

AI-Driven Employee Wellness Productivity Tracker

Yash Savani & Kenil Gopani

Master's in Software Engineering

San Jose State University

San Jose, United States

yashbharatbhai.savani@sjsu.edu

kenilghanashyambhai.gopani@sjsu.edu

Kartik Chindarkar & Vyshnavi D P

Master's in Software Engineering

San Jose State University

San Jose, United States

kartik.chindarkar@sjsu.edu

vyshnavi.dyvandinnepullareddy@sjsu.edu

Abstract—The AI-Driven Employee Wellness and Productivity Tracker is designed to monitor and improve the mental well-being and productivity of employees within organizations. By leveraging machine learning and cloud technologies, this system aims to offer personalized interventions for employees, ensuring a healthy work-life balance while enhancing overall workplace productivity. The solution collects data through surveys and uses Azure AI services to analyze and deliver meaningful insights. This paper provides a detailed overview of the system architecture, functionalities, use cases, and the technologies employed in building the solution.

Index Terms—Employee Wellbeing, Productivity Tracking, Artificial Intelligence, LLAMA 3.1, Personalized Recommendations, Work-Life Balance, Stress Management, Cloud Computing, Spring Boot, MongoDB, React, Azure AD B2C, Machine Learning, Data Analytics, Employee Engagement, Mental Health, HR Analytics, Survey Data, Data-Driven Interventions, Workplace Productivity, Employee Performance, Wellness Intervention.

I. INTRODUCTION

In today's fast-paced work environments, the mental well-being of employees is often overlooked despite its direct correlation with productivity and overall job satisfaction. Employees who experience high stress levels may face burnout, reduced efficiency, and engagement. Conversely, organizations that prioritize employee wellness can foster a healthier work culture, leading to increased productivity and reduced turnover. The AI-Driven Employee Wellness and Productivity Tracker is developed to address these challenges by providing organizations with a tool to monitor and improve the well-being and productivity of their employees.

The system collects data via user surveys and evaluates it using AI to generate personalized recommendations that promote better work-life balance and optimize productivity. These recommendations include suggestions for breaks, counseling options, and productivity tracking tailored to individual employees' needs.

II. SYSTEM ARCHITECTURE

The system is designed to provide a seamless user experience while ensuring robust backend processing for data analysis and personalized recommendation generation. The

architecture is split into several components, each serving a specific function:

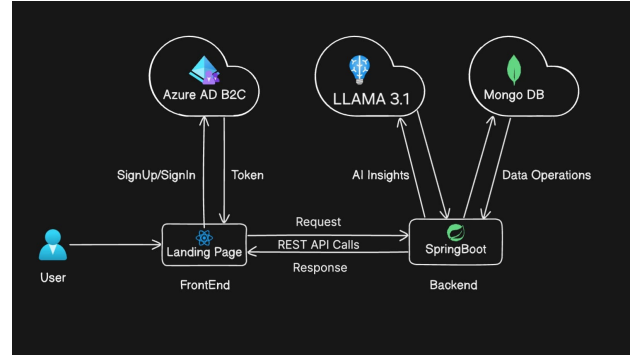


Fig. 1. System Architecture Diagram

A. Frontend (React)

The frontend of the system is built using React, a popular JavaScript library for building user interfaces. React enables the development of dynamic web pages that allow employees to interact with the platform seamlessly. The landing page provides the users with the option to sign up or sign in, and once authenticated, they can access their personalized wellness dashboard.

B. Azure AD B2C (Authentication)

For secure user authentication, the system uses Azure Active Directory B2C. Azure AD B2C enables secure login experiences by allowing employees to use their organizational credentials or social accounts. This service ensures the protection of user data while maintaining ease of access.

C. Backend (Spring Boot)

The backend is powered by Spring Boot, a Java-based framework known for its efficiency and scalability. Spring Boot handles REST API calls from the frontend, facilitating communication between the user interface and the database. It also manages the flow of data, ensuring that survey responses and other inputs are captured accurately. Furthermore, it interacts with Azure AI services for data analysis.

D. MongoDB (Database)

MongoDB is used as the primary database for storing employee data, survey responses, and productivity metrics. As a NoSQL database, MongoDB provides flexibility in handling unstructured data, which is ideal for the variety of survey data collected. The system uses MongoDB's schema-less nature to store user inputs, ensuring scalability as more employees interact with the platform.

III. FUNCTIONALITIES

The system provides a range of functionalities aimed at improving both employee wellness and productivity:

A. Employee Wellbeing Monitoring

Through surveys, the system collects data on an employee's stress levels, personality traits, and other mental health indicators. By analyzing these inputs, the system can suggest interventions that help mitigate stress, such as:

- Personalized break suggestions
- Meditation or mindfulness activity recommendations
- Access to counseling or support services

B. Productivity Tracking

The system tracks employee performance and productivity through work-related data such as task completion rates, hours worked, and efficiency metrics. Employees can visualize their productivity trends through interactive graphs, which highlight areas for improvement. This allows for a data-driven approach to managing and improving productivity.

C. Personalized Recommendations

AI-driven personalized recommendations are provided to users based on the analysis of their survey responses and productivity data. These recommendations may include:

- Break time suggestions based on work intensity and stress levels
- Activity suggestions tailored to the individual's preferences and hobbies
- Counseling or support resources if stress or burnout is detected

IV. PERSONA

The system is designed with several stakeholders in mind, each benefiting from its features in different ways:

A. Employees

Employees are the primary users of the system. They interact with the platform to privately track their productivity and mental wellbeing. The data remains confidential and cannot be accessed by anyone, including HR or managers. By receiving personalized suggestions for wellness activities and productivity management, employees can make informed decisions to improve their overall work-life balance.

B. HR Teams

HR teams can use the system to monitor the general wellbeing and productivity trends of employees across the organization. While they cannot access individual employees' data, they can view aggregated data, such as the organization's average wellbeing and productivity metrics. By accessing this overall data, HR professionals can identify potential stressors, provide targeted interventions, and support employees more effectively.

C. Managers

Managers are able to use the system to gain insights into the overall productivity levels and stress factors affecting their teams. They can only view aggregated team-level data, such as averages, and cannot access any individual employee's personal data. This ensures employee privacy while enabling managers to take proactive steps to maintain a healthy and productive work environment, ensuring that team members are not overwhelmed.

V. TECHNOLOGIES USED

A. React (Frontend Development)

React was chosen for frontend development due to its flexibility, ease of use, and efficient rendering capabilities. It provides a smooth user experience and allows for dynamic content updates without requiring full page reloads.

B. Azure AD B2C (Authentication)

Azure AD B2C provides a scalable and secure way to authenticate users. It simplifies the sign-in and sign-up processes while ensuring that user credentials are stored securely.

C. Spring Boot (Backend Development)

Spring Boot was chosen for the backend due to its simplicity and the rich set of features it offers for building RESTful APIs. It handles all backend operations, including API management and data processing, enabling seamless interaction with the frontend and the database.

D. MongoDB (Database Management)

MongoDB's NoSQL structure is ideal for handling the large and varied data types generated by user surveys and productivity metrics. It scales easily, providing a robust database solution as the system grows.

E. LLAMA 3.1 (Learning and Modeling Algorithm 3.1)

LLAMA 3.1 is a machine learning framework used to analyze large datasets and generate actionable insights. In the context of the AI-Driven Employee Wellness and Productivity Tracker, it is utilized within Azure AI Services to model and predict employee stress levels and overall well-being. By processing data collected from employee surveys and productivity metrics, it helps generate personalized recommendations for each employee. These recommendations include suggestions for break times, wellness activities, and other interventions to improve work-life balance. Its advanced data modeling capabilities enable the system to continuously refine its insights and

adapt to new inputs, ensuring that the recommendations remain relevant and effective over time. The use of LLAMA 3.1 enhances the system’s ability to provide data-driven solutions for improving employee well-being and productivity.

VI. TESTING

The AI-Driven Employee Wellness and Productivity Tracker underwent comprehensive testing to ensure the reliability and performance of both the backend and frontend components. Two primary technologies were utilized for testing: JUnit for the backend and Jest for the frontend. The testing process involved unit tests, integration tests, and UI tests to verify that all components functioned as expected and met the requirements. The following sections detail the testing strategies applied to each part of the system.

A. Frontend Testing (Jest)

For the frontend, Jest was employed to ensure the functionality and performance of the user interface components developed with React. The following testing practices were applied:

- **Unit Testing:** Individual React components were tested to confirm that they rendered correctly and interacted with the backend as expected. This included testing user interactions such as form submissions, survey data input, and navigation between pages.
- **Mocking API Calls** API calls to the backend were mocked to simulate responses from the server without needing a live backend. This allowed for testing the frontend’s behavior in response to both expected and error states.

src/components	80.3	35.71	70	82.81
AboutSection.jsx	100	100	100	100
AdvancedDocumentsSection.jsx	100	100	100	100
AnalyticsDashboardSection.jsx	100	100	100	100
ContactDetails.jsx	100	100	100	100
DataDisplay.jsx	100	100	100	100
FAQItem.jsx	100	100	100	100
FAQSection.jsx	100	100	100	100
FeatureCard.jsx	100	100	100	100
Features.jsx	100	100	100	100
FeaturesSection.jsx	100	100	100	100
Footer.jsx	100	100	100	100
HeroSection.jsx	100	100	100	100
LivePublishingSection.jsx	100	100	100	100
NavigationBar.jsx	63.63	35.71	38.46	67.74
PageLayout.jsx	100	100	100	100
RealTimeEditingSection.jsx	50	100	0	50

Fig. 2. Frontend testing code coverage

B. Backend Testing (JUnit)

JUnit was used for unit testing and integration testing of the backend services built with Spring Boot. The testing process involved the following:

- **Unit Testing:** Individual methods and functions within the backend were tested to ensure they operated correctly in isolation. This included testing service methods, controllers, and utility functions to confirm that each component handled its responsibilities as expected.
- **Mocking Dependencies:** To isolate individual components during testing, external dependencies such as the

database and external services (like Azure AI) were mocked. This approach ensured that unit tests remained focused and independent of external systems.

Element	Missed Instructions	Cov	Missed Branches	Cov	Missed	Only	Missed	Lines	Missed	Methods	Missed	Classes
com.sisu.cmpw272.prodwell.service	83%	83%	14	80	47	250	6	52	0	9		
com.sisu.cmpw272.prodwell.controller	77%	77%	3	24	18	77	3	24	1	13		
com.sisu.cmpw272.prodwell.entity	71%	71%	9	32	12	52	9	29	0	5		
com.sisu.cmpw272.prodwell.security	96%	96%	2	18	3	58	0	13	0	3		
com.sisu.cmpw272.prodwell.entity	100%	100%	1	20	0	30	0	18	0	3		
com.sisu.cmpw272.prodwell	n/a	n/a	0	2	0	3	0	2	0	1		
Total	312 of 1,940	83%	12 of 76	84%	29	176	80	470	18	138	1	34

Fig. 3. Backend testing code coverage

VII. API SECURITY

To ensure secure access to the AI-Driven Employee Wellness and Productivity Tracker, the system employs MSAL (Microsoft Authentication Library) for authentication and authorization. MSAL integrates seamlessly with Azure Active Directory (Azure AD), providing a robust solution for securing APIs and controlling access to sensitive employee data. The following are the key aspects of the API security strategy:

A. Authentication with MSAL

MSAL is used to authenticate users by integrating the system with Azure AD B2C. This allows the application to securely handle sign-ins, providing users with a seamless authentication experience. MSAL enables the application to authenticate users with a variety of identity providers, including Microsoft accounts, work, and school accounts, as well as custom identity providers.

- **Secure Token Exchange:** Once a user is authenticated, MSAL issues an OAuth 2.0 token that is passed to the backend API. The backend uses this token to verify the identity of the user and ensure that the request is coming from a valid, authenticated source.
- **Access and Refresh Tokens:** The application uses access tokens to authorize API requests. When the access token expires, the system uses a refresh token to obtain a new access token, ensuring that the user’s session remains secure and continuous without requiring them to sign in repeatedly.

B. Authorization

After successful authentication, MSAL helps the backend validate the user’s authorization by checking the user’s roles and permissions in Azure AD. Based on the role assigned to the user, specific resources and endpoints are made accessible:

- **Role-Based Access Control (RBAC):** The system uses Azure AD’s RBAC to assign roles to users, such as employees, HR teams, and managers. Based on these roles, the application determines the level of access to various features and data within the system. For example, employees can view their own wellbeing and productivity data, while HR teams and managers may access broader insights across teams

C. API Security Best Practices

- **Token Validation:** The backend ensures that each API request contains a valid token issued by Azure AD. Tokens are validated using the Microsoft identity platform's libraries, ensuring the integrity and authenticity of the token.
- **Encryption:** All communication between the frontend, backend, and Azure AD is encrypted using HTTPS, protecting data in transit from unauthorized access.
- **Token Expiry and Rotation:** Access tokens are designed to expire after a short duration to limit the potential impact of a compromised token. Refresh tokens are rotated regularly to further enhance security.

D. Logging and Monitoring

To ensure compliance and detect any potential security breaches, Azure AD and MSAL provide built-in logging and monitoring capabilities. This allows the system to track authentication events, API access logs, and any suspicious activities, providing real-time alerts to administrators for any unusual patterns.

By utilizing MSAL in conjunction with Azure AD, the AI-Driven Employee Wellness and Productivity Tracker ensures that only authorized users can access sensitive data and perform critical actions within the system. This security model enhances trust and protects the integrity of the application and its data, ensuring a safe and secure user experience.

VIII. CONCLUSION

The AI-Driven Employee Wellness and Productivity Tracker is a sophisticated platform designed to monitor and improve employee mental wellbeing and productivity. With a robust System Architecture utilizing React for the frontend, Spring Boot for the backend, and MongoDB for data storage, the system integrates LLAMA 3.1 (Learning and Modeling Algorithm 3.1) for data analysis and personalized recommendations. The platform offers a variety of Functionalities, including stress level tracking, personalized wellbeing interventions, and productivity monitoring, ensuring employees receive tailored recommendations to enhance both their mental health and work performance. The system caters to various Use Cases, supporting employees, HR teams, and managers by providing insights into wellbeing and productivity trends, all while being built on secure, scalable technologies.

The Technologies Used in the project, such as React, Spring Boot, MongoDB, and Azure AD B2C, are key to the platform's functionality. The system underwent rigorous Testing using JUnit for backend and Jest for frontend, achieving high code coverage to ensure reliability and performance. In terms of API Security, the integration of MSAL (Microsoft Authentication Library) with Azure AD ensures secure authentication and authorization, protecting sensitive user data and maintaining system integrity. In conclusion, the AI-Driven Employee Wellness

and Productivity Tracker delivers a comprehensive, secure, and efficient solution for organizations to improve employee wellbeing and productivity through advanced technology and robust security practices.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to all those who have contributed to the successful completion of this project. Special thanks to Prof. Rakesh Ranjan for his continuous guidance, invaluable feedback, and support throughout the development process. His mentorship has been crucial in shaping this project and ensuring its successful execution.

We also acknowledge the valuable suggestions from our peers, which have significantly enhanced the quality of this work. Our appreciation extends to the developers and engineers behind the technologies used in this project, including React, Spring Boot, MongoDB, Azure AI Services, and MSAL, whose platforms and tools have been fundamental in building this system.

Lastly, we thank the organizations and individuals who have supported our work, providing the resources and environment to pursue this research and development.