EngageWise Project Plan

**Table of Contents**

1. [**Project Overview**](#mxtq32w09pez)
2. [**Computing Infrastructure**](#chwpoodg7w89)
3. [**Security, Privacy, and Ethics**](#5p9zgjf368zx) **(Trustworthiness)**
4. [**Human-Computer Interaction (HCI**](#5qs4bpn021ui)**)**
5. **Risk Management Strategies**
6. **Data Collection Management and Report**
7. **Model Development and Evaluation**
8. **Deployment and Testing Management Plan**
9. **Evaluation, Monitoring and Maintenance Plan**
10. [**GitHub Repository Link**](#7ih8zu3b3rnb)

# Project Overview

## Project Title:

“EngageWise”

~ Real-Time Focus Monitoring and Productivity Boosting System

## Project Overview:

* **Objective:**EngageWise aims to help users stay focused while working or studying by monitoring their attention in real-time, detecting distractions, and providing feedback and warning bells to improve productivity. It also ensures user data privacy as it runs locally and offers long-term performance tracking.
* **Scope:  
  What the AI system will do:**
  + Monitor facial movements, eye position, and blinks to assess focus and attention.
  + Provide real-time feedback and analysis on user attention and distraction patterns.
  + Alert users when attention wanes to help them regain focus.

**Data it will use:**

* + Real-time video data of the user’s face.
  + Eye movements, blinks, and facial expressions as indicators of focus and attention.
  + Detection of objects and emotions.

**Limitations:**

* + It might require clear facial visibility for accurate results (poor lighting or camera angles may impact accuracy).
  + It might only emit a sound, which may not always be enough to refocus people on their work.
  + It does not monitor on-screen activities or brainwave patterns (EEG).
  + It might struggle to account for natural distractions unrelated to focus.
* **AI Techniques and Tools:**We are planning to use Python, specifically OpenCV to do the project. Existing Drowsiness Detection and Object Detection techniques might be leveraged. Furthermore, we might use HTML, CSS, JS / React.js along with AJAX for frontend.

## Stakeholders:

* **Project Team:  
  1. AI Scientists:**
  + Research and design advanced algorithms for attention and focus detection based on facial and eye movements.
  + Test and fine-tune models using large datasets to ensure accuracy in various conditions.
  + Ensure the AI adheres to ethical guidelines, particularly around privacy and data security.

**2. AI Engineers:**

* + Develop and implement the AI models (e.g., facial recognition, eye tracking) using tools like OpenCV and TensorFlow.
  + Optimize the system for real-time processing and ensure it integrates smoothly into the app.
  + Collaborate with software developers to deploy the AI models into a user-friendly interface.

**3. Frontend Developer:**

* + Design and build a user-friendly interface for EngageWise, ensuring smooth interaction with features like real-time feedback and detailed reports.
  + Integrate visual elements like attention heatmaps and reports into the application, making them easy to interpret.
  + Ensure responsiveness and cross-platform compatibility, whether on web or mobile.

**4. DevOps Engineer:**

* + Set up and manage the deployment pipeline, ensuring smooth and continuous integration of updates and features.
  + Monitor server performance and optimize the infrastructure to handle real-time data processing efficiently.
  + Ensure system scalability, reliability, and security, particularly around handling user data and privacy.
* **End Users:**The main users are **students** and **working professionals** who regularly use computers for their daily tasks. They interact with EngageWise by starting a session, during which the system monitors their focus and attention by running in the background, provides alerts in the form of alarms, sounds and pop-ups when distractions occur, and generates a report at the end to help them understand their concentration levels.
* **Other Stakeholders:** None as of now.

**Team Members:**

* Ankith Reddy (AI Scientist and DevOps Engineer)
* Shreejit Cheela (AI Engineer and Frontend Developer)

# Computing Infrastructure

This section details the computational infrastructure required to support the development and deployment of the EngageWise platform, which monitors user attention in real-time and provides feedback. The infrastructure has been planned with the use of **OpenCV** for real-time attention monitoring and either **Django** or **Flask** for managing API requests and responses.

### 1. Project Needs Assessment

The EngageWise system requires a robust infrastructure capable of supporting the following:

* **Real-time processing**: EngageWise uses OpenCV to capture and analyze video feed for detecting eye movements, blinking, and yawning, which are crucial for monitoring user attention.
* **Efficient request handling**: Either Django or Flask will be used as the framework for managing server-side operations like handling user sessions, receiving real-time data from client devices, and providing feedback based on analyzed metrics.
* **Data storage**: The system will store historical data for each session, including reports on user focus patterns, which must be securely stored and retrieved for generating future insights.
* **Scalability**: EngageWise should be capable of handling varying user loads, from individual users to large groups of professionals or students using the platform simultaneously.
* **User interface**: The front-end requires a smooth, responsive interface where users can interact with the system in real time and receive feedback.

### 2. Hardware Requirements Planning

To ensure smooth operation and support real-time monitoring, EngageWise requires both **client-side hardware** for end-users and **server-side infrastructure** for backend operations.

* **Client-side hardware**:
  + **Webcam**: A 720p or higher resolution camera for tracking eye movements, blinking, and yawning via OpenCV.
  + **Processing Power**: At least a multi-core processor (2.5 GHz+) with 8GB RAM for client devices to ensure smooth performance of video capturing and data transmission.
  + **Internet Connectivity**: Stable, high-speed internet is essential for uploading live video streams to the server and receiving real-time feedback.
* **Server-side infrastructure**:
  + **CPUs**: Multi-core processors (Xeon or equivalent) are required to handle simultaneous user sessions and real-time data analysis.
  + **RAM**: At least 32GB of RAM is necessary to support real-time processing and efficient task handling for multiple users.
  + **Storage**: Fast SSDs for data processing and larger HDDs for long-term storage of user session data and reports.
  + **Network Bandwidth**: High network bandwidth is crucial to handle the transfer of video streams and real-time alerts, especially for concurrent users.

### 3. Software Environment Planning

The software stack is designed to efficiently handle the core functions of EngageWise—video processing, data transmission, and real-time feedback delivery. The system will use a combination of:

* **Operating Systems**:
  + The platform will support multiple operating systems, including Windows, macOS, and Linux, ensuring compatibility for a broad user base.
* **Backend Framework**:
  + **Django or Flask**: Django is preferred for its built-in features like authentication and security, while Flask offers flexibility for smaller, simpler applications. Either framework will be used for handling user requests and managing sessions.
  + **RESTful APIs** will be set up in Django/Flask to communicate between the client and the server, enabling the real-time transmission of attention data and feedback.
* **Real-time Video Processing**:
  + **OpenCV**: This library will handle video stream capture and real-time image processing on the client-side, analyzing metrics like eye movement and blinking patterns.
* **Database**:
  + A relational database, such as **PostgreSQL**, will be used to store user session data, reports, and configurations. The database should also support future scalability as more users are onboarded.
* **Machine Learning Libraries**:
  + EngageWise will rely on **TensorFlow** or **scikit-learn** for building models that detect attention patterns and provide real-time feedback.

### 4. Cloud Resources Planning

Given the nature of the project, cloud resources will be crucial for hosting and scaling the EngageWise platform.

* **Compute Resources**:
  + EngageWise will leverage cloud-based virtual machines (such as **AWS EC2**, **Google Cloud Compute Engine**, or **Azure Virtual Machines**) to handle real-time video processing and request management.
  + These virtual machines will host both the web application (running Django/Flask) and the video analysis pipeline using OpenCV.
* **Storage**:
  + **Cloud storage solutions** (like **AWS S3**, **Azure Blob Storage**, or **Google Cloud Storage**) will be used to store user data, session reports, and historical data for future reference. These storage services will ensure secure, scalable, and redundant data handling.
* **Content Delivery Network (CDN)**:
  + To reduce latency and ensure smooth delivery of feedback and reports, **CDNs** like **CloudFront** or **Google CDN** will be used to distribute static and dynamic content globally.
* **AI/ML Services**:
  + The cloud platforms offer specialized machine learning services (e.g., **AWS SageMaker**, **Google AI Platform**) that will help in training, deploying, and scaling AI models for attention prediction without overburdening local infrastructure.

### 5. Scalability and Performance Planning

EngageWise’s architecture needs to be flexible enough to scale both vertically and horizontally as the number of users grows.

* **Horizontal Scaling**:
  + Services like **Kubernetes** or **Docker Swarm** will be used to manage containerized applications. This ensures that as user demand increases, additional containers (e.g., handling OpenCV video streams) can be spun up on demand.
* **Vertical Scaling**:
  + Depending on demand, the compute power (CPU, RAM) of individual servers can be increased to handle larger workloads. This ensures real-time video processing remains uninterrupted, even as more users join.
* **Load Balancing**:
  + Using **AWS Elastic Load Balancing** or **Google Cloud Load Balancer**, incoming traffic will be distributed across multiple servers to prevent any single server from being overwhelmed, ensuring smooth operations and high availability.
* **Monitoring and Performance Optimization**:
  + Tools like **AWS CloudWatch**, **Google Cloud Monitoring**, or **Datadog** will be used to monitor server performance, detect bottlenecks, and trigger auto-scaling when needed.
  + **Caching** solutions such as **Redis** will be implemented to reduce database queries and speed up response times.
* **Data Backup and Redundancy**:
  + EngageWise will implement daily backups of user session data, ensuring data is secure and recoverable in case of any system failure. **Multi-region redundancy** will also be employed to ensure the application can recover from outages.

By carefully considering the above infrastructure, EngageWise will be built to handle the real-time requirements of user attention monitoring, while maintaining flexibility to scale as user demand increases. This ensures the platform remains reliable, efficient, and future-proof.

# Security, Privacy, and Ethics (Trustworthiness)

In designing EngageWise, a focus and productivity tool that uses AI to monitor user attention, it’s crucial to ensure that trustworthiness is embedded throughout the AI lifecycle. Each stage of development—from problem definition to ongoing monitoring—requires strategies that address security, privacy, and ethics to protect users and build a reliable system. The following section outlines trustworthiness strategies for each stage, with a particular emphasis on technical implementations.

### 1. Problem Definition

At the outset, it’s important to clearly define the AI system’s goals and ensure that ethical considerations are embedded in the purpose of the system.

* **Strategy: Ethical Impact Assessment (EIA)**
  + Before any technical development, conduct an **Ethical Impact Assessment** to evaluate the potential societal and individual impact of EngageWise. This will involve identifying stakeholders, understanding potential risks (such as misuse of attention-tracking data), and ensuring the system aligns with ethical AI principles (e.g., fairness, non-discrimination, and user well-being).
  + **Example**: Consider how the system might unfairly affect users with disabilities or neurodiverse individuals, ensuring that the system does not unjustly penalize users for involuntary behaviors.

**Output**: A clearly defined problem with a commitment to fairness and inclusion built into the system’s core objectives.

### 2. Data Collection

Data collection presents significant privacy and security risks, particularly when dealing with sensitive user information like video streams or behavioral data.

* **Strategy: Privacy-by-Design & Differential Privacy (Technical Implementation)**
  + **Privacy-by-Design** will be embedded into the system, meaning that privacy safeguards are implemented from the start. For instance, the **OpenCV** system will be designed to process video data locally on the user’s device where possible, reducing the need to transmit raw video data to a central server.
  + **Differential Privacy** can be integrated to anonymize data. By adding noise to user session data before storage, individual user behaviors (like blinking or yawning) cannot be traced back to specific users. Libraries such as **Google’s Differential Privacy** can be utilized to implement this.
  + **Example**: Only aggregated metrics, such as the average time to first distraction, will be uploaded to the server, ensuring individual sessions remain private.

**Output**: A secure and privacy-conscious data collection process that limits exposure to sensitive user data.

### 3. Model Development

AI models must be built with transparency and accountability, ensuring that the decision-making process is fair, explainable, and non-discriminatory.

* **Strategy: Fairness Auditing with AI Fairness 360 (Technical Implementation)**
  + During model development, use tools like **IBM’s AI Fairness 360** library to evaluate the models for potential biases. This involves testing the system to ensure it does not exhibit biased behavior based on race, gender, or other protected attributes. For example, the model should not overestimate the level of distraction for individuals from specific demographic groups.
  + Additionally, transparency should be maintained throughout the model development process. **Explainable AI (XAI)** methods can be implemented to provide clarity on how decisions (e.g., distraction detection) are made.
  + **Example**: Regularly test the attention model for demographic bias, ensuring that it responds equally well across different user groups, and explain why the system prompts a user to refocus.

**Output**: A fair and explainable AI model that promotes trust among users and ensures ethical decision-making.

### 4. Deployment

The deployment phase introduces challenges in protecting the AI system from external threats and ensuring the integrity of user data.

* **Strategy: End-to-End Encryption and Secure API Management**
  + All communication between the client (user) and the server must be secured using **end-to-end encryption** to prevent interception of sensitive data, such as video feeds or user attention metrics. Libraries like **SSL/TLS** encryption protocols can be used to secure data in transit.
  + **Authentication and Authorization**: Use **OAuth 2.0** or similar protocols to securely manage user access to the system, ensuring that only authorized individuals can view or modify data.
  + **Example**: Every API request made via Django/Flask must be securely authenticated using access tokens, and encryption should be enforced on both server-client communications and database storage.

**Output**: A secure deployment environment that guarantees the confidentiality and integrity of sensitive data throughout system use.

### 5. Monitoring and Maintenance

Once deployed, the system must be continuously monitored to ensure it remains secure, ethical, and trustworthy. New data may introduce unforeseen biases, or external threats may target the system’s operations.

* **Strategy: Continuous Bias Monitoring and Threat Detection**
  + **Bias Monitoring**: Post-deployment, set up automated monitoring systems to continuously assess the model for potential drifts or biases over time. This will ensure that the system remains fair and doesn’t evolve in a way that discriminates against certain user groups. Use tools like **Fairlearn** to conduct ongoing bias evaluations.
  + **Security Auditing**: Implement **Intrusion Detection Systems (IDS)** to detect any unauthorized access or malicious activity targeting the backend infrastructure (e.g., trying to extract personal data from the system). Regular security audits and vulnerability scanning should be performed to keep the system secure from external threats.
  + **Example**: EngageWise will run scheduled fairness tests on the model’s outputs and perform regular security audits using tools like **Snort** for threat detection.

**Output**: A continuously maintained and trustworthy system that adapts to new data while ensuring fairness and security.

# Human-Computer Interaction (HCI)

## **1. Understanding User Requirements**

### **Strategy: User Journey Mapping and Data-Driven Analysis**

Rather than relying solely on surveys or interviews, a powerful approach is to map out user journeys. Imagine creating a visual representation of the steps users take when interacting with similar tools—like focus apps or productivity tools.

**For example:** take a student using a focus tool during a study session. You’d map out the process from launching the app to receiving feedback and reviewing the session report. This journey reveals important pain points users face at different stages of their interaction.

#### **Steps:**

* **Identify touchpoints:** Find key moments where users engage with the system, like logging in, starting a session, or receiving alerts.
* **Analyze pain points:** Spot areas where users struggle, like confusion with alerts or complex reports.
* **Identify opportunities:** Use this analysis to highlight areas where you can improve the experience—maybe by offering customizable alert types or simplifying the feedback reports.
* In parallel, **data-driven analysis** taps into user behavior from similar apps (think Pomodoro timers or time tracking tools). By studying this data, you can uncover patterns that lead to distractions or prolonged focus. Tools like Mixpanel or Heap Analytics can give a clear picture of how users manage their attention in real-time.  
  **Example Metrics to Track:**
  + Time spent focused before distractions.
  + Triggers for distractions (like app switching or notifications).
  + Duration of productive sessions.

**Output:** Combining user journey mapping with real-world data helps ensure that EngageWise is finely tuned to users' natural workflows and addresses their pain points, leading to a system that better aligns with their focus and productivity needs.

## **2. Creating Personas and Scenarios**

### **Strategy: Persona Development and Scenario Mapping**

Personas, as rightly stated in the class, breathe life into the design process and understanding user requirements.

In my view, there are three types of users of **EngageWise**

* **Persona 1: Rachel (Student)**
  + **Description:** Rachel, a 22-year-old college student, is preparing for exams.
  + **Challenges:** She’s easily distracted by social media and frequently switches between study materials and entertainment apps.
  + **Goals:** To stay focused for long periods without distractions.
  + **Scenario:** During a 3-hour study session, EngageWise tracks Rachel’s eye movements, drowsiness, yawns and mobile usage. When the app detects an increase in any aspect, it sends her a prompt to refocus.
* **Persona 2: John (Professional)**
  + **Description:** John is a 35-year-old software developer working on complex tasks.
  + **Challenges:** After extended periods of work, he loses focus and productivity.
  + **Goals:** Maximize deep work periods without distractions.
  + **Scenario:** EngageWise helps John through a 4-hour coding session by monitoring his focus. It offers real-time feedback, like suggesting breaks when signs of fatigue (yawning or blinking) appear.
* **Persona 3: Sophia (Teacher)**
  + **Description:** Sophia is a 45-year-old high school teacher who spends long hours grading.
  + **Challenges:** Fatigue sets in after grading for hours, leading to a loss of focus.
  + **Goals:** Improve productivity and avoid drowsiness during long grading sessions.
  + **Scenario:** EngageWise tracks Sophia’s attention while grading and helps her to refocus with her favorite playlist when signs of drowsiness appear.

**Output:** These personas and their respective scenarios will guide EngageWise’s design to ensure it meets the diverse needs of its users.

## **3. Conducting Task Analysis**

### **Strategy: Hierarchical Task Analysis (HTA)**

EngageWise needs to make the process of setting up and running a focus session smooth and intuitive. At the same time, it should provide meaningful real-time feedback and detailed post-session reports.

#### **Main Task: Using EngageWise for a focus session.**

* **Initiate Session:**
  + User logs into EngageWise.
  + Selects session duration (1, 2, or 3 hours).
  + EngageWise starts tracking metrics like eye movement, blinking, and yawning.
* **Monitor Attention:**
  + EngageWise watches user behavior and sends real-time feedback.
  + Alerts are sent when distraction or drowsiness is detected.
  + Metrics like focus duration and blink/yawn frequency are tracked.
* **End Session:**
  + Once the session ends, a detailed report is generated.
  + This report highlights trends in focus, periods of low attention, and suggestions for improvement.

**Potential Issues:** The HTA can highlight bottlenecks, like delays in alerting users or difficulties interpreting real-time feedback. Optimizing how feedback is delivered (through visual and auditory cues) could resolve these issues.

**Output:** Task analysis ensures that EngageWise is designed for smooth navigation and effective feedback at every step.

## **4. Identifying Accessibility Requirements Strategy: WCAG Compliance for Diverse Users**

To ensure inclusivity, EngageWise must follow the Web Content Accessibility Guidelines (WCAG).

* **Screen Reader Compatibility:** Use ARIA attributes to label interface elements, making them navigable for visually impaired users. For instance, a “Start Session” button would have an ARIA label for clarity.
* **Keyboard Navigation:** Ensure that every feature of the app can be accessed with just a keyboard. Assign keyboard shortcuts like pressing "S" to start a session and "R" to review reports.
* **High Contrast Design:** Follow WCAG’s recommended 4.5:1 contrast ratio to make text and background colors accessible to visually impaired users.
* **Auditory Cues:** Provide auditory feedback alongside visual cues, so users with visual impairments can still engage with notifications and alerts.
* **Testing:** Use tools like WAVE and Axe to regularly test accessibility compliance.

**Output:** With full WCAG compliance, EngageWise will be accessible to a wide range of users, including those with disabilities.

## **5. Outlining Usability Goals**

### **Strategy: SMART Usability Goals**

Defining clear, measurable usability goals will ensure EngageWise is effective, efficient, and satisfying to use.

* **Goal 1: Improve Focus by 30-50%**
  + **Specific:** Reduce distractions by sending timely alerts.
  + **Measurable:** Compare user focus levels before and after using EngageWise.
  + **Achievable:** Based on user feedback and testing, refining the alert system should make this possible.
  + **Relevant:** Enhancing focus is a core objective of EngageWise.
  + **Time-bound:** Target this improvement within 6 months of launch.
* **Goal 2: Session Setup Time Under 2 Minutes**
  + **Specific:** Make sure users can configure and start a session in less than 2 minutes.
  + **Measurable:** Track time to log in, set session length, and begin tracking.
  + **Achievable:** Streamlining the UI/UX can help reach this goal.
  + **Relevant:** Fast setup is key for regular user adoption.
  + **Time-bound:** Achieve this goal within 3 months of launch.
* **Goal 3: Achieve 85% User Satisfaction**
  + **Specific:** Ensure at least 85% of users are satisfied with EngageWise’s ease of use, feedback, and reports.
  + **Measurable:** Use post-session surveys to measure satisfaction.
  + **Achievable:** Ongoing improvements based on feedback will drive this.
  + **Relevant:** High satisfaction is vital for user retention.
  + **Time-bound:** Reach this within 6 months of launch.

**Output:** These goals will be monitored through analytics and feedback to ensure continuous improvement in user experience.

# Risk Management Strategies

Risk management is crucial to protect user data and maintain system integrity, building trust and compliance with data protection standards. It helps prevent security breaches, ensuring the safe, reliable use of EngageWise. EngageWise plans to prioritizes Privacy-by-Design by processing real-time video data locally, minimizing the need for sensitive data transfer. Utilizing the concept of Differential privacy helps in anonymizing session data, so only aggregated insights, not individual behaviors, are stored.

## **1. Problem Definition**

In EngageWise, the primary goal is to accurately detect and track user engagement through facial expressions and micro-behaviors in real-time. Defining this problem carefully helps mitigate risks related to user privacy, potential misuse, and system misalignment with intended purposes.

### **Key Risks:**

* **Misalignment with Objectives**: Risk that EngageWise may not accurately measure engagement as desired by users.
* **Bias in Problem Framing:** Unintentional bias may be introduced in defining engagement, potentially leading to unfair or skewed results.
* **Ethical Concerns:** Risks related to potential privacy invasion due to real-time monitoring of users.

### **Mitigation Strategies:**

* Engage with **stakeholders** (students, educators, regulatory bodies) to confirm EngageWise’s objectives, features, and success metrics.
* Define metrics for engagement (e.g., blink rate, yawn frequency, etc.), specifying what constitutes "engaged" vs. "distracted" behavior.
* Conduct an **Ethical Impact Assessment (EIA)** prior to technical development of EngageWise to evaluate its societal and individual impacts, identify stakeholders, assess risks (like misuse of attention-tracking data), and ensure alignment with ethical AI principles such as fairness and user well-being, particularly considering the effects on users with disabilities or neurodiverse individuals.

### **Technical Mitigation Tools:**

* **Ethical Impact Assessment (EIA)**
* **Lucidchart** for mapping requirements and goals.

## **2. Data Collection**

EngageWise relies heavily on real-time visual data, which does not require external data. However, data is essential for mobile detection, demanding effective collection and management practices to prevent biased, poor-quality, or unrepresentative data that might affect model performance and fairness.

### **Key Risks:**

* **Data Quality**: Poor video quality can affect the accuracy of engagement detection.
* **Data Bias**: Datasets need to be diverse to ensure EngageWise accurately represents different demographics.
* **Data Privacy**: Real-time monitoring can raise privacy concerns, especially for sensitive populations.
* **Data Representativeness**: Without representative data, the model may not generalize well across users.

### **Mitigation Strategies:**

* **Data Quality Checks**: Pre-process images using tools like **OpenCV** (used in your code) for noise reduction and quality enhancement.
* **Diverse Data Collection**: Gather a diverse sample to capture a range of demographics, settings, and lighting conditions.
* **Bias Detection**: Apply **Fairlearn** or **AIF360** to assess and correct any demographic or behavioral biases in the collected data.

### **Technical Mitigation Tools:**

* Utilizing **Pandas** and **OpenCV** for automated data validation and preprocessing.
* Plan to use libraries like **AIF360** or **Fairlearn** for assessing and correcting data bias.

## **3. Model Development**

Building and training the EngageWise model requires attention to risks like overfitting, algorithmic bias, and mainly lack of explainability, which can impact performance and fairness.

### **Key Risks:**

* **Algorithmic Bias**: Potential for certain facial expressions or demographics to be misrepresented in engagement measurement.
* **Explainability**: Difficulty in explaining what factors contribute to engagement classifications.
* **Overfitting**: Risk of the model becoming overly tuned to the mobile phones data, reducing generalizability.

### **Mitigation Strategies:**

* **Bias Mitigation Techniques**: Use fairness-aware algorithms, such as **adversarial debiasing** or **reweighting** approaches.
* **Overcoming Overfitting**: Firstly, adding more data of mobile phones. Apply regularization to prevent overfitting, and use **cross-validation** for more reliable results.

### **Technical Mitigation Tools:**

* Cross-validation and sensitivity analysis reduce overfitting and improve model adaptability.
* Adjustments to alert thresholds ensure a balanced response to user behaviour.

## **4. Deployment**

In deploying EngageWise, maintaining system stability and trustworthiness is key. Deployment risks like environmental variation or system integration challenges could impact EngageWise’s effectiveness.

### **Key Risks:**

* **Integration with Systems**: Compatibility issues with video capture and data transfer protocols.
* **Cybersecurity Threats**: Risks of unauthorized access to real-time monitoring systems, potentially leading to data breaches.
* **Overwhelming CPU:** As EngageWise operates in the background of the user's PC, there is a potential risk of it overwhelming all CPU cores.

### **Mitigation Strategies:**

* **A/B Testing**: Test EngageWise in different environments and lighting conditions to assess system performance before full rollout.
* **Security Audits**: Regularly update and patch software vulnerabilities. Implement secure authentication to limit access.

### **Technical Mitigation Tools:**

* EngageWise utilizes the `cv2.VideoCapture` method to control the frame capture rate, ensuring it only processes frames at a defined interval to prevent excessive CPU usage.
* EngageWise employs the `cv2.waitKey(1)` function to introduce a slight delay between frames, which helps to reduce CPU load by preventing the program from consuming 100% of the CPU resources during continuous frame processing.

## **5. Monitoring and Maintenance**

Post-deployment monitoring is critical to ensure EngageWise continues to operate as intended and remains aligned with evolving user needs and risks.

### **Key Risks:**

* **Model Drift**: As user demographics or behaviours change, the model may become less accurate.
* **Emerging Security Threats**: New vulnerabilities can be exploited over time.

### **Mitigation Strategies:**

* **Drift Detection**: Monitor engagement metrics over time, using tools like **Evidently** or custom drift detection scripts.
* **Regular Retraining**: Periodically update the model with recent data to maintain accuracy and relevance.
* **Security Audits**: Schedule audits and updates to address emerging security threats.

### **Technical Mitigation Tools:**

* Yet to implement logic to skip frames based on a defined threshold (e.g., processing every nth frame) to reduce the frequency of resource-intensive computations.
* Adjust the frame rate dynamically based on system load and performance metrics, ensuring that processing can be throttled if CPU usage exceeds a certain threshold.
* Willing to use Python's threading module to run the background tasks in separate threads, allowing the main application to remain responsive and manage CPU load more effectively.
* Integrates checks for CPU and memory usage, allowing the application to adjust its processing load or halt non-essential tasks when resource usage is high.

## **6. Monitoring and Maintenance**

Post-deployment monitoring is critical to ensure EngageWise continues to operate as intended and remains aligned with evolving user needs and risks.

### **Key Risks:**

* **Model Drift**: As user demographics or behaviours change, the model may become less accurate.
* **Emerging Security Threats**: New vulnerabilities can be exploited over time.

### **Mitigation Strategies:**

* **Drift Detection**: Monitor engagement metrics over time, using tools like **Evidently** or custom drift detection scripts.
* **Regular Retraining**: Periodically update the model with recent data to maintain accuracy and relevance.
* **Security Audits**: Schedule audits and updates to address emerging security threats.

### **Technical Mitigation Tools:**

* Yet to implement logic to skip frames based on a defined threshold (e.g., processing every nth frame) to reduce the frequency of resource-intensive computations.
* Adjust the frame rate dynamically based on system load and performance metrics, ensuring that processing can be throttled if CPU usage exceeds a certain threshold.
* Willing to use Python's threading module to run the background tasks in separate threads, allowing the main application to remain responsive and manage CPU load more effectively.
* Integrates checks for CPU and memory usage, allowing the application to adjust its processing load or halt non-essential tasks when resource usage is high.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | Impact | | | |
| 0 Acceptable | 1 Tolerable | 2 Unacceptable | 3 Intolerable |
| Little or No Effect | Effects are felt but not clear | Serious Impact | Could be disastrous |
| Likelihood | Improvable | Unlikely to Occur | 1. Data representativeness concerns | 1. Misalignment with objectives | 1. Overwhelming CPU usage | 1. Cybersecurity threats leading to data breaches |
| Possible | Will Likely Occur |  | 1. Data quality affecting engagement detection | 1. Bias and Ethical concerns regarding real-time monitoring |  |
| Probable | Will Occur |  |  | 1. Data privacy issues |  |

## **Residual Risk Assessment**

# Data Collection Management and Report

## **1. Data Type**

The primary data type for EngageWise consists of visual data captured from users in real-time, focusing on facial expressions and micro-behaviours that indicate engagement levels.

* **Visual Data**: Video streams and images depicting users' facial expressions.
* **Behavioural Data:** Derived metrics such as blink rate, facial movement, and head orientation.
* **Environmental Data:** Information about the lighting conditions, camera angles, and other contextual factors during data collection.

## **2. Data Collection Methods**

The data collection methods utilized in EngageWise are designed to ensure high-quality, representative data while minimizing privacy concerns.

* **Video Capture**: Use of real-time video cameras equipped with appropriate software (e.g., OpenCV) to continuously monitor user engagement during sessions.
* **Surveys and Questionnaires**: Post-session surveys to gather qualitative feedback on user experience and perceived engagement levels.
* **User Consent**: Informed consent is obtained from users prior to data collection, ensuring they are aware of the data being collected and its intended use.

## **3. Compliance with legal frameworks**

EngageWise is committed to complying with relevant legal frameworks to protect user data and privacy.

* **General Data Protection Regulation (GDPR)**: Compliance includes obtaining explicit consent from users for data collection and usage, offering opt-out options, and ensuring users can access and delete their data.
* **California Consumer Privacy Act (CCPA)**: Adherence to requirements regarding user rights to know about data collection practices and their rights to request data deletion.
* **Institutional Policies**: Follow the guidelines set forth by the educational institution regarding data collection, storage, and handling of personal data.

## **4. Data Ownership**

Clarifying data ownership is essential for ethical management and compliance.

* **User Ownership**: Users maintain ownership of their data, including the right to withdraw consent and request data deletion.
* **Institutional Ownership**: The data collected is owned by the institution conducting the research, ensuring that data is used for the intended purposes only.
* **Shared Data Access**: Access to data for research and analysis is granted on a need-to-know basis, protecting user privacy and complying with legal requirements.

## **5. Metadata**

Metadata plays a critical role in data management by providing context and facilitating effective data usage.

* **Descriptive Metadata**: Information such as timestamps, user identifiers (not recorded), environmental conditions during capture, and the settings of the camera (resolution, frame rate).
* **Technical Metadata**: Details about file formats, storage locations, and data collection tools used.
* **Provenance Metadata**: Records of data lineage, including who collected the data, when it was collected, and any transformations applied.

## **6. Versioning**

Effective version control helps maintain the integrity and consistency of data.

* **Version Control System**: Use of version control software (e.g., Git) to manage code and model versions associated with data preprocessing, analysis, and model training.
* **Data Versioning**: Maintaining distinct versions of datasets, particularly after significant updates or modifications, to ensure reproducibility and traceability.
* **Change Logs**: Documentation of changes made to datasets and versions, including timestamps and descriptions of modifications.

## **7. Data Preprocessing, Augmentation, and Synthesis**

Preprocessing is crucial for ensuring high-quality data for model training.

* **Preprocessing:**
  + **Noise Reduction:** Use of techniques such as Gaussian blur or median filtering to enhance video quality.
  + **Face Detection:** Implementation of facial recognition algorithms (e.g., Haar Cascades or DNNs) to focus on relevant facial regions.
* **Augmentation:**
  + **Techniques:** Employing transformations (rotation, scaling, flipping) to increase dataset size and diversity of mobile phones.
  + **Synthesis:** Yet to be decided but generating synthetic data to supplement real data, ensuring a well-rounded dataset that can improve model robustness.

## **8. Report on Risk Management in Data Collection**

Effective risk management during data collection for EngageWise is essential to protect user privacy, ensure system integrity, and foster trust. This step identifies potential risks and outlines mitigation strategies to uphold compliance with privacy standards and system reliability.

Key Risks and Mitigation Strategies:

1. **Problem Definition Risks**
   * ***Key Risks*:** Misalignment with user objectives, potential biases in defining "engagement," and ethical concerns around real-time monitoring.
   * ***Mitigation Steps Implemented*:** EngageWise clearly aligns with stakeholder expectations by confirming objectives with students, educators, and regulatory bodies. Clear engagement metrics, such as blink rate, yawn frequency and drowsiness detection, are defined, distinguishing "awake" from "drowsy" behaviours.
   * ***Results:*** EngageWise aligns with stakeholder expectations, defines clear engagement metrics (blink rate, yawn frequency, drowsiness detection), and differentiates "awake" from "drowsy" behaviours.
2. **Data Collection and Quality Risks**
   * ***Key Risks:*** Poor data quality, lack of demographic diversity, privacy concerns due to real-time monitoring, and the need for data representativeness. Additionally, inaccurate or inconsistent data quality could impact engagement detection reliability.
   * ***Mitigation Steps Implemented:*** Data quality checks are applied on Mobile Phones Data using Python for noise reduction and quality enhancement, improving detection accuracy. Filtering out irrelevant or noisy data to avoid distracting or confusing the model, helping it focus only on the features that truly reflect user engagement. Diverse data is collected to ensure representation of different demographics and environments. Data is also pre-processed to ensure clarity and consistency, enhancing the reliability of engagement detection. Metadata about video quality and context is recorded to adjust analysis as needed.
   * ***Results:*** Data quality checks, noise reduction, and filtering improve detection accuracy, while diverse, pre-processed data ensures reliable engagement detection across demographics, with metadata recorded for analysis adjustments.
3. **Privacy and Security Risks**
   * ***Key Risks*:** Privacy concerns due to real-time video data capture, especially if sensitive data is mishandled or exposed.
   * ***Mitigation Steps Implemented*:** Privacy-by-Design practices are implemented, with local processing of real-time data to minimize sensitive data transfer. Anonymization or encryption of data is unnecessary due to local processing. Insights and data are not stored, and each session operates independently, as tracking user behaviour is not required to detect attentiveness in real-time.
   * ***Results:*** Privacy-by-Design practices ensure local processing of real-time data, avoiding sensitive data transfer, with no data storage or tracking required for attentiveness detection.
4. **Computer Overload:**
   * ***Key Risk:*** Running multiple real-time processes like facial landmark detection, distance calculation, and event handling in parallel can lead to high CPU usage, causing delays or system instability.
   * ***Mitigation Steps Implemented:*** A short delay is introduced using *cv2.waitKey(1)*, which helps in managing the frame rate and reducing the load on system resources by allowing brief pauses between each frame capture and analysis cycle. This limits CPU overload and optimizes performance.
   * ***Results:***A short delay using *cv2.waitKey(1)* manages the frame rate, reduces CPU load, and prevents system instability by optimizing performance during real-time processes.
5. **Cybersecurity Threats:**
   * ***Key Risk:*** Since the code runs locally, unauthorized access to the device or phishing bots could expose sensitive functionalities or in-process visual data.
   * ***Mitigation Steps Implemented:*** To enhance local security, password-based authentication requires users to verify their identity before accessing the program. This measure helps prevent unauthorized access, blocking phishing bots or automated scripts from running the code and ensures that only authorized individuals can initiate the program, providing a basic security layer against potential misuse.
   * ***Results:***Password-based authentication ensures only authorized users can access the program, blocking phishing bots or automated scripts and preventing unauthorized access to sensitive functionalities and in-process visual data.

## **9. Report on Trustworthiness in Data Collection**

Ensuring trustworthiness was essential throughout EngageWise’s AI lifecycle, focusing on privacy, fairness, and transparency from initial problem definition through to monitoring. Trustworthy AI in EngageWise centres on clear ethical commitments and proactive technical implementations.

1. **Problem Definition**  
   Ethical Impact Assessment (EIA) of the defined problem sets a clear, inclusive objective. It involves understanding potential misuse of attention data and ensures alignment with fairness and inclusion, prioritizing non-discriminatory practices during development.
2. **Data Collection**  
   Privacy-by-Design prioritized user privacy. EngageWise processes data locally using OpenCV and no data is stored, thereby zeroing the possibility of individual data exposure.
3. **Model Development**  
   Explainable AI (XAI) techniques, such as adding a delay and securing the code with password protection, ensure authorized access and provide transparency in distraction detection. These measures help users understand the model's decisions, reinforcing greater trust and accountability.
4. **Deployment**  
   Since there is no “data-in-transit” and user access is constrained to their personal computer, user interactions and data integrity are protected.
5. **Monitoring and Maintenance**  
   Continuous bias monitoring and security audits help EngageWise remain adaptive to new data, ensuring it remains secure, ethical, and fair throughout its usage.

### **Results:**

EngageWise ensured trustworthiness throughout its AI lifecycle by focusing on privacy, fairness, and transparency. Ethical Impact Assessments aligned objectives with fairness, while Privacy-by-Design practices safeguarded data. XAI techniques and password protection promoted transparency. Local data processing and continuous monitoring ensured data integrity, and security audits maintained the system's ethical and fair operation.

# Model Development and Evaluation

The EngageWise system integrates attention-tracking metrics to keep users focused by monitoring blink and yawn counts, detecting drowsiness, and identifying mobile phone usage. Each module uses computer vision to track different aspects of user attentiveness, and the addition of mobile phone detection will complete the suite of real-time attention-tracking functionalities.

## **1. Model Development**

1. **Algorithm Selection:**

* **Blink, Yawn, and Drowsiness Detection**: A dlib-based approach with facial landmark detection calculates Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR), providing blink and yawn counts.
* **Mobile Phone Detection (Planned)**: A YOLO-based CNN or similar architecture will be used for real-time detection of mobile phones, chosen for its efficiency in fast object detection.

1. **Feature Engineering and Selection:**

Feature extraction is primarily achieved through landmark detection:

* **EAR** and **MAR** measure specific face metrics relevant to blinks, yawns, and drowsiness.
* Mobile phone detection will involve training the CNN model to recognize unique spatial features of mobile phones within the visual scene, with additional context learned from hand position and orientation.

1. **Model Complexity and Architecture:**

We named the architecture: "**Mesh-Based Real-Time Attention Monitoring System (M-RAMS)**"

This name reflects the following aspects:

* **Mesh**: Highlights the use of facial mesh detection, specifically the detection of facial landmarks to calculate metrics like EAR and MAR, which are essential for assessing user states.
* **Real-Time**: Emphasizes the system's ability to process video feeds live.
* **Attention Monitoring**: Highlights the primary focus on tracking user attention through facial features.
* **System**: Indicates the integrated approach of various modules (face detection, feature extraction, and state assessment) working together.

## **2. Model Training**

1. **Training Process:**

* **Blink and Yawn**: Uses pre-trained dlib landmarks without additional training.
* **Mobile Phone Detection**: Custom Trained YOLO (CNN) model trained on a labelled dataset with various angles, lighting conditions, and different types of mobile devices.
* **Data Augmentation**: Rotate, flip, and scale mobile phone images to create a robust model.
* **Model Monitoring**: Track model accuracy and loss during training to detect overfitting or underfitting early.

1. **Hyperparameter Tuning:** Cross-validation will also be employed in the hyperparameter tuning process. Different hyperparameter settings will be evaluated using the cross-validation framework, and the combination that yields the best average performance metrics will be selected.

## **3. Model Evaluation:**

1. **Performance Metrics:**

* **Processing** **Time:**  
  Measure the time taken to process each frame. This metric helps assess how well the model performs in real-time. Using `time` module to calculate the processing time per frame.
* **Accuracy:**  
  In mobile phone detection, high accuracy indicates that the model is correctly identifying mobile phones and non-mobile phone images. However, it can be misleading if the dataset is imbalanced (e.g., many more non-phone images than phone images).
* **IoU (Insertion over Union):**  
  In object detection, especially in real-time applications, IoU helps assess how well the model is localizing mobile phones within the images. A higher IoU indicates a better fit between the predicted and actual bounding boxes.

1. **Cross-Validation:  
   K-Fold** Cross Validation technique employed during fine-tuning ensures a robust mobile phone detection model.

## **4. Implementing Trustworthiness and Risk Management in Model Development:**

1. **Risk Mitigation Strategies:**EngageWise addresses potential risks through clear alignment with stakeholder objectives, ensuring accurate and diverse data collection, and employing privacy-by-design principles. Real-time video processing is done locally, with no data stored, reducing privacy risks. System performance is optimized by introducing delays to manage CPU load and prevent overload. Password-based authentication protects against unauthorized access and phishing bots, securing sensitive functionalities and ensuring only authorized users can interact with the system.
2. **Trustworthiness:**EngageWise ensures trustworthiness by integrating ethical impact assessments in the problem definition, adhering to privacy-by-design principles, and processing data locally without storing it. Explainable AI techniques, such as secured access and transparency in decision-making, reinforce user confidence. Continuous monitoring for bias and regular security audits maintains ethical standards and system integrity, ensuring fairness and accountability throughout the AI lifecycle.

## **5. Apply HCI Principles in AI Model Development:**

1. **Interactive components include:**

* **Alert and Notification System**: Clear, non-intrusive nudges guide users back to work if distracted.
* **Display Transparency**: The interface shows current states (drowsy, awake) and real-time metrics (blinks, yawns), making the system’s decisions transparent.

1. **Design Transparent Interfaces:**Transparent design elements include real-time counters for blinks and yawns and the detected state (e.g., "Drowsy" or "Awake") on the screen. This transparency builds user trust and helps them understand system decisions.

# Deployment and Testing Management Plan

* + - 1. **Deployment Environment Selection:  
         Local deployment** is the best choice because it is simple, cost-effective, and does not require additional infrastructure. Running the application on a local system makes testing and debugging easier, especially with real-time webcam input. Once a stable version is achieved, cloud or edge deployment can be considered if scalability or faster performance is needed.
      2. **Deployment Strategy:**The deployment strategy involves setting up Flask to handle the core functionality of the application, such as processing webcam input and detecting drowsiness. The entire application, including Flask and dependencies like OpenCV, dlib, and pygame, is packaged into a Docker container. This container ensures consistent execution across different environments by isolating the application from the host system. The Docker container can then be deployed locally or on a cloud service, providing portability and ease of management. We hope, this strategy enables seamless testing, scalability, and future deployment to various platforms.
      3. **Security and Compliance in Deployment (Trustworthiness and Risk Management):**  
         Plan for security and compliance is to implement manual testing in a Docker deployment, the approach would involve reviewing the Docker file for adherence to security best practices, such as ensuring minimal base images and non-root user configurations. Next, inspect the environment variables to ensure sensitive data is properly encrypted and not exposed in plain text. Perform security checks by attempting to access restricted areas without authorization, testing for privilege escalation and access control vulnerabilities. Additionally, audit logs should be manually reviewed to ensure they capture and track all critical actions in the system. Finally, test for compliance by manually verifying that all necessary security policies and access controls are in place, and that data protection measures are properly implemented.

# Evaluation, Monitoring and Maintenance Plan

1. **System Evaluation and Monitoring:**For System Evaluation and Monitoring, continuous monitoring of EngageWise’s performance and behaviour will be essential. This will include tracking the accuracy of the focus and attention detection algorithms, assessing the responsiveness of the system, and monitoring system performance to identify any latency or errors. Real-time performance metrics will be collected and analysed to ensure the system functions as intended under various conditions, such as varying lighting and user activity levels.
2. **Feedback Collection and Continuous Improvement:**This will be a key part of the prototype's ongoing development. Users will be encouraged to provide feedback on their experience with EngageWise, specifically regarding usability and any issues with accuracy or performance. This feedback will be used to refine the system, ensuring that it meets the needs of the users. Regular updates will be pushed to the system, addressing any identified issues, and incorporating new features to enhance its functionality over time.
3. **Maintenance and Compliance Audits:**  
   Regular maintenance will involve checking for system updates, dependencies, and security patches. Additionally, compliance audits will be conducted periodically to ensure that the system continues to meet required standards and regulatory requirements. This will include reviewing access controls, privacy measures, and data security to ensure that EngageWise adheres to any relevant legal or industry-specific compliance frameworks.

# GitHub Repository Link

<https://github.com/Shreejit2401/AIS-Project_EngageWise>

\* \* \* \* \*