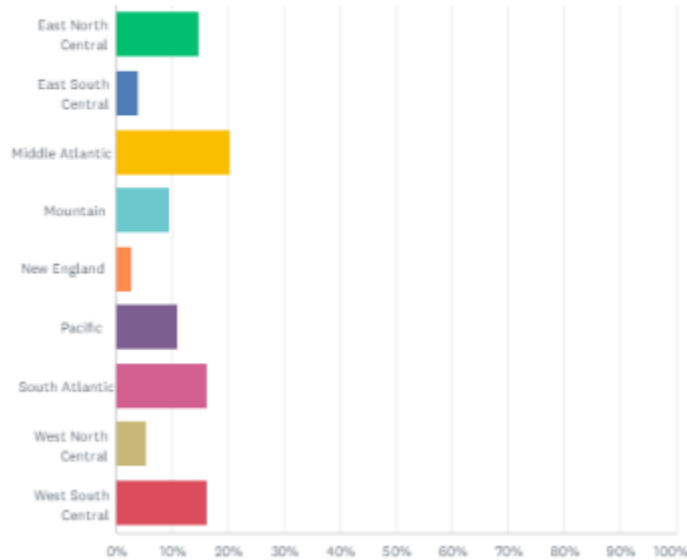
ANSWER CHOICES	RESPONSES	
\$0-\$9,999	8.97%	7
\$10,000-\$24,999	6.41%	5
\$25,000-\$49,999	15.38%	12
\$50,000-\$74,999	16.67%	13
\$75,000-\$99,999	20.51%	16
\$100,000-\$124,999	10.26%	8
\$125,000-\$149,999	8.97%	7
\$150,000-\$174,999	1.28%	1
\$175,000-\$199,999	3.85%	3
\$200,000+	1.28%	1
Prefer not to answer	6.41%	5
<b>TOTAL</b>		<b>78</b>

Figure 23: Major Vs Region

A Centralized Approach to Reducing Burnouts in the Industry Using Work Pattern Monitoring Using Artificial Intelligence

Q5 Major US Region

Answered: 74 Skipped: 4



ANSWER CHOICES	RESPONSES	
East North Central	14.86%	11
East South Central	4.05%	3
Middle Atlantic	20.27%	15
Mountain	9.46%	7
New England	2.70%	2
Pacific	10.81%	8
South Atlantic	16.22%	12
West North Central	5.41%	4
West South Central	16.22%	12
TOTAL		74

## 5. Findings and Discussion

The opinions gathered from the questionnaire were several key findings on the current state of technological integration and its influence on the rates of burnout within the IT sector. Most respondents seem to conclude that the use of technology has had a moderate to positive effect on their working activities meaning that though most people feel positively about the use of technology, there are inherent issues. As might be observed, the largest percentage (29.49%) saw the negative aspect of technology, possibly pointing out sectors where technology could be making them anxious as opposed to providing them with consolation. From the survey, it can be deduced that significantly more than half of those who took the survey felt prepared to handle new technologies and the result was 73.08% for the affirmative. This shows that the training programs offered in their organizations are mostly fruitful.

Still, there remains 26.92% of the patients who stated that they did not feel prepared enough, which indicates a potentially important group of the clients that could be provided with better training and assistance. When asked whether efficiency and productivity had increased, decreased or remained unchanged, many chose the option that technology had an overall positive effect with 79.49%. This we believe, affirms the possibility of improved performance in organizations through the adoption of more advanced technologies. However, the 20.51% that reported not observing such enhancements points to possibilities of challenges that are either on the adoption or application of these technologies.

Respondents were also asked about areas that could be improved regarding the application of technology in their organization and, although mostly the answers provided focused on general ideas, several respondents referred to specific needs, for instance, better communication tools within the organization and improved project management tools. This feedback is imperative for organization which aims to enhance the business technology approach and strengthen the support for employees.

Another aspect was the perceived comfort with the rate of the technological advancement and most of the participants (69. 23%) described themselves as comfortable. Still, 7. 69% described their feeling as being uncomfortable this could suggest that there are some employees who feel challenged or overwhelmed by the fast pace of change. Caring for the needs of such individuals is important in maintaining a healthy workforce and high job satisfaction.

Some of the additional qualitative responses that were collected from the survey participants regarding the areas that can be improved for further use of technology added more insight to the survey results. Participants described general patterns of what could be optimized or added in the field of tools and technology to support work and communication. For example, some respondents pointed out that there is a lack of effective tools for real-time collaboration, which are relevant for remote workers and team members who work asynchronously and rarely meet face to face, especially in the new normal of the post-COVID-19 era.

Some called for more simplicity in the current project management tools and methods due to what they noted to be an overload of paperwork that takes time away from employees to achieve their main goals. It also featured calls for the inclusion of deeper analytics and artificial intelligence that can assist a person with their workload and detect the first signs of stress or burnout.

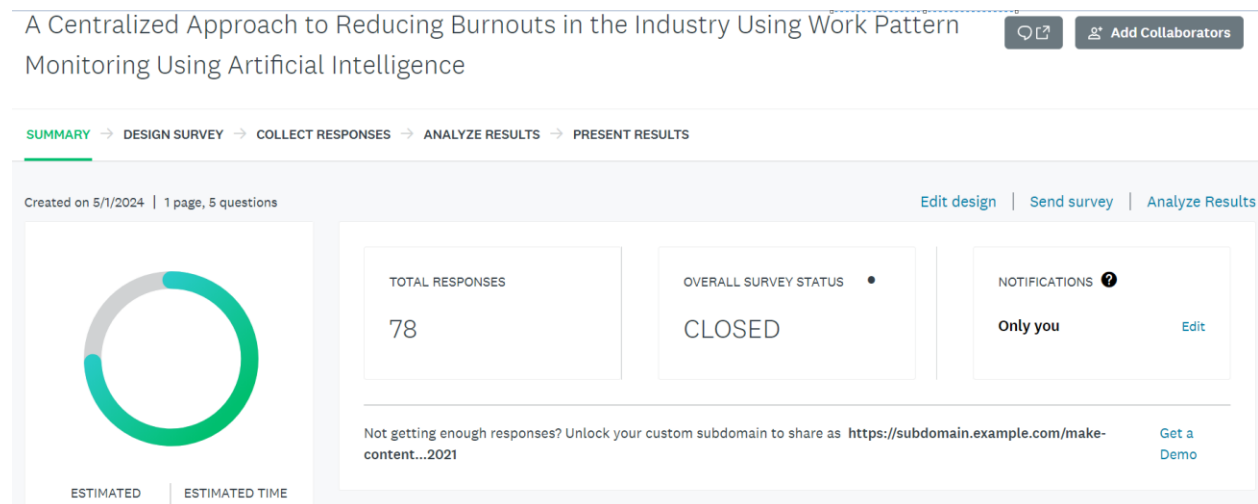
For this reason, the information obtained from the self-survey emphasizes several areas in which technology can support the process of decreasing burnout rates and increasing engagement among employees. There is a clear need to continue technological development and to strengthen some aspects as obtained from respondents' views. Managers can utilize these ideas to better develop their technological plans, for them not only adopt new technologies but also recognize the need to offer training and assistance in order to better implement them. Further, the feedback for the areas requiring improvements can be used in the future to invest more in technological tools to improve communication as well as collaboration and project management tools.

By addressing these needs, it implies establishing more congenial organizational conditions that could minimize burnout and enhance performance and satisfaction. The author's self-survey

results give the reader a good starting point from which to find statistics about technology and burnout affecting the IT industry. They stress the need to proceed with improvement and enhance the support needed to promote the technological implementation to foster the improvement of the work environment and its health.

After reviving quality surveys, employee engagement scores are returned after performing the mechanism that can be monitored over time. So, look at all employee survey and questionnaire responses and rapidly verify the ‘why’ behind the dimension’s engagement scores. Work-life balance, management, teamwork, and appreciation. The industry specialists within the corporation can act after reviewing the report and on the perceptions from the AI-driven data. After viewing the insights, industry specialists can understand what influences employee contentment and then take action to make more optimistic where applicable. I can forward the survey results on request – (Please send me your email id for survey results)

Figure 24: Work Pattern using AI



### 5.1 Advantages based on Technology.

Let me explain every point and their potential advantages in detail for real time:



**Vector Search Interface:** This code focuses on creating a vector search interface for managing employee data stored in a MongoDB database (Dimitriu MC, Pantea-Stoian A, Smaranda AC, Nica AA, Carap AC, Constantin VD, Davitoiu AM, Cirstoveanu C, Bacalbasa N, Bratu OG, Jacota-Alexe F, 2020). It utilizes the Sentence Transformer model to generate embeddings for textual data such as employee names and job titles.

**Semantic Search:** Storing textual information in high dimension vectors captures semantic similarities between different input texts (Tiwari R, 2024). This will allow for the performance of semantic search operations in which one can easily search for similar or related string entries.

**Efficient Retrieval:** Vectors make it easier to search for relevant information since they reduce the time taken in searching on the internet. It enables - the basic but efficient search of employees on their Identification numbers and other features.

**Scalability:** When data is large or the application is trafficked, using data storage like MongoDB along with the vector representation will be quite beneficial.

**Natural Language Processing (NLP) Applications:** The conversion of text into meaningful representations helps in several natural language processing activities, including sentiment analysis, document classification and clustering, and similarity analysis (Solanky MD, Gupta S, 2022).

**HR Search Interface:** This code uses python to develop an application that searches information related to employees from a MongoDB database. It provides features specific for HR operations and search.

**Customized Querying:** It enables one to ask general questions about the data such as how many software engineers the company does have, which employees have a certain skill or employees whose performance review is above average.

**Insight Generation:** Thus, by searching through the whole database in line with some parameters, the HR professionals can get the desired information about some aspects of the workers including

skill availability, performance, and training requirements (Abavisani M, Khoshrou A, Foroushan SK, Sahebkar A, 2024).

**Decision Support:** The above interface can be used in supporting the HR professionals in decision-making processes since it avails information concerning the workforce.

**Automation:** With helping to answer more mechanical questions like which employees possess a particular skill or credential, the interface can also help lessen the HR workload.

**Overall Advantages:** Real time: The utilization of the vector representations to enhance the search and retrieval of information in both interfaces results in the enhancement of the performance of real time systems.

**Customization:** These interfaces can be further extended and configured as per the need or requirement of the organization or application being used (Rosenfeld A, Benrimoh D, Armstrong C, Mirchi N, Langlois-Therrien T, Rollins C, Tanguay-Sela M, Mehlretter J, Fratila R, Israel S, Snook E, 2021).

**Data-Driven Insights:** Therefore, with real-time analysis and querying, employee information can be effectively interrogated for organizational advantage in decision-making and planning.

**Automation and Streamlining:** It can be faster, more cost-effective and timesaving to develop automated systems and options for manual HR and queries responses.

**Scalability:** Scalability of the applied database such as MongoDB or using vector representations enables accommodating vast amounts of data and future user increases. All these will help in querying and analyzing the data that concerns the employees and enhances organizational performance and management of people.

This achieved is discussed based on some set of employees with technological perception. In real time scenarios, millions of employee data in real-time by keeping all the data of employee's records can be loaded. As it is the Cloud-based infrastructures support software interfaces that will provide

environmental proactivity and powering virtual services such as vector search facility to improve the energy efficiency of employees.

Burnout Detection Mechanism can make organizations more desirable to prospective employees and maintain the individuals they appoint for much longer. The industry specialists within the corporation can act after reviewing the analysis report and on the perceptions from the AI-driven data. After viewing the insights, industry specialists can understand what influences employee contentment and then take action to make decisions more optimistic where applicable.

Employees will be appreciative that work pattern monitoring using Artificial Intelligence works well to keep them contented and the industry generally will earn the benefits.

The 'MongoDB Database' will synchronize the employee burnout data and the log and provides a real-time view interface for the specialists to view the state of the employees to improve their experience.

## ***5.2 Implementation of this process in financial domain and advantages.***

To adapt the interfaces for financial customers and the financial product purchases, we need to revise the functionality to focus on managing products, purchases, and feedback.

### **Product Management System Interface:**

Create functions to manage products, including adding new products, updating product information, and deleting products from the database. Implement features to categorize products by type, brand, price range, or any other relevant attributes. Integrate functionality to retrieve product details, such as product descriptions, prices, availability, and customer ratings. Customize the interface to display product information, allowing financial customers to browse and search for products based on their preferences.

### **Purchase Management Interface:**

Design functions to facilitate the purchasing orders, such as capturing the purchasing details which may include the date of purchase, details of the customer, product code, number of items purchased, and total cost. Prescribe elements that will enable the monitoring of stock positions and modify product status in accordance with client demand (Tanner EC, Epler R, Tanner JF, 2022). Extend and integrate the payment processing part to ensure the safe execution of transactions. Some of the samples of report/summary on the purchase history are sales records, most popular products and sales revenue records, and trends over time.

### **Feedback Management Interface:**

Design a system for capturing and processing feedback based on purchased products by the customer. Activities: Incorporate additional functionalities that allow customers to give ratings, write reviews or leave comments on quality, satisfaction as well as feedback on the improvements that can be made. Create discrete functions for feedback data treatment, for instance, sentiment analysis to measure overall customer satisfaction and potential product improvement areas.

To support this, offer the customers an opportunity to view feedback and submit them through the interface as conveniently as possible.

**Search and Analysis Interface:**

Design procedure for searching and evaluating product information according to specific parameters, including product characteristics, reviews from the customers, or the overall sales figures. Divide features aimed at producing detailed reports and graphs illustrating product effectiveness, evaluations from customers, and sales data. Help the financial customers to analysis the most popular products, customer preferences, and opportunities for the development of business.

**Integration with Financial Systems:**

Synchronize the interfaces with financial tools to record revenue, costs, and profitability linked to the products. It is critical to introduce settings that would allow tracking essential financial coefficients tied to machinery performance, including gross margin, ROI, and inventory turnover. Maintain data purity and legal requirements regarding financial operations as well as protect financial records and specific customer information. If the application's interfaces and functionality are properly aligned with these primary financial customer interests, namely, product management, purchasing, and feedback, then financial customers will be given powerful tools for improving product portfolios, increasing customer satisfaction, and thus, the business.

## 6. Conclusion

This research paper shows how self and other's work pattern monitoring and the utilization of Artificial Intelligence in a centralized manner can help in the prevention of burnout especially in the IT industry. With the proposed system using artificial intelligence and real-time data analysis, the early symptoms of burnout can be detected, and appropriate actions be initiated promptly, thereby enhancing the health of the employees and effectiveness of the organization. Centralizing the real-time database and the use of AI vector search user interfaces make it convenient and efficient to address the issue of employee burnout. This approach does not just seek to treat the symptoms of the burnout problem but goes further to offer a roadmap for constant evaluation. This integration points to the increased development of technology in the field of human resource management and its applicability toward improving the workplace climate and the well-being of the employees.

## References

- Sasibhushan Rao Chanthati , (2022 ) " A Centralized Approach to Reducing Burnouts in the it Industry Using Work Pattern Monitoring Using Artificial Intelligenc " , International Journal of Soft Computing And Artificial Intelligence (IJSCAI) , pp. 65-69, Volume-10,Issue-1  
<https://iraj.doionline.org/dx/IJSCAI-IRAJ-DOIONLINE-18853>
- Ajayi FA, Udeh CA. Combating Burnout in the it industry: a review of employee well-being initiatives. International Journal of Applied Research in Social Sciences. 2024 Apr 10;6(4):567-88.
- Gandhi TK, Classen D, Sinsky CA, Rhew DC, Vande Garde N, Roberts A, Federico F. How can artificial intelligence decrease cognitive and work burden for front line practitioners?. JAMIA open. 2023 Oct 1;6(3):ooad079. DOI: 10.1093/jamiaopen/ooad079
- Rožman M, Oreški D, Tominc P. Artificial-intelligence-supported reduction of employees' workload to increase the company's performance in today's VUCA Environment. Sustainability. 2023 Mar 12;15(6):5019.

Mer A, Viridi AS. Navigating the paradigm shift in HRM practices through the lens of artificial intelligence: A post-pandemic perspective. *The Adoption and Effect of Artificial Intelligence on Human Resources Management, Part A*. 2023 Feb 10:123-54.

Sonderling KE, Kelley BJ, Casimir L. The promise and the peril: artificial intelligence and employment discrimination. *U. Miami L. Rev.*. 2022;77:1.

De Stefano, Valerio, 'Negotiating the Algorithm': Automation, Artificial Intelligence and Labour Protection (May 16, 2018). *Comparative Labor Law & Policy Journal*, Vol. 41, No. 1, 2019, Available at SSRN: <https://ssrn.com/abstract=3178233> or <http://dx.doi.org/10.2139/ssrn.3178233>

Blankenhagel KJ, Theilig MM, Koch H, Witte AK, Zarnekow R. Challenges for preventive digital stress management systems-identifying requirements by conducting qualitative interviews. DOI: 10.24251/HICSS.2019.461

Koh DM, Papanikolaou N, Bick U, Illing R, Kahn Jr CE, Kalpathi-Cramer J, Matos C, Martí-Bonmatí L, Miles A, Mun SK, Napel S. Artificial intelligence and machine learning in cancer imaging. *Communications Medicine*. 2022 Oct 27;2(1):133.

Chang A. The role of artificial intelligence in digital health. In *Digital health entrepreneurship 2023* Sep 5 (pp. 75-85). Cham: Springer International Publishing.

Borgstadt JT, Kalpas EA, Pond HM. A Qualitative Thematic Analysis of Addressing the Why: An Artificial Intelligence (AI) in Healthcare Symposium. *Cureus*. 2022 Mar 31;14(3).

Faqihi A, Miah SJ. Artificial intelligence-driven talent management system: Exploring the risks and options for constructing a theoretical foundation. *Journal of Risk and Financial Management*. 2023 Jan 4;16(1):31.

Xames MD, Topcu TG. Toward Digital Twins for Human-in-the-loop Systems: A Framework for Workload Management and Burnout Prevention in Healthcare Systems. In *2023 IEEE 3rd International Conference on Digital Twins and Parallel Intelligence (DTPI) 2023* Nov 7 (pp. 1-6). IEEE.

Adler DA, Tseng E, Moon KC, Young JQ, Kane JM, Moss E, Mohr DC, Choudhury T. Burnout and the quantified workplace: tensions around personal sensing interventions for stress in resident physicians. *Proceedings of the ACM on Human-computer Interaction*. 2022 Nov 11;6(CSCW2):1-48. DOI: 10.1145/3555531

De Stefano VM. 'Negotiating the algorithm': Automation, artificial intelligence and labour protection. *Comparative Labor Law and Policy Journal*. 2020 Jul 1;41(1):1-32.

De Stefano V. Algorithmic bosses and what to do about them: automation, artificial intelligence and labour protection. *Economic and policy implications of artificial intelligence*. 2020:65-86.

Kavitha M, Roobini S, Prasanth A, Sujaritha M. Systematic view and impact of artificial intelligence in smart healthcare systems, principles, challenges and applications. *Machine Learning and Artificial Intelligence in Healthcare Systems*. 2023 Jan 3:25-56.

Unver A. Artificial intelligence, authoritarianism and the future of political systems. *EDAM Research Reports*. 2018 Jul 15.

Kumar W, Noul D. Artificial Intelligence and Sales Management: Enhancing Self-Efficacy and Reducing Stress for Industrial Sales Executives in Latin America. DOI: 10.13140/RG.2.2.19777.16486

Dimitriu MC, Pantea-Stoian A, Smaranda AC, Nica AA, Carap AC, Constantin VD, Davitoiu AM, Cirstoveanu C, Bacalbasa N, Bratu OG, Jacota-Alexe F. Burnout syndrome in Romanian medical residents in time of the COVID-19 pandemic. *Medical hypotheses*. 2020 Nov 1;144:109972.

Tiwari R. The Impact of Artificial Intelligence in the Workplace and its Effect on the Digital Wellbeing of Employees. DOI: 10.5281/zenodo.10936348

Solanky MD, Gupta S. Brain and Behavior: Blending of Human and Artificial Minds Toward Stress Recognition and Intervention in Organizational Well-Being. *Impact of Artificial Intelligence on Organizational Transformation*. 2022 Feb 7:201-27.

Abavisani M, Khoshrou A, Foroushan SK, Sahebkar A. Chatting with artificial intelligence to combat antibiotic resistance: Opportunities and challenges. *Current Research in Biotechnology*. 2024 Mar 11:100197.

Rosenfeld A, Benrimoh D, Armstrong C, Mirchi N, Langlois-Therrien T, Rollins C, Tanguay-Sela M, Mehlretter J, Fratila R, Israel S, Snook E. Big Data analytics and artificial intelligence in mental healthcare. In *Applications of big data in healthcare 2021* Jan 1 (pp. 137-171). Academic Press.

Kaufman R. Understanding and Gauging Burnout in Agile Information Technology Teams (Master's thesis, The College of St. Scholastica).

Tanner EC, Epler R, Tanner JF. Masking the role or masking the toll? The effects of career fit on salesperson burnout. *Journal of Business-to-Business Marketing*. 2022 Oct 2;29(3-4):335-52.



Khogali HO, Mekid S. The blended future of automation and AI: Examining some long-term societal and ethical impact features. *Technology in Society*. 2023 May 1;73:102232.

Narayanan LK, Subbiah P, Muralidharan RR, Baskaran AP, Srinivasan V, Baskaran AP, Victor P, Ramachandran H. A survey on AI-and ML-based demand forecast analysis of power using IoT-based SCADA. *Smart Energy and Electric Power Systems* 2023 Jan 1 (pp. 65-78). Elsevier.

Leitão J, Pereira D, Gonçalves Â. Quality of work life and contribution to productivity: Assessing the moderator effects of burnout syndrome. *International Journal of Environmental Research and Public Health*. 2021 Mar 2;18(5):2425. doi: 10.3390/ijerph18052425

Gordon S, Adler H. Challenging or hindering? Understanding the daily effects of work stressors on hotel employees' work engagement and job satisfaction. *International journal of hospitality management*. 2022 May 1;103:103211. <https://doi.org/10.1016/j.ijhm.2022.103211>

Reddy S. Generative AI in healthcare: an implementation science informed translational path on application, integration and governance. *Implementation Science*. 2024 Mar 15;19(1):27.

Grządzielewska M. Using machine learning in burnout prediction: A survey. *Child and Adolescent Social Work Journal*. 2021 Apr;38(2):175-80.

Kresge L. Data and algorithms in the workplace: a primer on new technologies.

Zaza S, Riemenschneider C, Armstrong DJ. The drivers and effects of burnout within an information technology work context: a job demands-resources framework. *Information Technology & People*. 2022 Dec 7;35(7):2288-313. DOI: 10.1108/ITP-01-2021-0093

Djokic A. IMPACT OF ORGANIZATIONAL CLIMATE ON EMPLOYEE BURNOUT IN AN INDUSTRIAL COMPANY. In *DIEM: Dubrovnik International Economic Meeting 2024* (Vol. 9, No. 1, pp. 0-0). Sveučilište u Dubrovniku. DOI: 10.17818/DIEM/2024/1.1

Ajayi FA, Udeh CA. Combating Burnout in the it industry: a review of employee well-being initiatives. *International Journal of Applied Research in Social Sciences*. 2024 Apr 10;6(4):567-88.

Durodolu OO, Marutha NS. Job Burnout and Performance as a Determinant of Organisational Loyalty of Records Professionals. Mousaion: *South African Journal of Information Studies*. 2024 May 27:19-pages.

Kuadey NA, Ankora C, Tahiru F, Bensah L, Agbesi CC, Bolatimi SO. Using machine learning algorithms to examine the impact of technostress creators on student learning burnout and perceived academic performance. *International Journal of Information Technology*. 2024 Apr;16(4):2467-82.

Wu L. Designing for healthy self-reflection on cognitive data in the context of burnout.

Liu Q, Ma Y, Chen L, Pedrycz W, Skibniewski MJ, Chen ZS. Artificial intelligence for production, operations and logistics management in modular construction industry: A systematic literature review. *Information Fusion*. 2024 Apr 12:102423. DOI: 10.1016/j.inffus.2024.102423

Ozakca NS, Bulus A, Cetin A. Artificial Intelligence Based Employee Attrition Analysis and Prediction. In 2024 6th International Conference on Computing and Informatics (ICCI) 2024 Mar 6 (pp. 512-517). IEEE. DOI: 10.1109/ICCI61671.2024.10485157

Luz A, Olaoye G. Leveraging Data for Actionable Insights and Engagement Strategies. *EasyChair*; 2024 May 7.

Algabri HK. *Artificial Intelligence and ChatGPT*. Academic Guru Publishing House; 2024 Feb 14.

Koutsimani, P.; Montgomery, A.; Masoura, E.; Panagopoulou, E. Burnout and Cognitive Performance. *Int. J. Environ. Res. Public Health* 2021, 18, 2145.  
<https://doi.org/10.3390/ijerph18042145>

## Appendices

Proof of concepts and minimum viable product

### Fields in Employee Documents:

**employee\_id:** Unique identifier for each employee.

**first\_name:** First name of the employee.

**last\_name:** Last name of the employee.

**job\_title:** Job title of the employee.

**reviews:** An object containing various aspects of employee reviews, such as `management_review`, `self_survey`, `peer_feedback`, `development_programs`, and `absence`. Each aspect is stored as a key-value pair within the reviews object.

Embeddings: Embeddings generated from the employee's first name, last name, and job title using the SentenceTransformer model. These embeddings are stored as a list.

**(Please create your own Mongo DB Database Atlas Free account and use your user id and password.)**

**If you need my password and User ID for MongoDB Atlas, please let me know I can provide them on request.**

### **Connection URI for MongoDB Atlas:**

```
mongodb+srv://yourusername:yourpassword@cluster0.ivvkqso.mongodb.net/?retryWrites=true
&w=majority&appName=Cluster0
```

Database Name: "employee\_management\_DB"

Collection Name: "employees"

The fields which we created in the collections are:

```
employee_id = input("Enter employee ID: ")
```

```
first_name = input("Enter first name: ")
```

```
last_name = input("Enter last name: ")
```

```
job_title = input("Enter job title: ")
```

```
reviews = {management_review = input("Enter new management review: ")
```

```
    self_survey = input("Enter new self survey: ")
```

```
    peer_feedback = input("Enter new peer feedback: ")
```

```
    development_programs = input("Enter new development programs: ")
```

```
    absence = input("Enter new absence: ")}
```

**Appendix 1: System Interface – Code - Vectorembdding.py**

```
from sentence_transformers import SentenceTransformer
from pymongo import MongoClient

# Load the pre-trained model
model = SentenceTransformer('sentence-transformers/all-MiniLM-L6-v2')

# Function to connect to MongoDB and retrieve employee documents
def connect_to_mongodb():

    try:

        client = MongoClient("mongodb+srv://yourid:yourpassword@cluster0.ivvkqso.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0")

        db = client["employee_management_DB"]
        collection = db["employees"]
        return collection

    except Exception as e:
        print(f"Error connecting to MongoDB: {e}")
        return None

# Function to retrieve employee details by employee ID
def get_employee_details_by_id(collection, employee_id):

    try:

        employee = collection.find_one({"employee_id": employee_id})
```

```
if employee:
    return employee
else:
    return None
except Exception as e:
    print(f"Error retrieving employee details: {e}")
    return None

# Function to create a new employee
def create_new_employee(collection):
    try:
        employee_id = input("Enter employee ID: ")
        first_name = input("Enter first name: ")
        last_name = input("Enter last name: ")
        job_title = input("Enter job title: ")
        reviews = {}

        # Encode the text fields
        embeddings = model.encode([first_name, last_name, job_title])

        # Create a new employee document with embeddings
        new_employee = {
            "employee_id": employee_id,
            "first_name": first_name,
            "last_name": last_name,
            "job_title": job_title,
            "reviews": reviews,
            "embeddings": embeddings.tolist() # Convert embeddings to list for MongoDB
        }
```

```
}

# Insert the new employee document into the collection
collection.insert_one(new_employee)
print("New employee created successfully.")
except Exception as e:
    print(f"Error creating new employee: {e}")

# Function to update employee review
def update_employee_review(collection, employee_id):
    employee = get_employee_details_by_id(collection, employee_id)
    if employee:

        # Display employee information except for review
        print("Employee Information:")
        print(f"Employee ID: {employee['employee_id']}")
        print(f"First Name: {employee['first_name']}")
        print(f>Last Name: {employee['last_name']}")
        print(f"Job Title: {employee['job_title']}")
        management_review = input("Enter new management review: ")
        self_survey = input("Enter new self survey: ")
        peer_feedback = input("Enter new peer feedback: ")
        development_programs = input("Enter new development programs: ")
        absence = input("Enter new absence: ")

    # Update the employee document
    collection.update_one(
        {"employee_id": employee_id},
```

```

    {"$set": {
        "reviews": {
            "management_review": management_review,
            "self_survey": self_survey,
            "peer_feedback": peer_feedback,
            "development_programs": development_programs,
            "absence": absence
        }
    }}
)
print("Employee review updated successfully.")
else:
    print("Employee not found.")

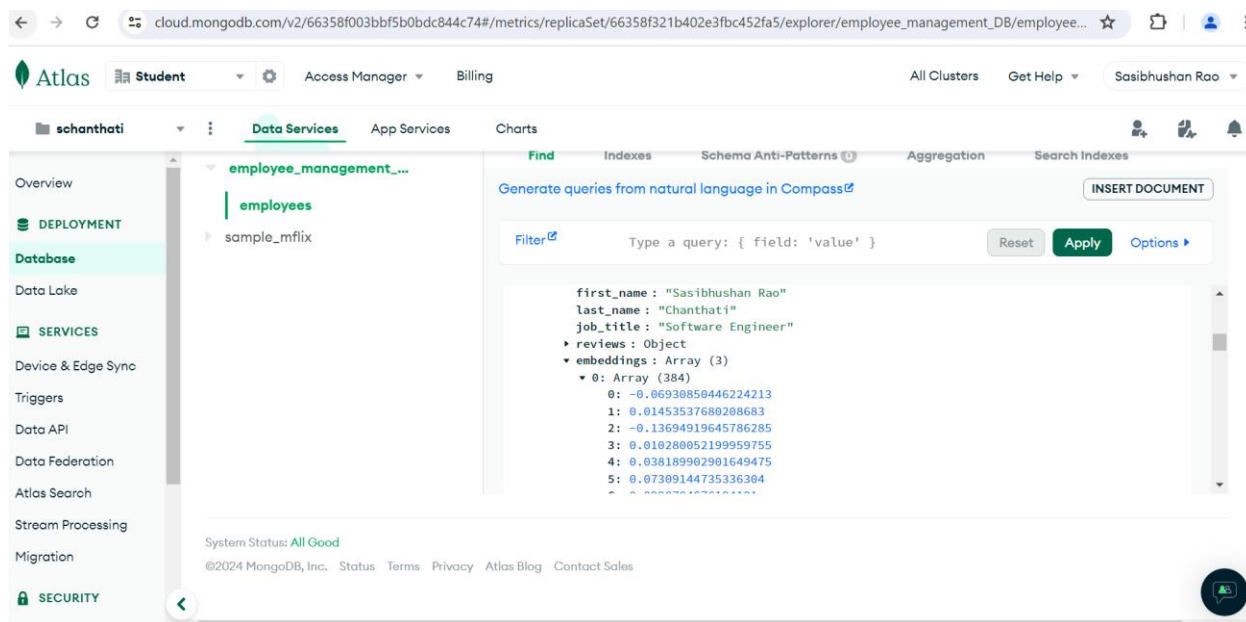
# Main vector search interface function
def vector_search_interface():
    print("Hi! How can I help you today?")
    collection = connect_to_mongodb()
    if collection is None:
        return
    while True:
        print("\n1. Display Employee Information")
        print("2. Create New Employee")
        print("3. Update Employee Review")
        print("4. Exit")
        choice = input("Enter your choice: ")
        if choice == "1":
            employee_id = input("Enter the employee ID: ")

```

```
employee = get_employee_details_by_id(collection, employee_id)
if employee:
    print("Employee Information:")
    print(f"Employee ID: {employee['employee_id']}")
    print(f"First Name: {employee['first_name']}")
    print(f>Last Name: {employee['last_name']}")
    print(f"Job Title: {employee['job_title']}")
    if 'reviews' in employee:
        print("Reviews:")
        for key, value in employee['reviews'].items():
            print(f"- {key}: {value}")
    else:
        print("No reviews available.")
    print("-----")
else:
    print("Employee not found.")
elif choice == "2":
    create_new_employee(collection)
    elif choice == "3":
        employee_id = input("Enter the employee ID: ")
        update_employee_review(collection, employee_id)
elif choice == "4":
    print("Exiting...")
    break
    else:
        print("Invalid choice. Please try again.")
if __name__ == "__main__":
```



vector\_search\_interface()



## Appendix 2: Execution of Code - Vectorsearch.py

```

from sentence_transformers import SentenceTransformer, util
from pymongo import MongoClient

# Load the pre-trained model
model = SentenceTransformer('sentence-transformers/all-MiniLM-L6-v2')

# Function to connect to MongoDB and retrieve employee documents
def connect_to_mongodb():
    try:

```

```

client =
MongoClient("mongodb+srv://yourid:yourpassword@cluster0.ivvkqso.mongodb.net/?retryWrite
s=true&w=majority&appName=Cluster0")
    db = client["employee_management_DB"]
    collection = db["employees"]
    return collection
except Exception as e:
    print(f"Error connecting to MongoDB: {e}")
    return None

# Function to retrieve all software engineers
def get_software_engineers(collection):
    try:
        software_engineers = collection.find({"job_title": "Software Engineer"})
        return list(software_engineers)
    except Exception as e:
        print(f"Error retrieving software engineers: {e}")
        return None

# Function to count people who know machine learning and AI from reviews
def count_ml_ai_experts(collection):
    try:
        ml_ai_count = 0
        employees = collection.find({})
        for employee in employees:
            reviews = employee.get("reviews", {})
            for review in reviews.values():
                if "machine learning" in review.lower() or "artificial intelligence" in review.lower():

```

```
        ml_ai_count += 1
        break # Assuming one mention per review is sufficient
    return ml_ai_count
except Exception as e:
    print(f"Error counting ML/AI experts: {e}")
    return None

# Function to find the employee with the best performance review
def find_best_performance(collection):
    try:
        best_employee = collection.find_one(sort=[("reviews.performance_review", -1)])
        return best_employee
    except Exception as e:
        print(f"Error finding best performance: {e}")
        return None

# Function to check if development program includes AI/ML
def check_development_program_ai_ml(collection):
    try:
        ai_ml_programs = collection.find({"reviews.development_programs": {"$regex":
"AI|ML|Machine Learning|Artificial Intelligence"}})
        return ai_ml_programs.count() > 0
    except Exception as e:
        print(f"Error checking development programs: {e}")
        return None

# Function to check if self survey includes AI/ML certifications
def check_self_survey_ai_ml_certifications(collection):
```

```

try:
    ai_ml_certifications = collection.find({"reviews.self_survey": {"$regex": "AI|ML|Machine
Learning|Artificial Intelligence|Certification"}})
    return ai_ml_certifications.count() > 0
except Exception as e:
    print(f"Error checking self survey certifications: {e}")
    return None

# Function to check if anyone has excellent management review and mentions AI/ML
def check_excellent_management_review_ai_ml(collection):
    try:
        excellent_employees = collection.find({"$and": [{"reviews.management_review":
"excellent"},
{"$or": [{"reviews.management_review": {"$regex":
"AI|ML|Machine Learning|Artificial Intelligence"}},
{"reviews.self_survey": {"$regex": "AI|ML|Machine
Learning|Artificial Intelligence"}]}]}])
        return excellent_employees.count() > 0
    except Exception as e:
        print(f"Error checking excellent management review and AI/ML mentions: {e}")
        return None

# Placeholder function to retrieve project managers
def get_project_managers(collection):
    # Placeholder implementation
    print("Placeholder: Function 'get_project_managers' is not implemented yet.")
    return None

```

```
# Placeholder function to count Python proficiency
def count_python_proficiency(collection):
    # Placeholder implementation
    print("Placeholder: Function 'count_python_proficiency' is not implemented yet.")
    return None

# Placeholder function to find employee with highest number of absences
def find_highest_absence(collection):
    # Placeholder implementation
    print("Placeholder: Function 'find_highest_absence' is not implemented yet.")
    return None

# Main HR search interface function
def hr_search_interface():
    collection = connect_to_mongodb()
    if collection is None:
        return
    while True:
        question = input("Ask your HR-related question: ")

        if "software engineers" in question.lower():
            software_engineers = get_software_engineers(collection)
            if software_engineers:
                print("List of Software Engineers:")
                for engineer in software_engineers:
                    print(f"- {engineer['first_name']} {engineer['last_name']}")
            else:
                print("No software engineers found.")
```

```
elif "machine learning and ai" in question.lower():
    ml_ai_count = count_ml_ai_experts(collection)
    if ml_ai_count is not None:
        print(f"Count of people who know Machine Learning and AI: {ml_ai_count}")
    else:
        print("Error counting ML/AI experts.")
elif "best performance review" in question.lower():
    best_employee = find_best_performance(collection)
    if best_employee:
        print("Employee with the best performance review:")
        print(f"- {best_employee['first_name']} {best_employee['last_name']}")
    else:
        print("No employee found with the best performance review.")
elif "development program" in question.lower() and ("ai" in question.lower() or "ml" in
question.lower() or "machine learning" in question.lower() or "artificial intelligence" in
question.lower()):
    if check_development_program_ai_ml(collection):
        print("Development program includes AI/ML.")
    else:
        print("Development program does not include AI/ML.")
elif "self survey" in question.lower() and ("ai" in question.lower() or "ml" in question.lower()
or "machine learning" in question.lower() or "artificial intelligence" in question.lower()) and
"certifications" in question.lower():
    if check_self_survey_ai_ml_certifications(collection):
        print("Self survey includes AI/ML certifications.")
    else:
        print("Self survey does not include AI/ML certifications.")
```

```
elif "excellent management review" in question.lower() and ("ai" in question.lower() or "ml"
in question.lower() or "machine learning" in question.lower() or "artificial intelligence" in
question.lower()):
    if check_excellent_management_review_ai_ml(collection):
        print("There are employees with excellent management review and AI/ML mentions.")
    else:
        print("No employees found with excellent management review and AI/ML mentions.")
# Additional questions and answers
elif "project managers" in question.lower():
    project_managers = get_project_managers(collection)
    if project_managers:
        print("List of Project Managers:")
        for manager in project_managers:
            print(f"- {manager['first_name']} {manager['last_name']}")
    else:
        print("No project managers found.")
elif "python proficiency" in question.lower():
    python_proficiency_count = count_python_proficiency(collection)
    if python_proficiency_count is not None:
        print(f"Count of people proficient in Python: {python_proficiency_count}")
    else:
        print("Error counting Python proficient employees.")

elif "highest number of absences" in question.lower():
    highest_absence_employee = find_highest_absence(collection)
    if highest_absence_employee:
        print("Employee with the highest number of absences:")
```

```

                print(f"- {highest_absence_employee['first_name']}
{highest_absence_employee['last_name']}")
        else:
            print("No employee found with the highest number of absences.")
        # Add more questions and corresponding functions as needed
    else:
        print("Sorry, I couldn't understand your question. Please ask again or type 'exit' to quit.")
    if question.lower() == "exit":
        print("Exiting...")
        break
if __name__ == "__main__":
    hr_search_interface()

```

```

from datetime import datetime, timedelta
from sentence_transformers import SentenceTransformer
from pymongo import MongoClient
# Load the pre-trained model
model = SentenceTransformer('sentence-transformers/all-MiniLM-L6-v2')
# Function to connect to MongoDB and retrieve employee documents
def connect_to_mongodb():
    try:
        client = MongoClient("mongodb+srv://yourid:yourpassword@cluster0.ivvkqso.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0")
        db = client["employee_management_DB"]
        collection = db["employees"]
        return collection
    except Exception as e:

```



```

print(f"Error connecting to MongoDB: {e}")
return None

# Function to retrieve recently promoted employees based on mentions in reviews
def get_recently_promoted_employees(collection):
    try:
        six_months_ago = datetime.now() - timedelta(days=30 * 6)
        recently_promoted_employees = collection.find({
            "reviews": {
                "$elemMatch": {
                    "$or": [
                        {"text": {"$regex": "promoted", "$options": "i"}}, # Look for mentions of
"promoted" in reviews
                        {"text": {"$regex": "new job title", "$options": "i"}} # Look for mentions of a new
job title
                    ],
                    "date": {"$gte": six_months_ago}
                }
            }
        })
        return list(recently_promoted_employees)
    except Exception as e:
        print(f"Error retrieving recently promoted employees: {e}")
        return None

# Main HR search interface function
def hr_search_interface():
    collection = connect_to_mongodb()
    if collection is None:

```

```
return
while True:
    question = input("Ask your HR-related question: ")

    if "promoted in the last six months" in question.lower():
        recently_promoted_employees = get_recently_promoted_employees(collection)
        if recently_promoted_employees:
            print("Employees recently promoted:")
            for employee in recently_promoted_employees:
                print(f"- {employee['first_name']} {employee['last_name']}")
        else:
            print("No recently promoted employees found.")

    # Add other HR-related questions and corresponding functionalities here

    elif question.lower() == "exit":
        print("Exiting...")
        break
    else:
        print("Sorry, I couldn't understand your question. Please ask again or type 'exit' to quit.")
if __name__ == "__main__":
    hr_search_interface()
```

-----END OF THE DOCUMENT-----