Task -3 Banao Ai

Procrastination Preventer

1. Introduction

The "Procrastination Preventer" is an automated system designed to monitor user activity, analyze browser content, and send notifications when non-work-related activities are detected. It uses OCR (Optical Character Recognition) and AI-based text classification to determine whether the user is engaged in productive tasks or being distracted by entertainment.

2. Objectives

- Monitor active browser tabs and detect content.
- Use AI models to classify activities as "work-related" or "not work-related."
- Send notifications only when non-work-related activities are detected.
- Log user activity and provide productivity insights.
- Ensure privacy by processing data locally.

3. System Workflow

1. **User Inputs Work Activities**: The system prompts users to define their expected work-related tasks.
2. **Screen Capture & Text Extraction**: The program captures active tabs and extracts text using OCR.
3. **Text Analysis & Classification**:
    - If the extracted text matches predefined work-related keywords, it is classified as "work-related."
    - If the text matches entertainment-related keywords, it is classified as "not work-related."
    - AI-based classification is used only as a fallback.
4. **Notification System**: The system alerts users only if they are engaged in distractions.
5. **Logging & Analytics**: Logs all activities for productivity tracking.

4. Technologies Used

- **Programming Language**: Python
- **Libraries**:
    - pyautogui: Screenshot capturing
    - pytesseract: OCR for text extraction
    - transformers: AI text classification

- plyer.notification: Notifications
- matplotlib: Productivity analysis

5. Code Explanation

Step 1: Import Required Libraries

The system imports various libraries to handle different functionalities:

- **Screenshot capturing** to track the active tab.
- **OCR (Optical Character Recognition)** to extract text from the tab.
- **AI-based classification** to detect work vs. non-work-related activities.
- **Notification system** to alert the user when distractions are detected.

Step 2: Configuration & User Input Handling

The system asks the user to input their expected work activities, which are stored and used for comparison during monitoring. The expected activities list contains both **work-related** and **non-work-related** keywords.

Step 3: Monitoring Active Browser Tabs

- The system continuously takes screenshots of the active browser tab.
- It extracts text from the screenshots using OCR.
- The extracted text is then analyzed to check whether it matches the user's intended work.

Step 4: Classification of Activities

- The extracted text is compared with predefined **work-related** and **non-work-related** keywords.
- If the extracted text contains work-related keywords (such as "programming," "Github," "GFG"), the system classifies the activity as **work-related**.
- If it contains non-work-related keywords (such as "Netflix," "YouTube," "movies"), it is classified as **not work-related**.
- If no clear match is found, the system uses an AI model to classify the text as a fallback.

Step 5: Sending Notifications

- The system **only sends notifications if the activity is non-work-related.**
- If the detected activity is a distraction, a notification is sent to the user asking them to **refocus on work**.
- If the activity is work-related, no notification is triggered to avoid unnecessary interruptions.

Step 6: Logging & Productivity Analysis

- Every detected activity (work-related or non-work-related) is logged in a local file.
- The logs are used to track productivity trends.
- Users can generate **productivity reports** to see how much time was spent working versus being distracted.

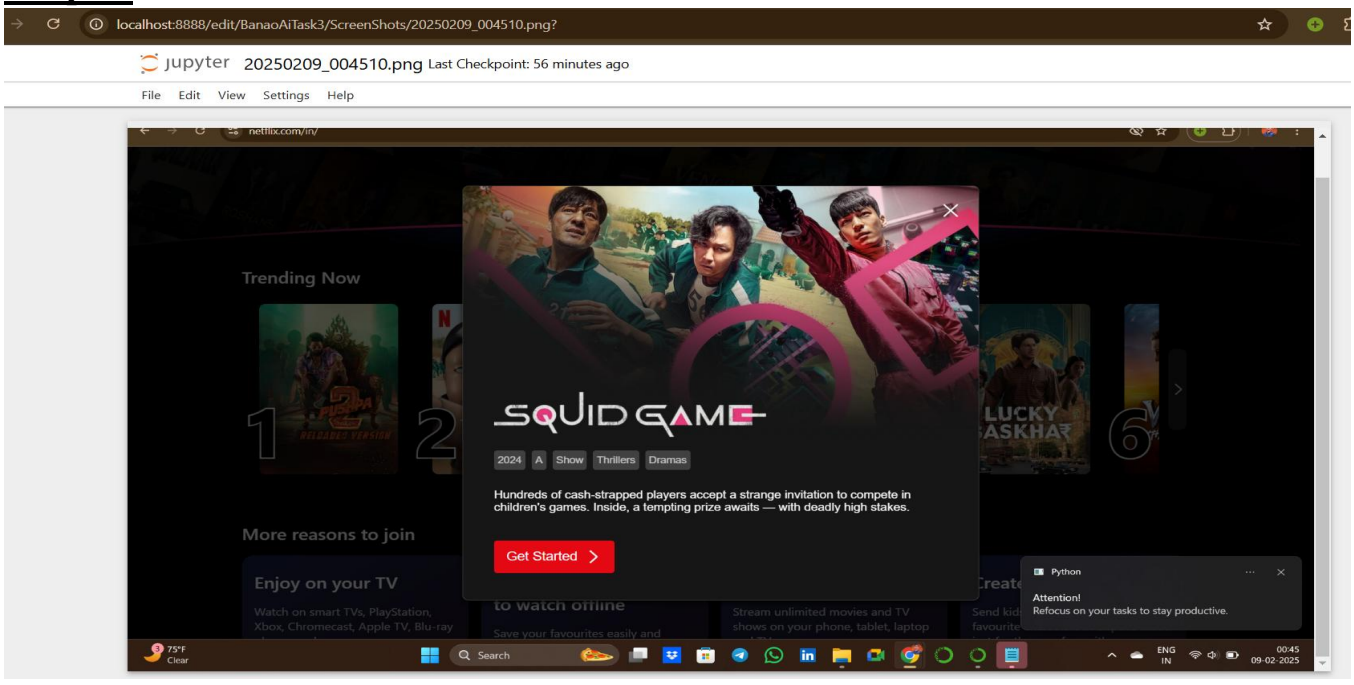
Step 7: Privacy & Security

- All data (screenshots, logs) are **processed locally** and never transmitted.
- Users have the option to **delete logs** and clear stored data at any time.
- The system ensures that no personal or sensitive information is stored permanently.

6. Testing & Validation

- The system was tested to ensure **notifications are only sent for distractions**.
- The **accuracy of text extraction and classification** was verified using different scenarios.
- Debugging logs were added to track how the system processes information and ensures correct classification.

Output:



7. Conclusion

The "Procrastination Preventer" effectively monitors and classifies user activity, ensuring users stay focused on work while limiting distractions. With privacy-preserving local processing and productivity analytics, it serves as a robust tool for improving time management and efficiency.