Employee Activity Tracker

Project Overview

Description

The Employee Activity Tracker is a software solution designed to monitor and analyze employee activities while ensuring their privacy. It tracks various metrics such as window and tab switching, mouse click movements, screen time, and location. The software includes an interactive UI for employees, with features like leave application, performance details, and warnings for irrelevant activities.

Goals

- Monitor and analyze employee activities effectively.

- Provide a user-friendly interface for employees.

- Enhance productivity and accountability.

- Ensure privacy and ethical use of monitoring data.

- Track employee locations.

- Display performance details and warnings for irrelevant activities.

- Integrate with the cloud for scalability.

Project Plan

Milestones

1. Initial Setup and Requirements Gathering

2. Design and Architecture

3. Core Feature Development

4. UI/UX Design and Development

5. Leave Application Feature

6. Performance Details and Irrelevant Activity Warnings

7. Integration and Testing

8. Cloud Integration

Requirements

Functional Requirements

- Employee Data Analysis: Collect and analyze data on employee activities.

- Window/Tab Switching Tracking: Track switches between different windows and browser tabs.

- Mouse Click Movements: Monitor mouse click movements and interactions.

- Screen Time Monitoring: Measure the time spent on the screen.

- Location Tracking: Track employee locations using a free API such as OpenCage Geocoding API.

- UI for Employee Interaction: Develop a user-friendly interface for employees to interact with the software.

- Leave Application Feature: Allow employees to apply for leave when performing tasks away from their PC.

- Performance Details: Display performance metrics to employees.

- Irrelevant Activity Warnings: Show warnings to employees when they attempt irrelevant activities.

Non-Functional Requirements

- Performance: Run efficiently without significant impact on system performance.

- Security: Ensure data privacy and secure handling of employee information.

- Usability: The UI should be intuitive and easy to use.

- Scalability: Handle multiple employees and large data volumes through cloud integration.

- Privacy: Protect employees' private data and handle it ethically.

Technology Stack

Frontend

- React.js: For building the interactive user interface.

- Redux: For state management in the frontend.

Backend

- Django: For server-side logic and handling requests.

- Django Rest Framework: For creating RESTful APIs.

Database

- PostgreSQL: For storing activity data and user information.

Tracking and Monitoring

- Python: For backend processing and data collection.

- JavaScript: For capturing client-side events such as mouse clicks and tab/window changes.

- Electron.js: For building a desktop application that can track window and tab switching.

- PyAutoGUI: For capturing mouse movements and clicks in a Python-based desktop application.

Location Services

- OpenCage Geocoding API: For tracking and managing employee location data.

Analytics

- Python with Pandas: For data analysis and processing.

Authentication

- JWT (JSON Web Tokens): For secure user authentication.

Cloud Services

- AWS (Amazon Web Services): For cloud integration, including hosting, storage, and scaling.

Additional Tools

- Celery: For handling background tasks and periodic tasks.

- Redis: For task queuing and caching with Celery.

Installation and Setup

Prerequisites

- Python: For running the Django backend.

- Node.js: For running the frontend server.

- PostgreSQL: For database storage.

- Git: For version control.