EMPROVE - AN EMPLOYEE PRODUCTIVITY SYSTEM USING MACHINE LEARNING

A PROJECT REPORT

by

ALEN GEORGE (VJC19CS018)
JOEL RAJU (VJC19CS079)
ROSHAN ROY (VJC19CS106)
SAN BABY FRANCIS (VJC19CS108)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VISWAJYOTHI COLLEGE OF ENGINEERING AND TECHNOLOGY, VAZHAKULAM

MAY 2023

EMPROVE - AN EMPLOYEE PRODUCTIVITY SYSTEM USING MACHINE LEARNING

A PROJECT REPORT

by

ALEN GEORGE (VJC19CS018)
JOEL RAJU (VJC19CS079)
ROSHAN ROY (VJC19CS106)
SAN BABY FRANCIS (VJC19CS108)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

under the guidance

of

Mrs. Anju Markose

Assistant Professor, CSE Dept.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISWAJYOTHI COLLEGE OF ENGINEERING AND TECHNOLOGY, VAZHAKULAM

MAY 2023

VISWAJYOTHI COLLEGE OF ENGINEERING AND TECHNOLOGY, VAZHAKULAM

Department of Computer Science and Engineering

Vision

Moulding socially responsible and professionally competent Computer Engineers to adapt to the dynamic technological landscape

Mission

1. Foster the principles and practices of computer science to empower life-long learning and build careers in software and hardware development.

2. Impart value education to elevate students to be successful, ethical and effective problem-solvers to serve the needs of the industry, government, society and the scientific community.

3. Promote industry interaction to pursue new technologies in Computer Science and provide excellent infrastructure to engage faculty and students in scholarly research activities.

Program Educational Objectives

Our Graduates

1. Shall have creative aid critical reasoning skills to solve technical problems ethically and responsibly to serve the society.

2. Shall have competency to collaborate as a team member and team leader to address social, technical and engineering challenges.

3. Shall have ability to contribute to the development of the next generation of information technology either through innovative research or through practice in a corporate setting

4. Shall have potential to build start-up companies with the foundations, knowledge and experience they acquired from undergraduate education

Program Outcomes

1. **Engineering knowledge**:Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis**:Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences

3. **Design / development of solutions**:Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety and the cultural, societal and environmental considerations.

4. **Conduct investigations of complex problems**:Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data and synthesis of the information to provide valid conclusions.

5. **Modern tool usage**:Create, select and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society:**Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability:**Understand the impact of the professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of and need for sustainable development.

8. **Ethics:**Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice

9. **Individual and team work:**Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary settings

10. **Communication:**Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations and give and receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and unread in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**:Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes

1. Ability to integrate theory and practice to construct software systems of varying complexity
2. Able to Apply Computer Science skills, tools and mathematical techniques to analyse, design and model complex systems
3. Ability to design and manage small-scale projects to develop a career in a related industry.

VISWAJYOTHI COLLEGE OF ENGINEERING AND TECHNOLOGY, VAZHAKULAM

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that project work entitled **" EMPROVE - AN EMPLOYEE PRODUCTIVITY SYSTEM USING MACHINE LEARNING "** is a bonafide work done by **ALEN GEORGE (VJC19CS018), JOEL RAJU (VJC19CS079), ROSHAN ROY (VJC19CS106), SAN BABY FRANCIS (VJC19CS108)** in partial fulfillment of the award of the Degree of **Bachelor of Technology in Computer Science & Engineering** from APJ Abdul Kalam Technological University, Thiruvananthapuram, Kerala during the academic year 2022-2023

Internal Supervisor **External Supervisor**

Project Coordinator **Head of Department**

ACKNOWLEDGEMENT

First and foremost, we thank **God Almighty** for His divine grace and blessings in making all these possible. May He continue to lead us in the years. It is our privilege to render our heartfelt thanks to our most beloved Manager **Msg. Dr. Pius Malekandathil** , our director **Rev. Fr. Paul Nedumpurath** and our Principal **Dr. K K Rajan** for providing us the opportunity to do this project during the Fourth year (2023) of our B.Tech degree course. We are deeply thankful to our Head of the Department, **Mr. Amel Austine** for his support and encouragement. We would like to express our sincere gratitude and heartfelt thanks to our Project Guide **Mrs. Anju Markose**, Asst. Professor, Department of Computer Science and Engineering for her motivation, assistance and help for the project. We also express sincere thanks to our Project Coordinator **Mr. Joe Mathew Jacob**, Asst. Professor, Department of Computer Science and Engineering for his guidance and support. We also thank all the staff members of the Computer Science Department for providing their assistance and support. Last, but not the least, we thank all our friends and family for their valuable feedback from time to time as well as their help and encouragement.

DECLARATION

We undersigned hereby declare that the project report "EMPROVE - AN EMPLOYEE PRODUCTIVITY SYSTEM USING MACHINE LEARNING", submitted for partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology of the APJ Abdul Kalam Technological University is a bonafide work done by us under the supervision of Mrs. Anju Markose. This submission represents ideas in our own words and where ideas or words of others have been included, We have adequately and accurately cited and referenced the original sources. We also declare that We have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or formed the basis for the award of any degree, diploma or similar title of any other University.

Place : Vazhakulam

Date : .....

Alen George

Joel Raju

Roshan Roy

San Baby Francis

ABSTRACT

Efficient employee productivity has become a crucial factor for organizations to maximize profits and ensure employee satisfaction. While managerial decisions have traditionally been relied upon to achieve this, the ever-evolving software industry and rise of remote work necessitate an alternative approach. This paper proposes a novel software system, Emprove, designed to increase employee productivity by providing timely short breaks using the Pomodoro technique, allowing for the setting of concentration or focus music of choice and enabling the setting and tracking of tasks and deadlines. Emprove also utilizes machine learning techniques to detect employee alertness levels in real-time, calculating Mouth Aspect Ratio (MAR) and Eye Aspect Ratio (EAR) from facial keypoints generated by the MediaPipe Face Mesh model. Additionally, employees can monitor their work stress levels using a metric jointly proposed by the American Institute of Stress and the Marlin Company. Furthermore, detailed reports can be generated for both the employee and their manager to view, providing actionable insights to increase productivity. Ultimately, Emprove provides a comprehensive approach to ensure employee productivity and promote optimal performance in the workplace.

Key Words :– productivity, software engineering, stress management, machine learning, image processing

Contents

List of Figures

List of Tables

List of Abbreviations

EAR	Eye Aspect Ratio
MAR	Mouth Aspect Ratio
CNN	Convolution Neural Network
EEG	Electroencephalography
EOG	Electroocoulogram
EJS	Embedded Javascript
MAD	Mean Absolute Distance
IOD	Interocular Distance
GPU	Graphical Processing Unit
CPU	Central Processing Unit
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
AR	Augmented Reality

Chapter 1

INTRODUCTION

In today's fast-paced and rapidly-evolving software industry, employee productivity has become a key factor in determining the success of any organization. To address this need for a modern and reliable employee productivity system, we propose "Emprove," a web-based system designed to help employees manage their tasks, schedules and workload more efficiently. Emprove is an intuitive and feature-rich productivity management system that aims to increase employee productivity and job satisfaction while reducing stress. The system includes a variety of features, such as task management, a Pomodoro timer, concentration music, drowsiness detection, work stress assessment and productivity reports. These features are all geared towards promoting a productive work environment, allowing employees to set short-term goals and take timely short breaks during work hours. To ensure employee focus and productivity, Emprove follows the Pomodoro technique, a time-management method that involves breaking work into focused intervals with frequent short breaks. One of the unique features of Emprove is its real-time detection of facial features to assess the employee's level of alertness during work hours. This feature enables the employees to identify patterns and take appropriate measures to improve productivity. Emprove also generates a detailed report that highlights the employee's productivity level, time management skills and areas that need improvement. Emprove also allows users to choose a concentration music of their choice to enhance their focus and create a positive work environment. This is a proven method to enhance productivity and attention of the employees. The Work Stress Assessment designed by the American Institute of Stress and the Marlin Company would help the employees to assess the stress they face in their workplace. The scale measures physical and emotional well-being, workload, communication, utilization of skills, recognition, job pressure, interference with personal life and control over work duties. Tips are provided for reducing stress levels and professional assistance is suggested if health is significantly affected. The system also includes a manager dashboard that enables managers to efficiently manage their employees, view their tasks and progress and assign tasks and deadlines. Emprove is thus a comprehensive employee productivity system designed to increase employee productivity, job satisfaction and well-being while reducing stress. Its features

are geared towards promoting a positive and productive work environment and helping employees manage their time and workload more efficiently. With Emprove, organizations can effectively manage their employees and maximize their productivity.

1.1 Problem Definition

The problem at hand is the lack of a modern and reliable system to ensure employee productivity in the software industry, which is experiencing constant evolution and a shift towards remote work for many companies. This problem arises from the growing need to effectively monitor and manage the productivity of employees in a remote work environment, where traditional methods may be insufficient or impractical. The absence of such a system hinders companies' ability to maximize the productivity of their remote employees, potentially leading to decreased efficiency, missed deadlines and a lack of accountability. Therefore, there is a pressing need to develop a solution that addresses these challenges and provides an effective means of ensuring and enhancing employee productivity in the evolving software industry.

1.2 Objective

The objective of the software system is to create a comprehensive solution that maximizes employee productivity in the software industry. By incorporating features such as a Pomodoro Timer, Concentration Music and Drowsiness Detection, the system aims to provide employees and managers with tools and resources to work more efficiently and effectively. The Pomodoro Timer feature helps employees manage their time and increase focus by breaking work into intervals, typically 25 minutes of focused work followed by a short break. This technique enhances productivity by promoting task concentration and reducing the likelihood of burnout or mental fatigue. The Concentration Music feature aims to create a conducive work environment by providing background music specifically designed to enhance focus and concentration. Research has shown that certain types of music can improve cognitive performance and reduce distractions, leading to improved productivity and creativity. The Drowsiness Detection feature addresses the issue of employee fatigue and drowsiness, particularly relevant in remote work settings where physical presence may not be monitored. By leveraging technologies such as computer vision or biometric sensors, the system can detect signs of drowsiness and alert the employee to take necessary breaks or rest. This helps prevent productivity loss due to reduced alertness and potentially mitigates safety risks. Overall, the system's objective is to support both employees and managers in optimizing their work processes, fostering a more productive and satisfying work environment. By providing tools for time management, focus enhancement and addressing employee fatigue, the

system aims to increase productivity levels, improve task completion rates and enhance job satisfaction. The end result is a more efficient and effective workforce that contributes to the success and growth of software companies.

1.3 Scope

Our system is an employee productivity system that maximizes the productivity of employees by setting short-term goals and providing timely short breaks during working hours. It follows the Pomodoro technique, a time-management method that involves breaking work into focused intervals with frequent short breaks. This technique ensures that the employee takes timely breaks during work to help maintain focus and productivity. It includes a feature that detects the level of alertness of the employee during working hours using real-time detection of facial features. This feature helps the employees and managers to identify patterns and take appropriate measures to improve their productivity. The system generates a detailed report that highlights the employee's productivity level, time management skills and areas that need improvement. It also allows users to set a concentration music of their choice. This feature aims to enhance employee focus and create a positive work environment. Additionally, the system provides a manager dashboard that enables managers to efficiently manage his employees.

Chapter 2

LITERATURE SURVEY

2.1 Real-time Facial Surface Geometry from Monocular Video on Mobile GPUs

This paper proposes an end-to-end neural network-based model for inferring an approximate 3D mesh representation of a human face from single camera input for Augmented Reality (AR) applications. The model uses a custom but fairly straightforward residual neural network architecture and predicts facial landmarks as a separate output at the end of a dedicated network branch. The relatively dense mesh model of 468 vertices is well-suited for face-based AR effects. The paper discusses the problem of predicting facial geometry by aligning a facial mesh template, also called face alignment or face registration and evaluates the proposed model on a geographically diverse evaluation set of 1.7K images.

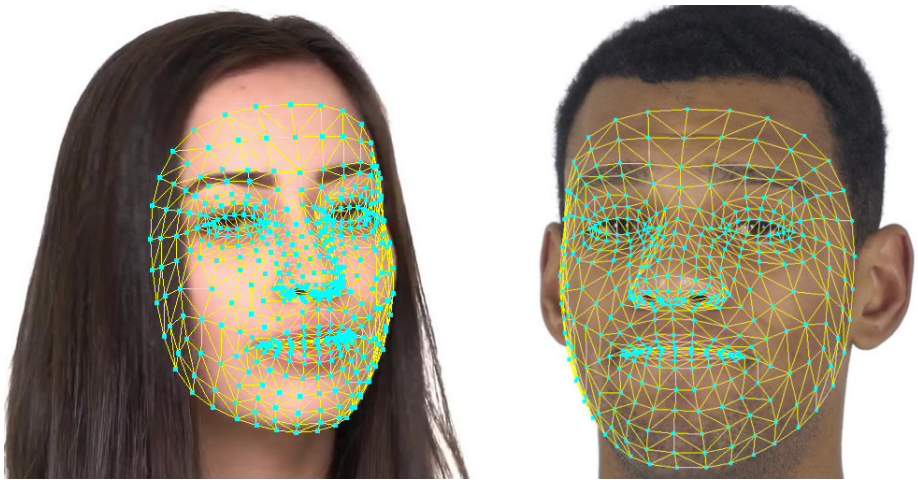

Figure 2.1: Face Mesh Prediction Examples

The authors use Mean Absolute Distance (MAD) between the predictions and the ground truth vertex locations, normalized by Interocular Distance (IOD), to measure the prediction quality. The proposed model demonstrates super-realtime inference speed on mobile GPUs (100–1000+ FPS, depending on the device and model variant) and a high prediction quality that is comparable to the variance in manual annotations of the same image. The authors address the issue of temporal

jitter in the trajectories of individual landmarks by employing a one-dimensional temporal filter applied independently to each predicted landmark coordinate. The authors also propose to use 2D landmarks corresponding to a small subset of the mesh vertices participating in a set of semantic contours annotated over the actual ”in-the-wild” dataset as supervision for training an initial model. Fast annotation refinement is enabled by a ”brush” instrument with adjustable radius that lets a whole range of points to be moved at once. The authors have designed lighter versions of the model to address CPU inference on the mobile devices lacking proper GPU support.

2.1.1 Advantages

- End-to-end approach: The model provides a complete solution for inferring a 3D mesh representation of a human face from a single camera input, simplifying the development process for AR applications.
- Custom residual neural network architecture: The model utilizes a tailored architecture that is straightforward to understand and implement, allowing for easier adoption and customization.
- Predicts 468 facial landmarks: In addition to the 3D mesh representation, the model also predicts 468 facial landmarks, which can be useful for various tasks such as facial animation and tracking.

2.1.2 Disadvantages

- Limited evaluation dataset: The evaluation set used in the study consists of 1.7K images, which may not cover the full range of potential scenarios and variations encountered in real-world applications.
- Dependency on GPU support: The model’s real-time performance is highly dependent on the availability of proper GPU support. This may limit its usage on mobile devices lacking adequate GPU capabilities.
- Temporal jitter in landmark trajectories: The authors address the issue of temporal jitter in landmark trajectories through the use of a one-dimensional temporal filter. However, the effectiveness of this approach in handling all types of temporal variations is not thoroughly discussed.

2.2 Driver Drowsiness Detection Using Machine Learning Algorithm

The paper discusses the problem of driver drowsiness, which is a major cause of road accidents. Several techniques which help to detect drowsiness or lack of attention have been developed recently. Many such techniques are used to detect driver drowsiness as it pose a serious threat of road accidents. The paper proposes a system for driver drowsiness detection using a machine learning algorithm. To stop accidents brought on by driver inattention, a variety of techniques have been employed, including EEG, EOG, image processing, physiological and visual signal-based techniques and simulator-based detection techniques. The paper discusses the limitation of these techniques including the need for costly equipment and discomfort of drivers. The proposed system involves capturing real-time video of the driver's face using a webcam and detecting certain facial features such as the eyes and mouth to determine their activity. The authors also emphasize the importance of reducing the number of road accidents caused by driver fatigue and drowsiness and propose using automotive systems to monitor signs of fatigue and prevent mishaps.

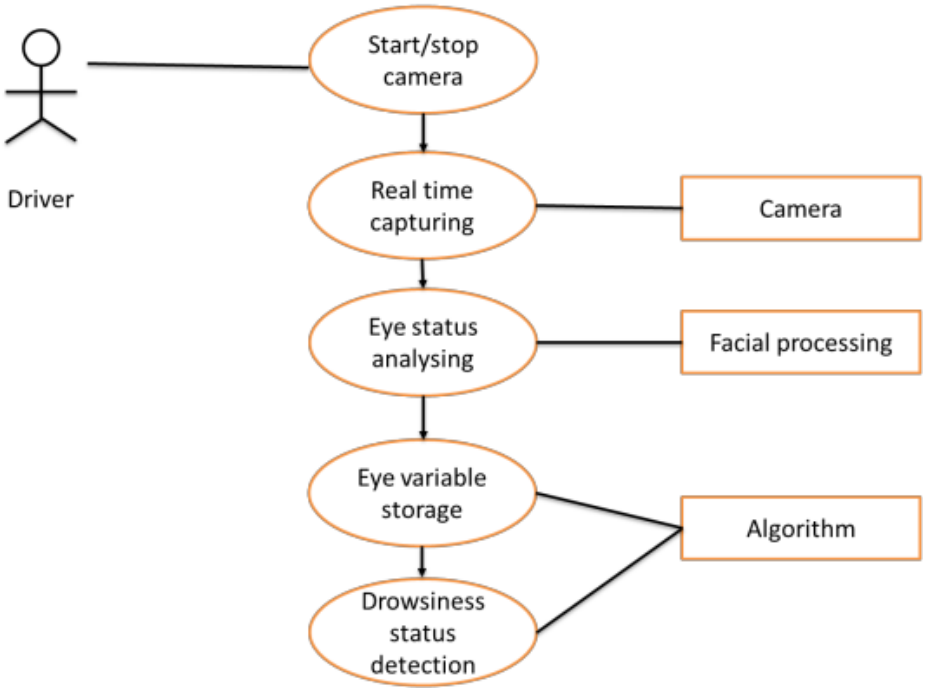

Figure 2.2: System Overview for Drowsiness Detection

The system proposed in the paper utilizes a webcam to capture real-time video of the driver's face and calculates the drowsiness level based on factors like emotional activity and the type of distraction. It maps 68 points that identify the coordinates of the facial structure and uses the Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR) to determine eye and yawn activity of the driver. The system monitors the distraction level of the driver and emotional activity to alert the driver whenever they are feeling drowsy. The system aims to minimize the number of accidents

caused by driver drowsiness and fatigue and the same concept has been applied to the employee productivity system to detect drowsiness and ensure employee productivity.

2.2.1 Advantages

- Addresses a significant problem of driver drowsiness, which is a major cause of road accidents, aiming to improve road safety.
- Utilizes the computation of MAR and EAR in realtime and image processing to detect drowsiness, providing a comprehensive approach to monitoring driver alertness.
- Focuses on real-time detection and alerts, allowing for timely warnings to drivers when they experience drowsiness, potentially preventing accidents.

2.2.2 Disadvantages

- The paper does not mention specific validation studies or accuracy rates for the proposed system, which may raise questions about its reliability.
- Reliance on video-based detection may have limitations in certain conditions, such as low lighting or driver obstruction.
- The complexity of analyzing physiological signals and image processing techniques may require sophisticated algorithms and computational resources.

2.3 The Power of Music - How environmental variables can disrupt or enable productivity

This paper presents a study aimed at investigating the influence of music on productivity during software development. The study examines how different aspects of music, such as tempo and presence of lyrics, affect the efficiency of software developers. Data was collected through questionnaires, interviews and observational studies. The findings indicate that music can enhance productivity when used appropriately and specific types of music can have a stimulating effect in the right circumstances. However, the study acknowledges limitations regarding the study's duration, the number of participants and the lack of consideration for vocal components and varying tempos in music. To assess the impact of music on software development productivity, a multi-method approach was adopted. The study incorporated questionnaires, interviews and observational studies to gather data. The questionnaires were designed to capture participants' music-listening habits, preferences and perceived impact on their work. Interviews provided deeper insights into individual experiences and subjective interpretations. Observational studies involved directly observing

software developers while they worked, noting any changes in their behavior or productivity in the presence of music. The study findings revealed that music can act as a facilitator of productivity if utilized appropriately. When software developers listened to music that aligned with their personal preferences and working styles, they reported improved focus, motivation and creativity. The presence of music created a conducive environment for concentration, leading to enhanced problem-solving capabilities. However, it was noted that the impact of music on productivity varied depending on individual preferences, task complexity and the specific characteristics of the music being played.

While the study demonstrated positive effects of music on software development productivity, several limitations were identified. The duration of the study was relatively short, which may have limited the opportunity to capture long-term effects or assess changes over time. Additionally, the study involved a limited number of participants, potentially affecting the generalizability of the findings. Furthermore, the study did not consider the influence of vocal components or music with varying tempos, which could have different effects on productivity. The findings of this study contribute to the growing body of research on the relationship between music and productivity in various domains. Understanding how music influences cognitive processes during software development can inform the design of work environments and strategies that maximize productivity. Further investigation into the specific characteristics of music that promote efficiency will provide valuable insights for both individuals and organizations seeking to optimize performance in software development and other knowledge-intensive tasks.

2.3.1 Advantages

- Comprehensive exploration: The paper aims to explore the effects of music on productivity in software development by considering several factors such as tempo and presence of lyrics. It attempts to provide a more nuanced understanding of the topic by examining various variables.
- Triangulation of data: The authors gather data through triangulation using questionnaires, interviews and observational studies. This approach allows for a more comprehensive analysis of the topic and strengthens the validity of the findings.
- Addressing previous limitations: The paper acknowledges the scattered and inconclusive results of previous studies and the use of overly simplified variables. By addressing these limitations and incorporating a wider range of variables, the authors aim to provide more robust and applicable insights.

2.3.2 Disadvantages

- Lack of specific methodology details: The paper does not provide specific details about the methodology employed in the study. It would be beneficial to know more about the sample size, selection process and data collection methods to assess the reliability and generalizability of the findings.
- Limited focus on software development: Although the paper mentions the focus on software development, it does not provide specific reasons or justifications for choosing this particular field. Further elaboration on the relevance and significance of studying music's impact on productivity specifically in software development would strengthen the paper.
- Limited generalizability: The paper does not mention the scope of the study, such as the specific demographics or context in which the research was conducted. This lack of information limits the generalizability of the findings and their applicability to different settings or populations.

2.4 Framework for Preventing Procrastination and Increasing Productivity

This paper discusses the use of gamification as a market strategy to track, control and reward consumer behavior. It explains that gamification involves the use of game design features in non-game contexts to exploit the motivational power of video games, leading to beneficial impacts on people. The Pomodoro Technique is introduced as a time management method that allows individuals to work for, rather than against, the time they have, helping to increase focus, consciousness and independence of mind. The technique involves breaking up a large task into sessions of 25 minutes followed by a 5-minute break, with longer breaks of 15 to 20 minutes after every four sessions. The document proposes a framework called ProScore that combines the Pomodoro timer, peer-influence, incentivization and gamification to avoid or stop procrastination in employees. The framework uses game elements such as user avatar, leaderboard, points, badges and process bar to provide engaging opportunities and improve behavioral outcomes. It aims to help optimize effort estimation mechanisms by requiring constant reflection of team activities and to increase engagement and productivity through real-time reviews and comparison with other members of the group. Finally, the paper argues that ProScore, being a simple mobile-based application, can play a major role in reducing procrastination even in small fields.

2.4.1 Pomodoro Technique in ProScore

Each job will have its own work time in the ProScore application and the entire slot will be split into a number of pomodoros, each of which will specify a timeslot and unit of work. We use

a timer to keep track of the pomodoros and for each one, there is a mandatory break period. Rewards will be granted in the form of points based on the completion of pomodoros. For each good completion, points will be awarded and in the event of any deviation, the points will be deducted by 5. As a result, if the user fails, the likelihood of falling through bad points is very high, motivating the user to be more aware of points and, as a result, increasing productivity. The user's task time will be taken as input and each task time will be split into pomodoros, with the following calculation:

$$N = t/120 \tag{2.1}$$

$$T = t \ mod \ 120 \tag{2.2}$$

Where, 'N' represents the number of cycles, 'T' represents the total time and 't' represents the input time. The explanation for the division and modulo by 120 is that the traditional pomodoro technique uses a 25-minute pomodoro followed by a 5-minute pause, for a total of four pomodoros in one cycle and a 15-minute break will be provided at the end of each loop . Since not every task in in the programme is supposed to take 120 minutes, the number of cycles and the time are measured and further pomodoro estimates are calculated.

Table 2.1: Pomodoro Time Estimation

Time (T)	Pomodoro(s)	Pomodoro Time (t)	Break Time (b)
0 < T <= 30	1	T	Nil
30 < T<= 60	2	T/2 - 5	5
60 < T<= 90	3	T/3 – 5	5
90<T<= 120	4	T/4 – 5	5

2.4.2 Advantages

- Comprehensive Understanding: The paper provides a comprehensive understanding of gamification and its application as a market strategy to track, control and reward consumer behavior. It explains the concept of gamification and its potential positive impacts on individuals.
- Introduction of Pomodoro Technique: The paper introduces the Pomodoro Technique as a time management method that can increase focus, consciousness and independence of mind. It explains the technique in detail, breaking down large tasks into manageable sessions with short breaks.

- Proposal of ProScore Framework: The paper proposes the ProScore framework, which combines the Pomodoro timer, peer-influence, incentivization and gamification to address procrastination in employees. The framework incorporates various game elements to engage employees, improve behavioral outcomes and increase productivity.

2.4.3 Disadvantages

- Practical Implementation Challenges: The paper does not address potential challenges or barriers to implementing the ProScore framework in real-world settings, such as integration with existing systems, user adoption, or organizational resistance to change. Considering these practical aspects would enhance the feasibility and applicability of the proposed framework.
- Absence of Comparison with Existing Solutions: The paper does not compare the ProScore framework with existing solutions or alternative methods to address procrastination or improve productivity. A comparative analysis would help evaluate the uniqueness and effectiveness of the proposed framework in relation to other approaches.
- Limited Scope: The paper primarily focuses on the application of gamification and the Pomodoro Technique to address procrastination in employees. It does not explore other potential drawbacks or limitations of these strategies or discuss alternative approaches to improve productivity.

2.5 The Effect of Work Environments on Productivity and Satisfaction of Software Engineers

This paper presents a study on the physical work environments of software engineers, which is fundamentally different from other types of office and knowledge work due to its highly collaborative nature and dealing with high complexity. While previous research has focused on physical work environments for office workers in general, there has been little empirical investigation into the types of physical environments that software engineers work in and their impact on productivity and satisfaction. The study was conducted at Microsoft, a large software company and involved a mixed-methods approach with qualitative and quantitative analyses. It explains that software development is highly collaborative and deals with high complexity, both on a social and technical level, requiring a specific investigation of work environments focused on software engineering. Today's software teams are increasingly global and use agile development and social coding tools. The paper presents a mixed-methods study that examines work environment factors, such as personalization, social norms, room composition and atmosphere and investigates the impact of these

factors on perceived productivity and employee satisfaction. Studies show that factors such as personalization, quiet workspaces and the ability to reduce interruptions positively impact productivity for software engineers. The study found that private offices were linked to higher perceived productivity across all disciplines and the ability to work privately with no interruptions and communicate with the team and leads were important factors in satisfaction models.

2.5.1 Advantages

- Unique Focus on Software Engineering: The paper addresses a gap in the existing research by specifically studying the physical work environments of software engineers. It recognizes the unique nature of software engineering work, which is highly collaborative and deals with high complexity and highlights the need for tailored investigations in this field.
- Mixed-Methods Approach: The study utilizes a mixed-methods approach, combining qualitative and quantitative analyses. This approach allows for a more comprehensive understanding of the subject matter by capturing both subjective experiences and objective data, enhancing the validity and reliability of the findings.
- Empirical Investigation: The paper emphasizes the need for empirical investigation by conducting the study at Microsoft, a large software company. By collecting data from real software engineers in a professional setting, the study provides valuable insights into the actual experiences and perceptions of individuals working in this field.
- Absence of controlled experimental design: The paper does not mention the use of controlled experiments or randomized controlled trials (RCTs) to establish causal relationships between work environment factors and productivity/satisfaction. Without a controlled design, it is challenging to determine whether the observed effects are solely due to the physical environment or influenced by other factors.

2.5.2 Disadvantages

- Limited generalizability: The study was conducted at a single company (Microsoft), which may limit the generalizability of the findings. The work environments and practices at Microsoft may not be representative of other software companies or organizations in different industries.
- Lack of diversity: The paper does not explicitly mention the diversity of the sample of software engineers included in the study. If the study lacks diversity in terms of gender, race, or other demographic factors, it may not fully capture the experiences and needs of a diverse range of software engineers.

- Reliance on self-reported data: The study primarily relies on self-reported data, such as surveys and interviews, to assess perceived productivity and satisfaction. Self-reporting can be influenced by various biases, including social desirability bias, which may affect the accuracy and reliability of the results.

Chapter 3

PROPOSED SYSTEM

In today's competitive business world, productivity is the key to success. Companies are constantly looking for ways to improve the efficiency and effectiveness of their employees and new technologies and techniques are being developed to help achieve this goal. Hence, we propose Emprove, an intelligent productivity tracker, which combines insights gained from modern systems and management strategies to provide a comprehensive tool for improving employee productivity.

There are several benefits to using our proposed system. Firstly, by providing regular breaks, employees are able to maintain focus and avoid burnout, which can lead to increased productivity and job satisfaction. Secondly, by using machine learning to detect fatigue and drowsiness, our system is able to detect employee drowsiness which can help to increase the overall efficiency and effectiveness of employees. Concentration music feature will allow the users to choose the track of their choice which would significantly improve one's focus and attention. Work Stress Assessment feature would help the employees to identify their areas of stress and would provide them with a score that indicates their stress level at workplace.

3.1 Architecture Diagram

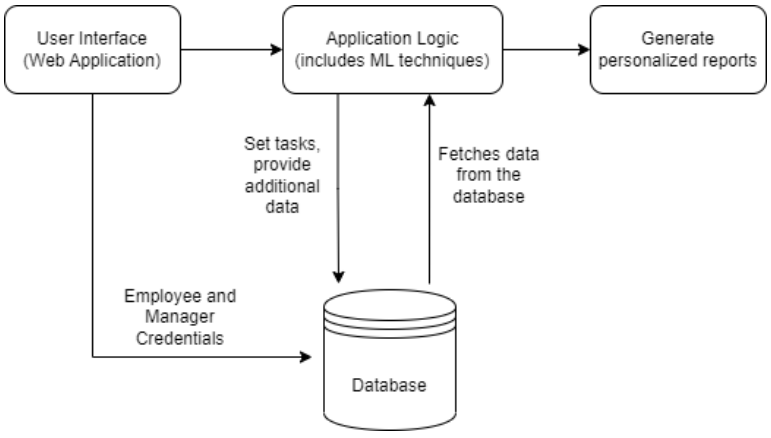

Figure 3.1: Architecture Diagram

3.2 Use Case Diagram

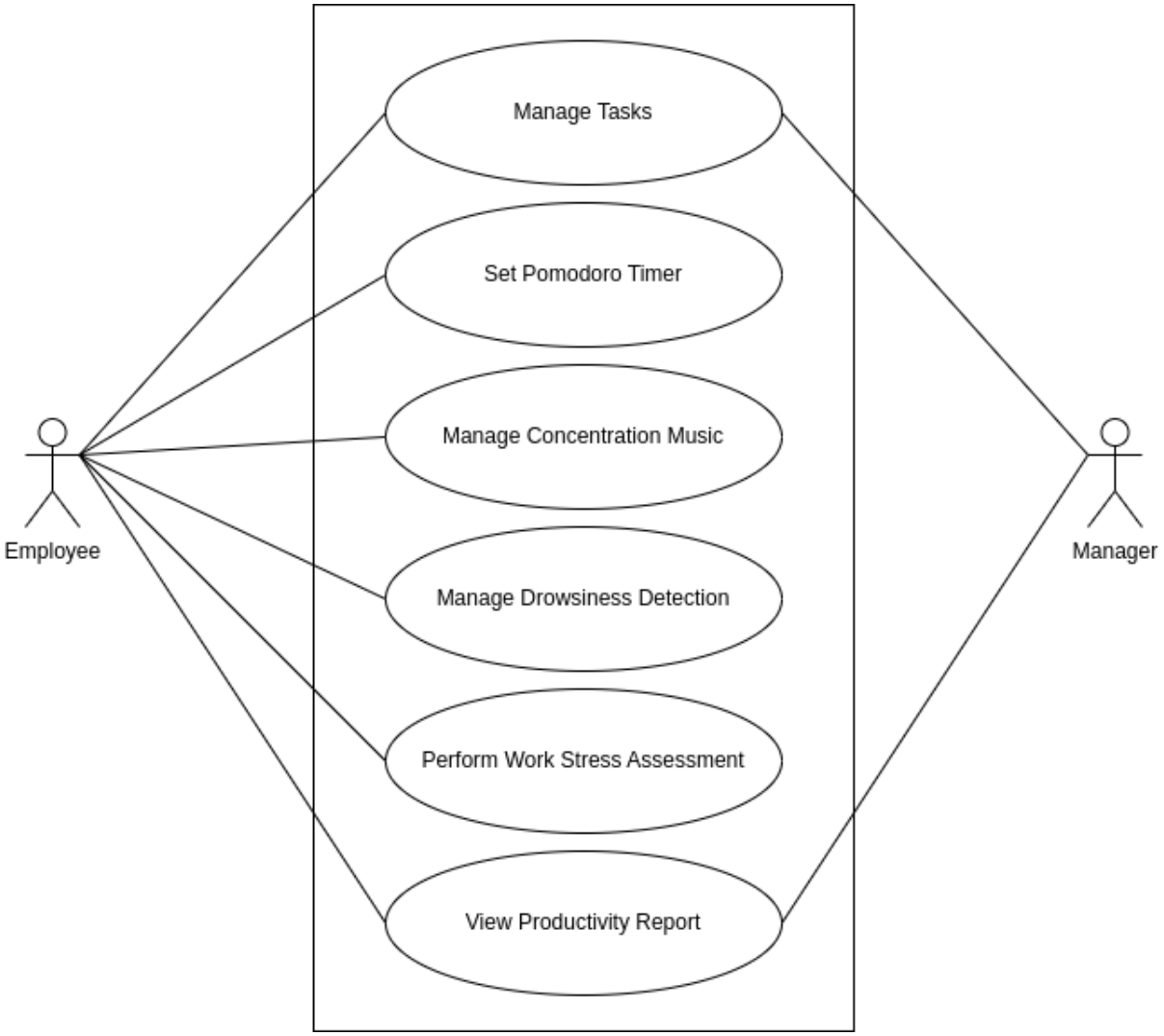

Figure 3.2: Use Case Diagram

3.3 Data Flow Diagram

3.3.1 Data Flow Diagram - Level 0

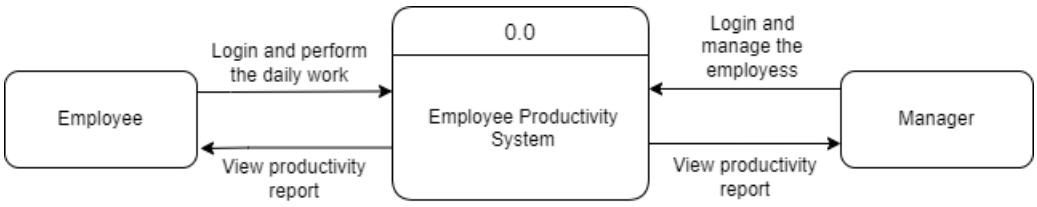

Figure 3.3: Data Flow Diagram Level 0

3.3.2 Data Flow Diagram - Level 1

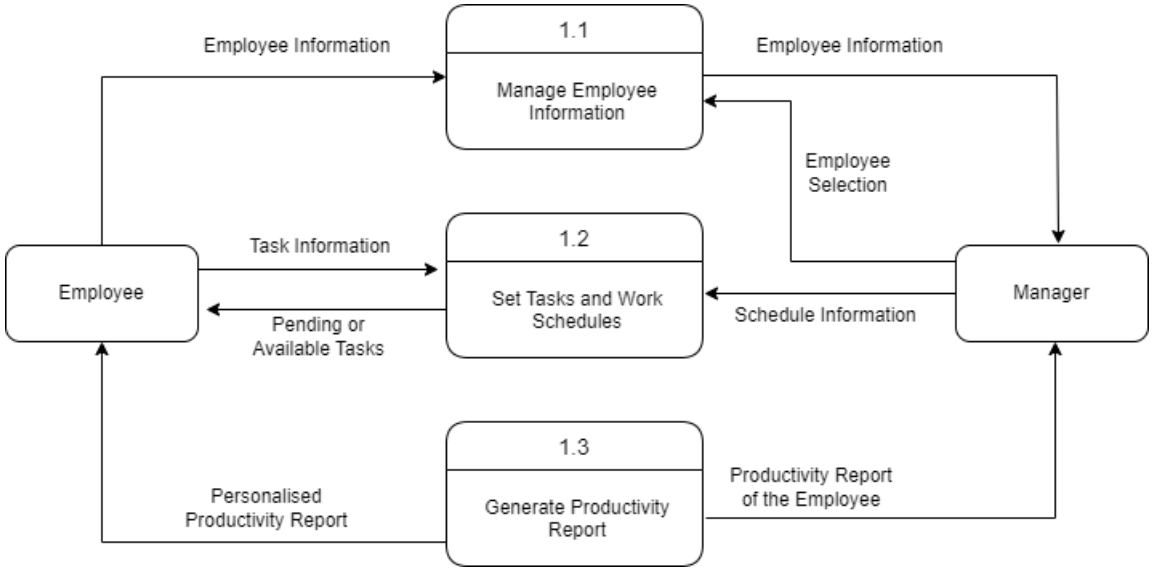

Figure 3.4: Data Flow Diagram Level 1

3.3.3 Data Flow Diagram - Level 2

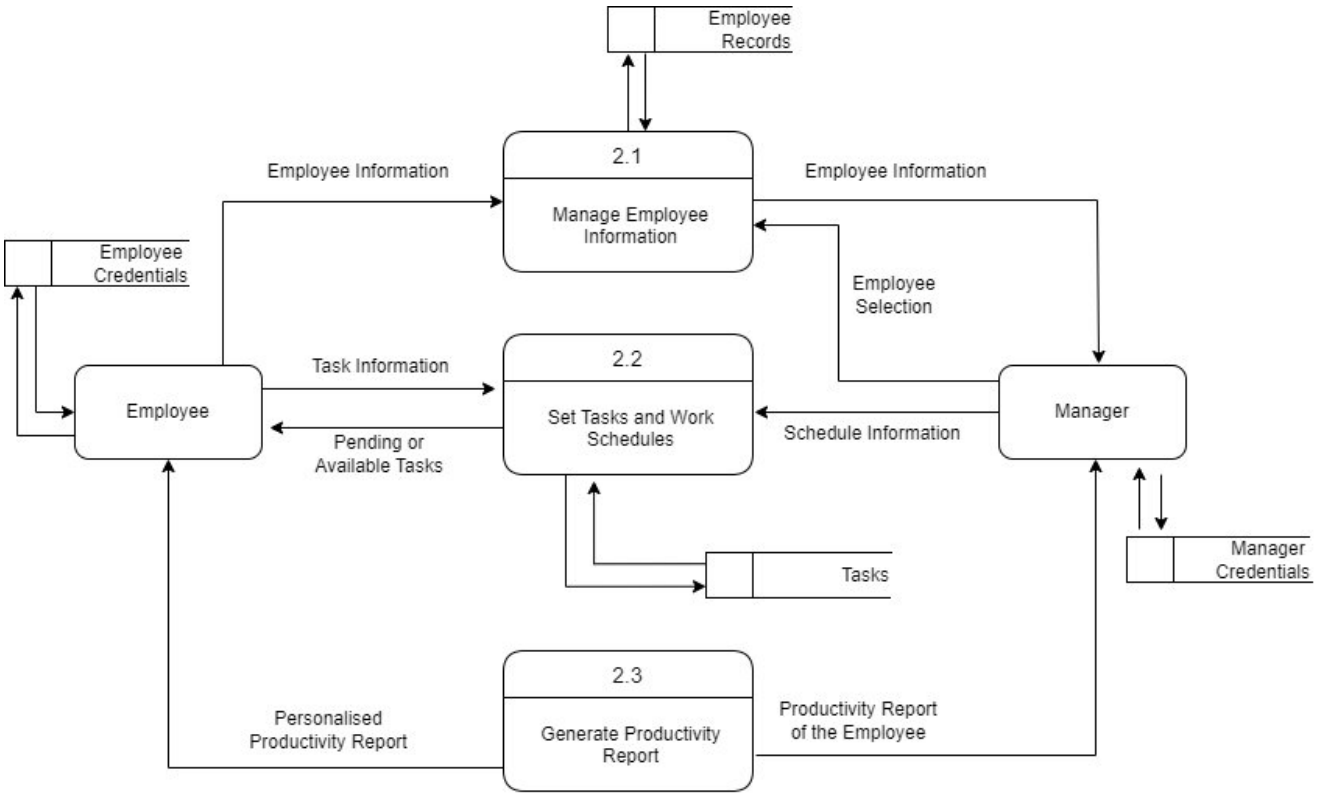

Figure 3.5: Data Flow Diagram Level 2

3.4 Class Diagram

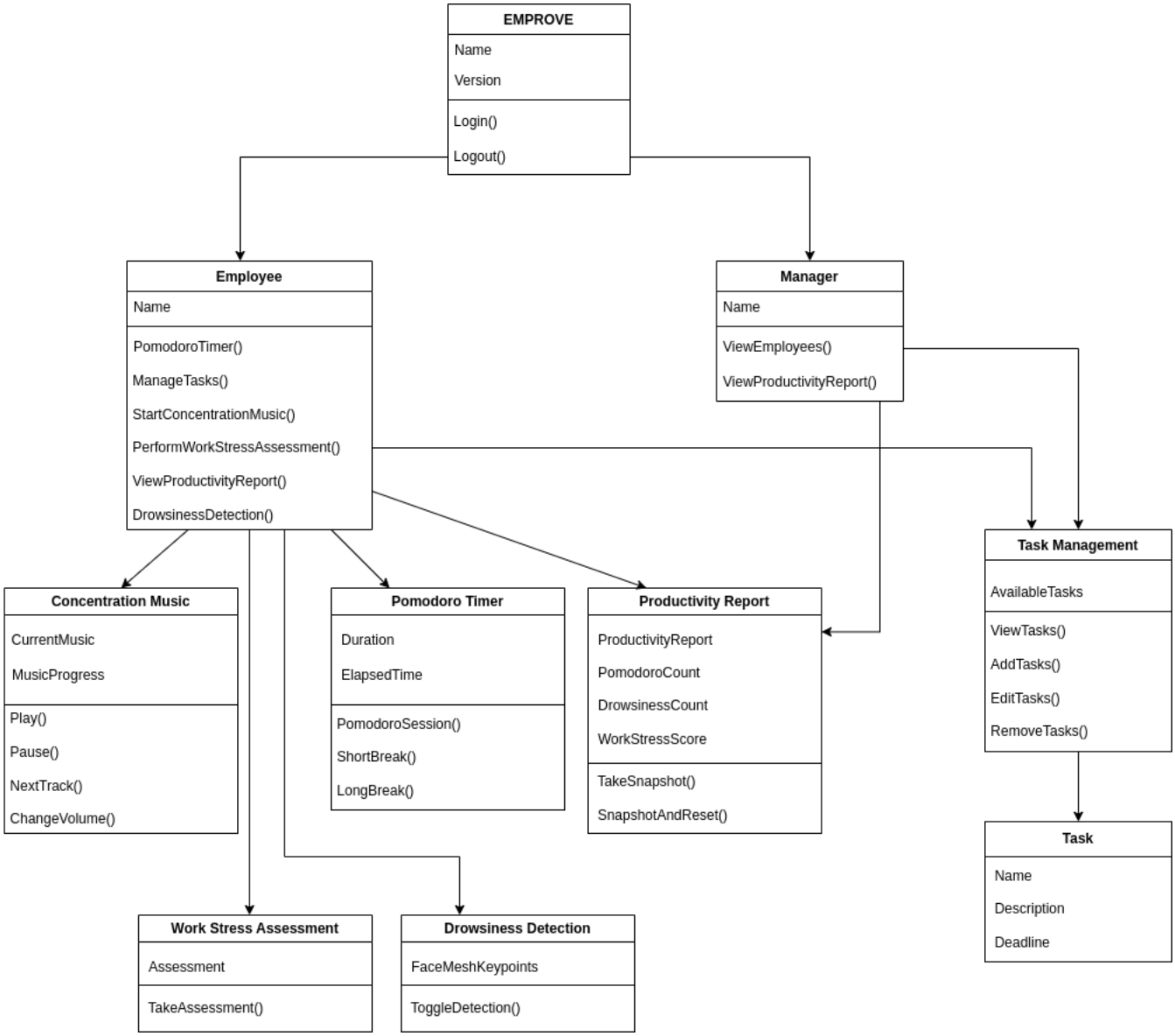

Figure 3.6: Class Diagram

3.5 System Overview

The employee productivity system consists of several key modules or processes that help employees manage their tasks and time effectively. The key modules include:

3.5.1 Task Management

Employees can create and prioritise their tasks through our task management interface. The interface includes features such as the ability to add new tasks, set deadlines, edit existing tasks, delete existing tasks, etc. Employees can also view their task list and deadlines on the employee dash-

board. The tasks can by created, read, updated and deleted (CRUD operations) by both employees and managers. The tasks are designed using Bootstrap cards and can be easily accessed through the employee dashboard.

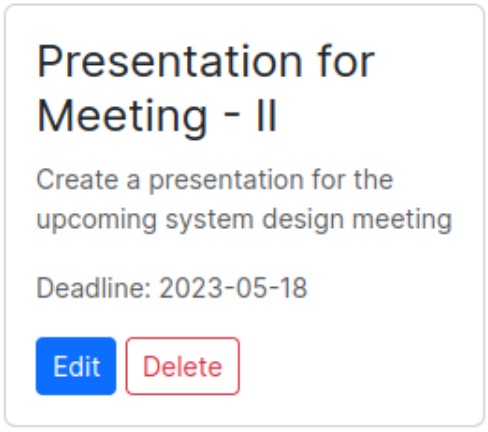

Figure 3.7: Example Task

3.5.2 Pomodoro Timer

Employees can track their work sessions using the Pomodoro timer, which helps them stay focused and take breaks at regular intervals. The Pomodoro timer is based on the Pomodoro Technique, which recommends working for 25 minutes and then taking a 5-minute break. After four Pomodoros (or 100 minutes of work), the employee takes a longer break of 15-20 minutes. Employees can choose from the standard Pomodoro session, short break or long break. The Pomodoro timer helps employees stay on track and avoid burnout by encouraging regular breaks. Each Pomodoro session is saved and is incremented in the productivity report.

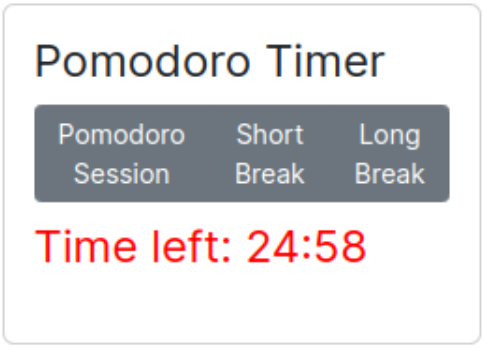

Figure 3.8: Pomodoro Session

3.5.3 Concentration Music

During working hours or breaks, employees can choose to listen to concentration music to help them relax and refocus. The concentration music feature includes a selection of instrumental tracks designed to promote relaxation and focus. Employees can choose the track they want to listen to and adjust the volume as needed. This module is implemented using the React Jinke music player component. The component has play-pause functionality, music progress bar, change track feature, etc.

3.5.4 Drowsiness Detection

In the background, the system uses the Media Pipe Face Mesh model to determine if the employee is drowsy or alert by calculating the Mouth Aspect Ration (MAR) and Eye Aspect Ratio (EAR) from the 468 facial keypoints in realtime. The MAR and EAR is computed using the formula:

$$MAR = \frac{\| p2 - p6 \| + \| p3 - p5 \|}{2 \| p1 - p4 \|} \tag{3.1}$$

$$EAR = \frac{\| p2 - p6 \| + \| p3 - p5 \|}{2 \| p1 - p4 \|} \tag{3.2}$$

Here, $\| pi - pj \|$ represents the Euclidean distance between the keypoints pi and pj. For any pi with coordinates (xi, yi) and pj with coordinates (xj, yj), the Euclidean distance is calculated using the formula:

$$\| pi - pj \| = \sqrt{(xj - xi)^2 + (yj - yi)^2} \tag{3.3}$$

Figure 3.9: Representation of Eye Aspect Ratio

A threshold value is set for MAR and EAR. If the instantaneous value of MAR is above the threshold or if the instantaneous value of EAR is below the threshold, the person is said to be

drowsy. If the system detects that the employee is drowsy, it can alert the employee. The pre-trained model achieves an accuracy of 95-98% with varying demographics. The system provides the drowsiness count as well as a detailed analysis in the productivity report.

3.5.5 Work Stress Assessment

The Workplace Stress Assessment proposed by the American Institute of Stress and the Marlin Company is a survey tool that assesses an individual's stress levels in the workplace. It consists of eight statements that the user rates on a scale of 1 to 5 based on how often they feel that way at work. The statements relate to physical and emotional well-being, workload, communication, utilization of skills, recognition, job pressure, interference with personal life and control over work duties. The total score is calculated by adding up the scores for each statement and the user can see how they compare to other people based on their demographic (gender and age). The total score is used to determine one's stress level and scores can range from "fairly low" to "severe." This scale helps individuals or organisations to assess the work stress experienced at their workplace and can take necessary steps to reduce stress levels or suggest professional assistance if health is significantly affected.

Table 3.1: Interpretation of Work Stress Score

Work Stress Score	**Description**
08 to 15	Chilled out and relatively calm
16 to 20	Fairly low
21 to 25	Moderate stress
26 to 30	Severe
31 to 40	Stress level is potentially dangerous

3.5.6 Productivity Report

Employees and Managers can view reports on employee productivity to help them identify areas for improvement and track progress over time. The reports include metrics such as the number of pending tasks, Pomodoro count, drowsiness count, work stress score and the employee's overall productivity over a given time period. The data is plotted and is visualised using a multi-line chart with x-axis for the timestamp and y-axis for Pomodoro count, drowsiness count and work stress score. The graph gives an insight of the productivity and progress of the employee over time. This data can help managers identify areas where the employees may be struggling and provide support to help them improve. Employees, on other hand, can have a detailed understanding of their workplace productivity from these statistics.

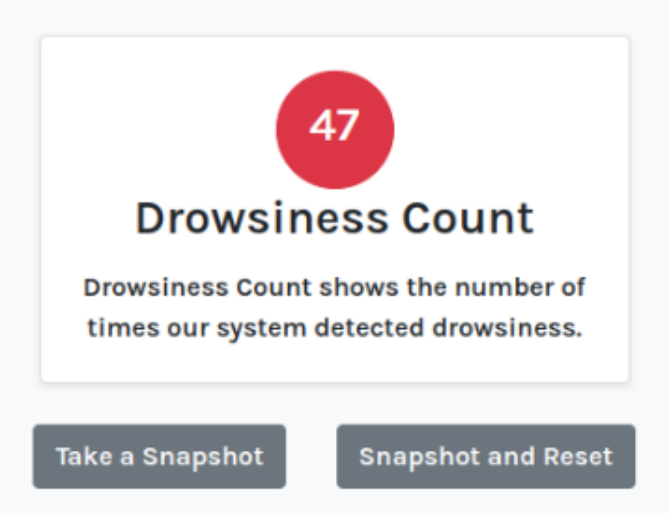

Figure 3.10: Drowsiness Count

3.6 System Requirements

3.6.1 Hardware Requirements

The productivity system will run on desktop or laptop computer with a minimum of 4 GB RAM, a storage capacity (HDD or SSD) of atleast 250 GB and an Intel Pentium processor or higher. A webcam, a speaker and a dedicated internet connection is necessary for the system to run.

3.6.2 Software Requirements

The productivity system will run on any web browser such as Google Chrome or Mozilla Firefox. It is independent of the operating system used and has tested to work with both Windows (Windows 7 or up) and Linux operating system such as Ubuntu and Fedora. The software system is built using React, Embedded Javascript (EJS) and Bootstrap for the frontend. It uses Node.js and Express.js for the backend. The database used is MongoDB, which is a NoSQL based database.

3.6.3 React

React is a popular JavaScript library for building user interfaces. It was created by Facebook and is widely used for developing web and mobile applications. React allows developers to create reusable UI components and manage the state of the application more efficiently. The library uses a declarative approach to programming, which means that the developer specifies what the UI should look like and React takes care of updating the DOM as needed. React was first released in 2013 and has since become one of the most widely used JavaScript libraries. It is open-source and has a large community of developers contributing to its development. React is often used in combination with other libraries and frameworks, such as Redux, React Native and Next.js. It is

also used by many popular websites, including Airbnb, Netflix and Instagram.

3.6.4 Embedded Javascript (EJS)

EJS, short for Embedded JavaScript, is a simple templating language used to generate HTML markup with plain JavaScript. It allows developers to write dynamic templates and reuse code snippets, making it easier to build complex web applications. EJS does not enforce any particular way of organizing code and does not require developers to learn a new syntax. Instead, it provides a familiar syntax that leverages the power of JavaScript to generate HTML. EJS also offers a range of features such as conditional statements, loops and partials that make it easier to create reusable components and build scalable applications.

3.6.5 Bootstrap

Bootstrap is a popular opensource frontend framework used for creating responsive web pages and applications. It is designed to make web development easier and faster by providing a set of pre-built HTML, CSS and JavaScript components that can be easily customized and combined to create a variety of user interfaces. Bootstrap is widely used by developers because of its ease of use, flexibility and responsiveness. It was originally developed by Twitter and is now maintained by a community of developers. Bootstrap includes a wide range of pre-built components such as navigation bars, forms, buttons and modals, among others. These components can be easily customized using CSS classes and JavaScript plugins. Bootstrap also includes a set of utility classes that can be used to style text, spacing and other elements on a page.

3.6.6 Node.js

Node.js is a cross-platform, opensource JavaScript runtime environment that can execute JavaScript code outside of a web browser. It allows developers to use JavaScript for server-side scripting, which means that server-side web applications can be written in JavaScript instead of languages like Python or Ruby. Node.js provides an event-driven architecture and non-blocking I/O API designed to optimize scalability and throughput. Node.js is commonly used for building scalable network applications such as web servers, chat servers and real-time data-intensive applications. It is built on the Chrome V8 JavaScript engine, which compiles JavaScript code into machine code for faster execution.

3.6.7 Express.js

Express.js is a web application server framework designed for building single-page, multi-page and hybrid web applications using Node.js. Express has become the standard server framework

for Node.js. Express is a minimal and flexible Node.js web application framework that provides a robust set of features for building web applications. The framework allows developers to create web applications quickly and easily by providing a variety of tools and features, including routing, middleware and templating engines.

3.6.8 MongoDB

MongoDB is a document-oriented NoSQL database that allows for high-volume data storage. Instead of using tables and rows, MongoDB uses collections and documents, with the latter consisting of key-value pairs which are the basic unit of data in MongoDB. Collections contain sets of documents and function as the equivalent of relational database tables. Some key features of MongoDB include the fact that each database contains collections which in turn contain documents, with each document able to be different with a varying number of fields. The document structure is more in line with how developers construct their classes and objects in their respective programming languages, with clear structures made up of key-value pairs. Rows or documents in MongoDB do not need to have a schema defined beforehand and the data model available within MongoDB allows for hierarchical relationships, arrays and other complex structures to be more easily represented.

3.6.9 MediaPipe Face Mesh

MediaPipe Face Mesh is a machine learning pipeline that predicts 468 3D facial landmarks to infer the surface geometry of a human face. It is a lightweight model that can estimate 468 3D face landmarks in real-time on mobile devices using deep neural networks to infer the 3D surface geometry. The pipeline makes use of multiple independent models, each of which is locally optimized to their task - pose, hand and face detection. These models coordinate with each other, but they use separate inputs and the input vector for one model may not be suitable for another model.

MediaPipe Face Mesh uses deep neural networks to estimate the 3D surface geometry of a human face. The model is lightweight and can run in real-time on mobile devices. It can track landmarks on a single face or multiple faces and establish a metric 3D space and use the face landmark screen positions to estimate face geometry within that space. MediaPipe Face Mesh can be used for a variety of applications, including remote gesture interfaces, full-body augmented reality, sign language recognition, sports/activity analytics and more.

3.7 Methodology

The employee can register himself at the website using the Sign Up feature. To register, the employee must provide his name, email address and must create a password. The password provided by the employee is hashed using MD5 hashing technique, which encrypts the user password into a 32 character long hexadecimal string. This encryption technique ensures that the password of the employees remain safe and secure in case of a database breach.

The registered users can sign in into their employee dashboard using the Sign In feature available at the homepage of the web application. Once the employee is signed in, he can see the available tasks and their deadlines, modify existing tasks or create new ones. The manager can also assign, edit or delete tasks for each employee. The task management feature is implemented as a React component and is displayed onto the dashboard using Bootstrap cards.

Within the employee dashboard, a Pomodoro timer component is also displayed. It consists of 3 options to choose from - Pomodoro session, short break and long break. Each of these are of 25 minutes, 5 minutes and 15 minutes in duration, respectively. Once the employee successfully complete a Pomodoro session, the Pomodoro count gets incremented by one. At the bottom, a music player is made available so that the employee can choose a focus music of his choice. The music player is implemented using the React Jinke library and has support for play-pause, next track, view playlist, control volume, seek progress bar, etc.

The ON-OFF detection button at the bottom acts as a toggle button for the drowsiness detection feature. When turned on, a real-time face mesh of the employee is generated from 468 facial keypoints. This live face mesh is displayed onto the screen. In the background, MAR and EAR is computed for each of the frames. A threshold value is set for MAR and EAR while implementation. If the instantaneous value of MAR is above the threshold or if the instantaneous value of EAR is below the threshold, the person is said to be drowsy. If the system detects drowsiness, it alerts the employee and the drowsiness count gets incremented by one for each alarm.

On the top, a link to the WS Assessment is provided. The Work Stress Assessment consists of 8 questions or statements that determines the level of stress experienced by the employee at his workplace. The employee can occasionally (weekly, monthly or half-yearly) take part in the assessment and evaluate his stress levels. The Work Stress Score is also recorded and can be viewed from the the productivity report. The productivity report provides a detailed overview of the productivity of an employee. It shows the Pomodoro count, drowsiness count and work stress score of the employee over a period of time. Further, a plot of these is presented and the employee can evaluate how his productivity has increased over time. The manager can also view the productivity report of each of the employees assigned to him.

Chapter 4

RESULTS

4.1 Case Study: Increasing Productivity with Emprove

We conducted a case study to evaluate the effectiveness of Emprove, the employee productivity system, in enhancing the productivity of a software engineer at an IT company. This case study aimed to assess the impact of using Emprove over a one-week trial period, specifically analyzing how the engineer's productivity improved during the trial period.

The case study involved a software engineer who had been facing challenges in managing tasks, prioritizing work and maintaining focus. The software engineer is working from home (WFH) for a software company based on Kochi, Kerala. The engineer was introduced to Emprove, the web-based employee productivity system designed to assist in organizing workload, managing time efficiently and increasing overall productivity.

During the one-week trial, the software engineer utilized Emprove during regular working hours, which were from 9 am to 6 pm, Monday to Friday. Throughout the one-week trial, the software engineer actively engaged with the various features provided by Emprove, including task management, the Pomodoro timer, concentration music, drowsiness detection, work stress assessment and productivity reports.

The use of the employee productivity system had a significant positive impact on the software engineer's productivity during the trial period. The following improvements were observed in various aspects of the engineer's work:

- Utilizing Emprove's task management feature, the software engineer effectively broke down projects into smaller, manageable tasks. This approach facilitated improved prioritization and focus on high-priority assignments, resulting in heightened efficiency and timely task completion.
- The integration of the Pomodoro technique within Emprove enabled the software engineer to practice effective time management and maintain focus. By working in concentrated intervals

of 25 minutes followed by short breaks, the engineer established a structured work routine, enhancing productivity and preventing burnout.

- The availability of concentration music within Emprove created an environment conducive to focused work. The software engineer selected soothing instrumental tracks, minimizing distractions and promoting concentration. The presence of background music enhanced focus, productivity and job satisfaction.
- Emprove's drowsiness detection feature played a vital role in maintaining the software engineer's alertness during working hours. By continuously monitoring facial features in real-time, the system detected signs of drowsiness and alerted the engineer when attention wavered. This proactive measure prevented productivity slumps caused by fatigue, ensuring consistently high performance throughout the day.
- The work stress assessment component provided valuable insights into the engineer's stress levels. Regular evaluation of stress patterns enabled the software engineer to identify sources of stress and take necessary steps to address them. This self-awareness empowered the engineer to manage stress effectively, leading to increased resilience and productivity.
- Emprove's comprehensive productivity reports offered a holistic overview of the engineer's performance and areas for improvement. The detailed analysis facilitated identification of strengths and weaknesses, enabling the engineer to refine work strategies and optimize productivity further.

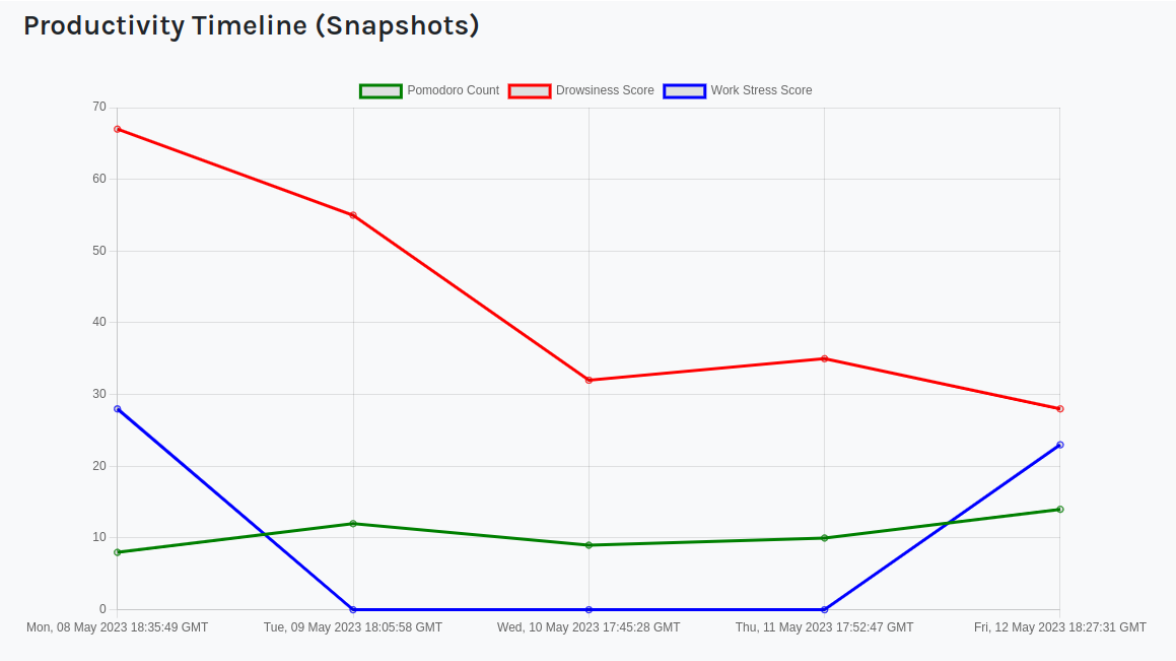

Figure 4.1: Productivity Timeline

The productivity timeline of the engineer is provided in Figure 4.1. The graph shows that the drowsiness count decreased over time and the employee managed to have increased the number of Pomodoro sessions. The work stress score on Friday also appears to have reduced (lower stress) when compared to the previous assessment taken on Monday.

Table 4.1: Productivity Metrics for a Week

Sl. No.	Timestamp	Pomodoro Count	Drowsiness Count	Work Stress Score
1	Mon, 08 May 2023 18:35:49 GMT	8	67	28
2	Tue, 09 May 2023 18:05:58 GMT	12	55	-
3	Wed, 10 May 2023 17:45:28 GMT	9	32	-
4	Thu, 11 May 2023 17:52:47 GMT	10	35	-
5	Fri, 12 May 2023 18:27:31 GMT	14	28	23

This case study demonstrates the significant positive impact of a comprehensive employee productivity system on the productivity of a software engineer during a one-week trial period. The integration of features such as task management, the Pomodoro technique, concentration music, drowsiness detection, work stress assessment and productivity reports contributed to enhanced productivity, focus and overall job satisfaction. The findings highlight the potential benefits of implementing a comprehensive employee productivity system, such as Emprove, within IT companies and similar professional environments. By empowering individuals to effectively manage workloads, optimize performance and mitigate stress-related challenges, such systems have the potential to significantly contribute to overall organizational productivity and employee well-being.

Chapter 5

CONCLUSION

The employee productivity system is a powerful tool for helping employees track their tasks, manage their time and stay focused and productive throughout the workday. With features such as a Pomodoro timer, task management, concentration music, work stress assessment and drowsiness detection, the software offers a range of benefits for both employees and managers. For employees, the software helps them increase their productivity, manage their tasks more effectively and achieve a better work-life balance. The Pomodoro timer and concentration music features help employees take breaks and relax, improving their overall well-being. The drowsiness detection feature can help employees stay alert and avoid mistakes caused by fatigue. For managers, the software provides valuable insights into employee productivity and areas for improvement. They can assign tasks to those employees assigned to them. The reports feature allows managers to view metrics such as the number of assigned tasks, Pomodoro count, drowsiness count, work stress score and the employee's overall productivity over a given time period. This data can help managers identify areas where employees may be struggling and provide support to help them improve.

References

[1] Prasath N, Sreemathy J, Vigneshwaran P, ”Driver Drowsiness Detection Using Machine Learning Algorithm”, 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Jun 2022.

[2] Yury Kartynnik, Artsiom Ablavatski, Ivan Grishchenko, Matthias Grundmann, ”Real-time Facial Surface Geometry from Monocular Video on Mobile GPUs”, arXiv preprint arXiv:1907.06724v1, Jul 2019.

[3] Amit A J, Gautam Shankararam S R, Pradeep P , Perumalraja R, Kamalesh S, ”Framework for Preventing Procrastination and Increasing Productivity”, 2021 3rd International Conference on Signal Processing and Communication (ICPSC), May 2021.

[4] Brittany Johnson, Thomas Zimmermann, Christian Bird, ”The Effect of Work Environments on Productivity and Satisfaction of Software Engineers”, IEEE Transactions on Software Engineering, Mar 2019.

[5] Sean Jentz, Victor Winkelmann, ”The Power of Music - How environmental variables can disrupt or enable productivity”.

[6] American Institute of Stress, Yonkers, NY and The Marlin Company, North Haven, CT, ”The Workplace Stress Scale”.

[7] Jalaja Shanmugalingam, David Lario, Yongsheng Ma, ”Feature Based Statistical Model of Employee Productivity with Real Time Checked Data”, IEEE International Conference on Industrial Engineering and Engineering Management, Mar 2021.

Appendix A

Example Code

A.1 Employee Dashboard

```
import React, { useState, useEffect } from "react"
import { createRoot } from "react-dom/client"
import Axios from "axios";
import HeaderBar from "./components/HeaderBar";
import CreateNewForm from "./components/CreateNewForm";
import TaskCard from "./components/TaskCard";
import ReactJkMusicPlayer from "react-jinke-music-player";
import 'react-jinke-music-player/assets/index.css';
import PomodoroTimer from "./components/PomodoroTimer";
import FacialLandmarks from "./components/FacialLandmarks";

function App() {
    const [tasks, setTasks] = useState([])
    const [users, setUsers] = useState([])
    let [toggleDetection, setToggleDetection] = useState(false);

    useEffect(() => {
        async function go() {
            const taskResponse = await Axios.get("/api/tasks")
            const userResponse = await Axios.get("/api/users")
            setTasks(taskResponse.data)
            setUsers(userResponse.data)
        }
        go()
```

```
}, [])

const audioList = [
    {
        musicSrc: "./music/letter-to-a-friend.mp3",
        cover: "./music/letter-to-a-friend.jpeg",
        name: "Letter to a Friend",
        singer: "Robert Gromotka"
    },
    {
        musicSrc: "./music/leaves-from-the-vine.mp3",
        cover: "./music/leaves-from-the-vine.jpeg",
        name: "Leaves from the Vine",
        singer: "Samuel Kim"
    }]

    const toggleFaceDetection = () => {
        return setToggleDetection(!toggleDetection);
      };

return (
    <div>
        <HeaderBar email={users.email} userComponentRender={userComponentRender}/>
        <div className="container">
            <div className="task-grid">
                <CreateNewForm setTasks={setTasks} />
                {userComponentRender && <PomodoroTimer email={users.email}/>}
                {tasks.map(function (task) {
                    return <TaskCard key={task._id} name={task.name}
                    description={task.description} deadline={task.deadline}
                    id={task._id} setTasks={setTasks} />
                })}
                {toggleDetection ? <FacialLandmarks email={users.email}/> : ""}
            </div>
        </div>
        {userComponentRender && <ReactJkMusicPlayer audioLists={audioList}
```

```
            extendsContent={<button className="btn btn-primary" onClick=
            {toggleFaceDetection}>ON | OFF Detection</button>} theme="dark"
            mode="full" autoPlay={false} toggleMode={false} responsive={false}
            showThemeSwitch={false} showDownload={false} showPlayMode={false}
            showReload={false} />}
        </div>
    );
}

const root = createRoot(document.querySelector("#app"));
root.render(<App />);
```

A.2 Face Mesh Generation using MediaPipe

```
import React, { useRef, useState } from "react";
import "@tensorflow/tfjs";
import "@tensorflow/tfjs-backend-webgl";
import "@mediapipe/face_mesh";
import Webcam from "react-webcam";
import { runDetector } from "./utilities/detector";

const inputResolution = {
    width: 380,
    height: 350,
};
const videoConstraints = {
    width: inputResolution.width,
    height: inputResolution.height,
    facingMode: "user",
};
function FacialLandmarks(props) {
    const canvasRef = useRef(null);
    const [loaded, setLoaded] = useState(false);

    const handleVideoLoad = (videoNode) => {
        const video = videoNode.target;
```

```
        if (video.readyState !== 4) return;
        if (loaded) return;
        runDetector(video, canvasRef.current);
        setLoaded(true);
    };
    return (
        <div>
            <Webcam
                width={inputResolution.width}
                height={inputResolution.height}
                style={{ visibility: "hidden", position: "absolute" }}
                videoConstraints={videoConstraints}
                onLoadedData={handleVideoLoad}
            />
            <canvas
                ref={canvasRef}
                width={inputResolution.width}
                height={inputResolution.height}
                style={{ position: "absolute" }}
            />
            {loaded ? <></> : <header>Loading...</header>}
        </div>
    );
}

export default FacialLandmarks;
```

A.3 MAR and EAR Computation

```
import * as faceLandmarksDetection from "@tensorflow-models/face-landmarks-
detection";
import { drawMesh } from "./drawMesh";
import axios from "axios";

function euclidianDistance(x1, y1, x2, y2) {
  return Math.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
```

```
}

function eyeAspectRatio(p1, p2, p3, p4, p5, p6) {
  var A = euclidianDistance(p2.x, p2.y, p6.x, p6.y)
  var B = euclidianDistance(p3.x, p3.y, p5.x, p5.y)
  var C = euclidianDistance(p1.x, p1.y, p4.x, p4.y)

  return ((A + B) / (2.0) * C)
}

function mouthAspectRatio(p1, p2, p3, p4, p5, p6) {
  var A = euclidianDistance(p2.x, p2.y, p6.x, p6.y)
  var B = euclidianDistance(p3.x, p3.y, p5.x, p5.y)
  var C = euclidianDistance(p1.x, p1.y, p4.x, p4.y)

  return ((A + B) / (2.0) * C)
}

export const runDetector = async (video, canvas) => {
  const model = faceLandmarksDetection.SupportedModels.MediaPipeFaceMesh;
  const detectorConfig = {
    runtime: "tfjs"
  };
  const detector = await faceLandmarksDetection.createDetector(
    model,
    detectorConfig
  );
  var updateCount;
  const detect = async (net) => {
    const estimationConfig = { flipHorizontal: false };
    const faces = await net.estimateFaces(video, estimationConfig);
    try {
      var MAR = mouthAspectRatio(faces[0].keypoints[61], faces[0].keypoints[37],
      faces[0].keypoints[267], faces[0].keypoints[291], faces[0].keypoints[314],
      faces[0].keypoints[84])
```

```
      var L_EAR = eyeAspectRatio(faces[0].keypoints[33],
      faces[0].keypoints[160], faces[0].keypoints[158], faces[0].keypoints[133],
      faces[0].keypoints[153], faces[0].keypoints[144])

      var R_EAR = eyeAspectRatio(faces[0].keypoints[362],
      faces[0].keypoints[385], faces[0].keypoints[387], faces[0].keypoints[263],
      faces[0].keypoints[373], faces[0].keypoints[380])

      if (MAR > 6000 || (L_EAR < 250 && R_EAR < 250)) {
        var audio = new Audio("notification.wav");
        audio.play();
        updateCount = true;
      }
      else {
        if (updateCount == true) {
          axios.post("/drowsiness-count", "updateCount");
          updateCount = false;
        }
      }
    }
    catch (e) {
      // neglect
    }
    const ctx = canvas.getContext("2d");
    requestAnimationFrame(() => drawMesh(faces[0], ctx));
    detect(detector);
  };
  detect(detector);
};
```

Appendix B

MongoDB Collections

B.1 User Database (Employee Database)

Table B.1: User Database

MongoDB Document	MongoDB Datatype
name	String
email	String
password	String
pomodoroCount	Number
drowsinessCount	Number
workStressScore	Number
pomodoroCountArray	Array(Number)
drowsinessCountArray	Array(Number)
workStressScoreArray	Array(Number)
timestampArray	Array(String)

B.2 Admin Database (Manager Database)

Table B.2: Admin Database

MongoDB Document	MongoDB Datatype
name	String
email	String
password	String
users	Array(String)

B.3 Task Database

Table B.3: Task Database

MongoDB Document	**MongoDB Datatype**
name	String
description	String
deadline	String
assignedTo	String
completed	Boolean

Appendix C

Screenshots

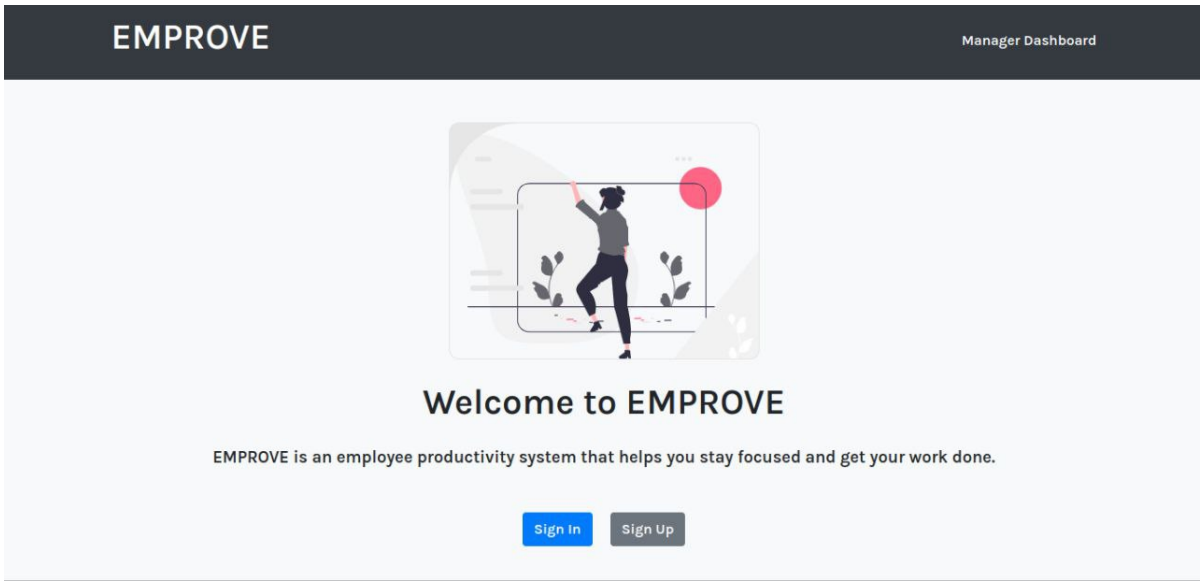

Figure C.1: Home Page

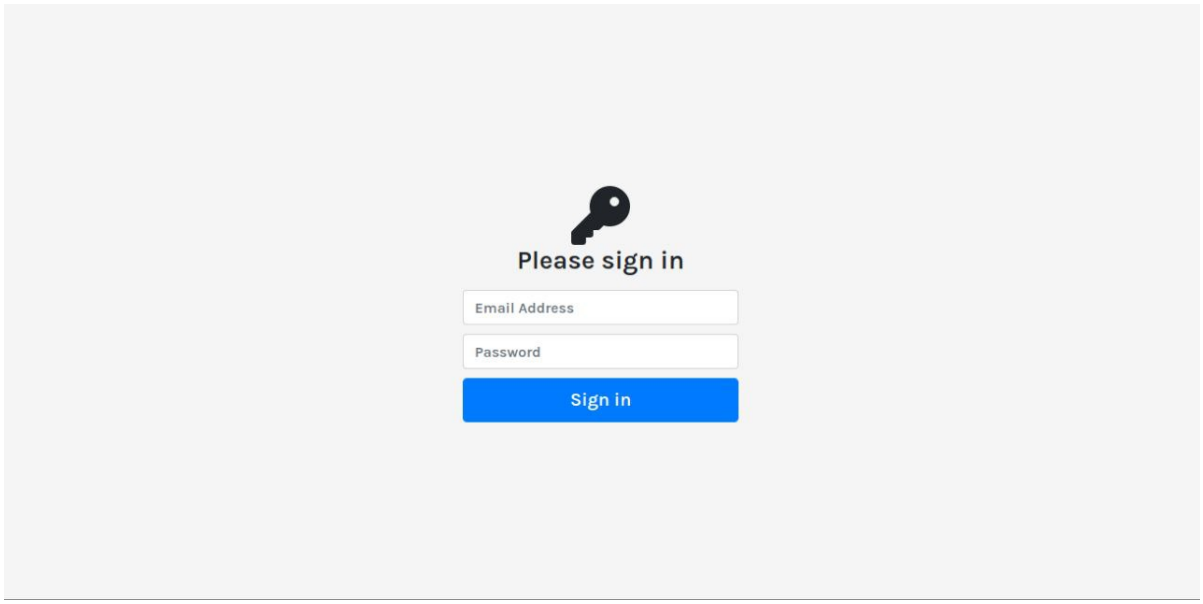

Figure C.2: Sign In Page

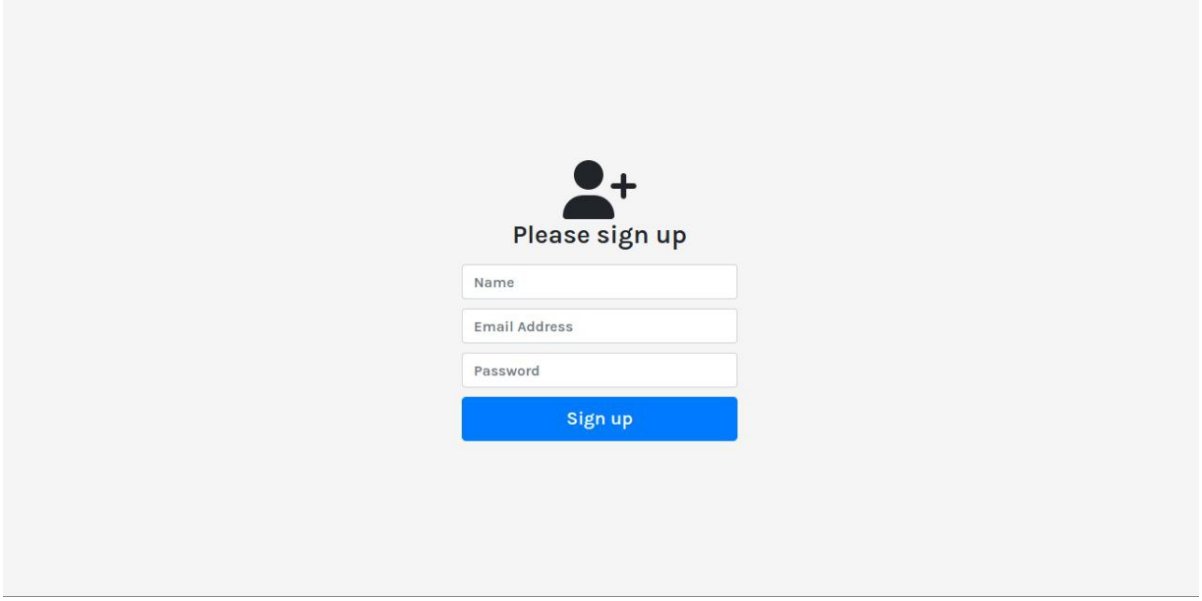

Figure C.3: Sign Up Page

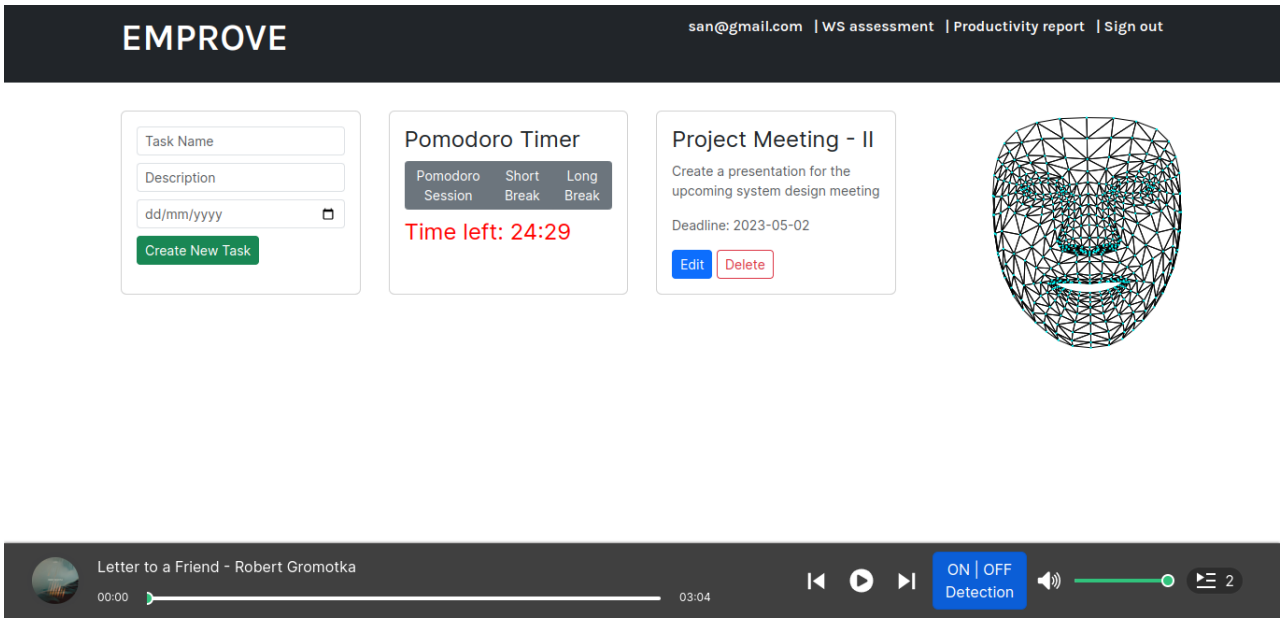

Figure C.4: Employee Dashboard

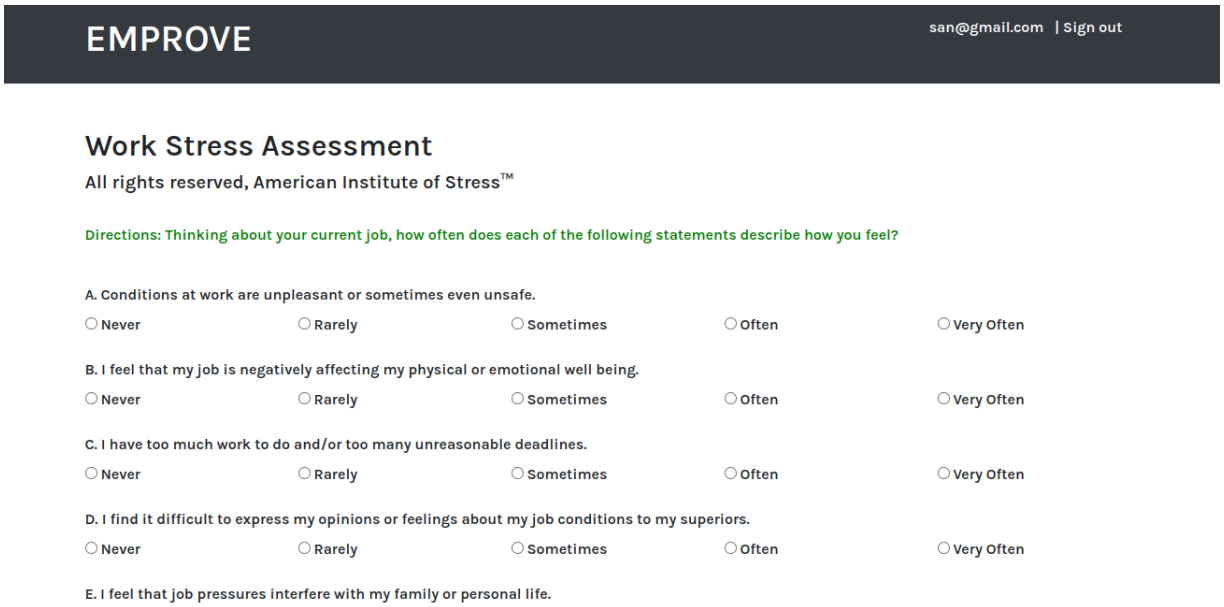

Figure C.5: Work Stress Assessment

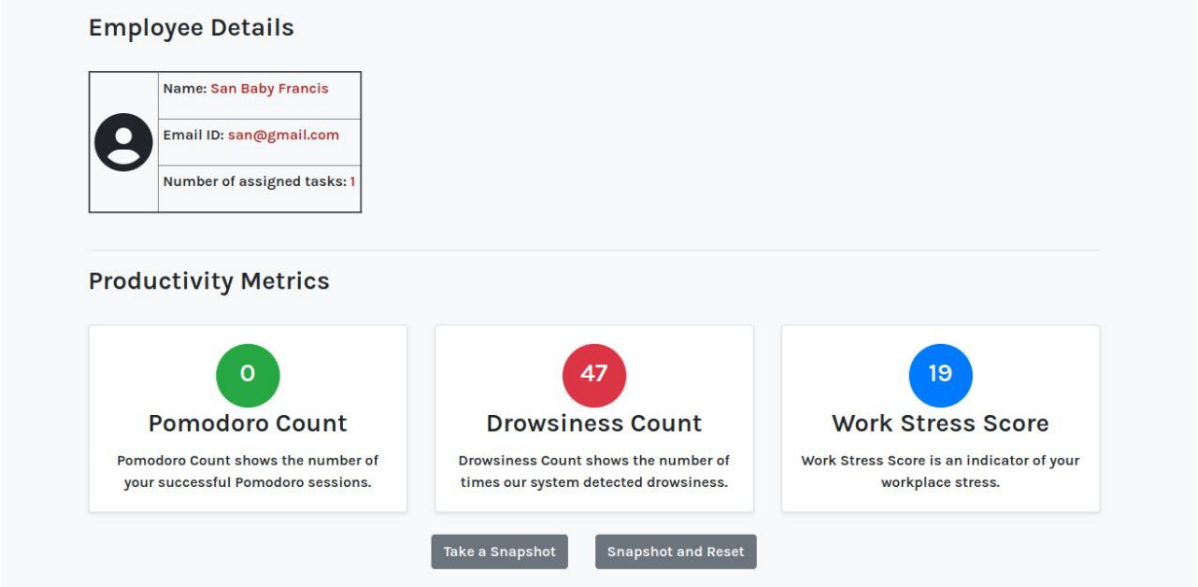

Figure C.6: Productivity Report