

Project Report: Python-based Desktop Agent Application

Team Members:

- Virender Kumar
- Vishwakarma Singh
- Vivek Kumar
- Wazida Tabbasum
- Yogesh Devtulla

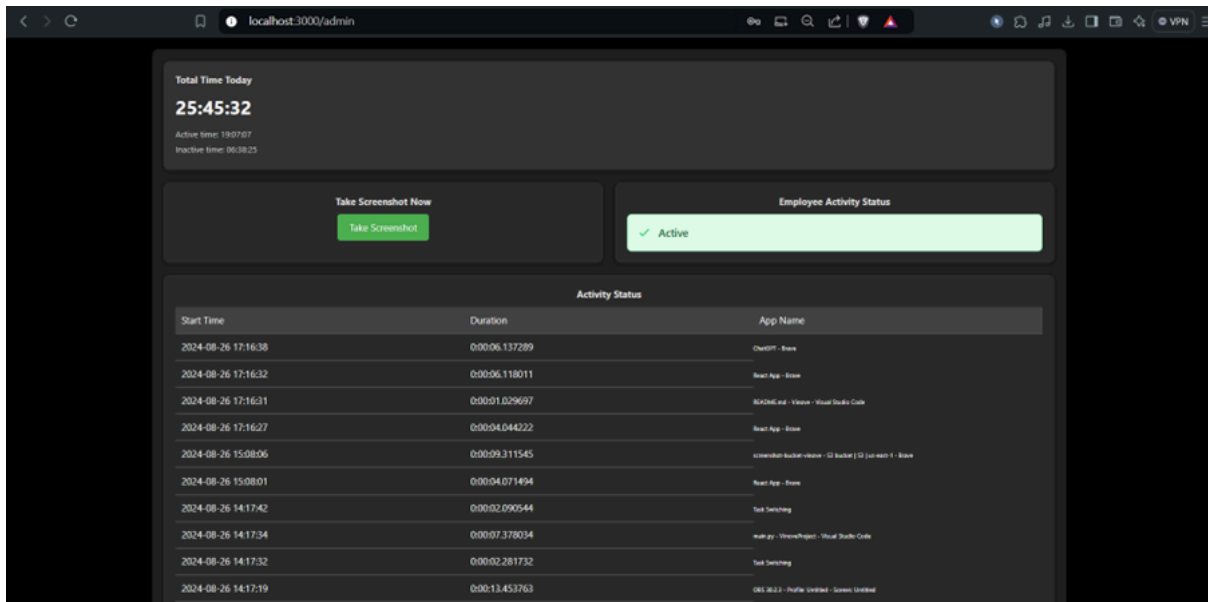
1. Introduction

This project involves developing a Python-based desktop agent application that tracks employee activity and securely uploads relevant data (e.g., screenshots) to Amazon S3 or a similar cloud storage service. The application is designed to monitor genuine user activity, handle configuration changes, and ensure that best practices in software development are adhered to.

2. Core Features

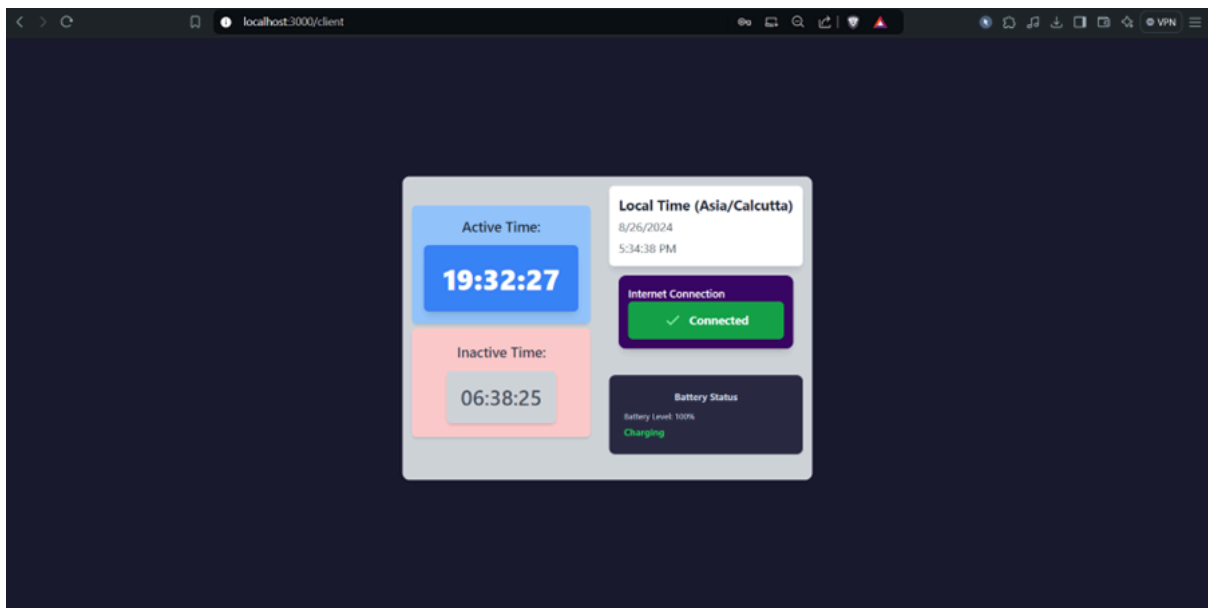
Activity Tracking:

The agent runs in the background, continuously monitoring and capturing user activity. It differentiates between genuine user input and scripted activity by detecting irregular patterns in mouse movements or keyboard inputs.



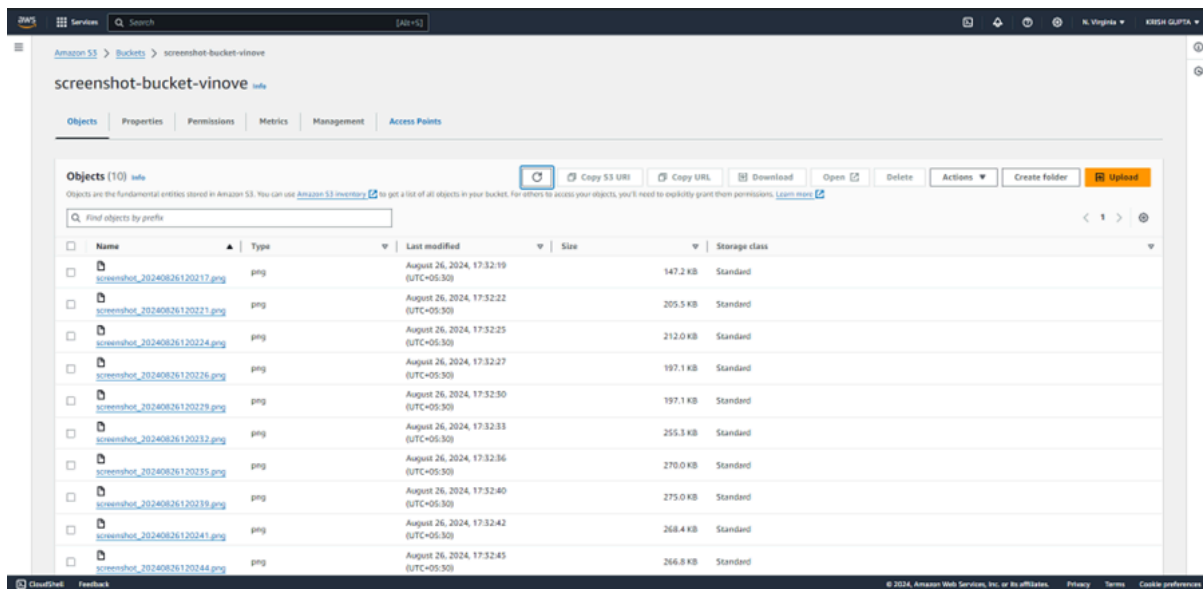
Time Zone Management:

The agent detects time zone changes on the user's system and adjusts timestamps in the activity logs accordingly. Time zone changes are captured in real-time and reflected accurately in the logs.



Data Upload:

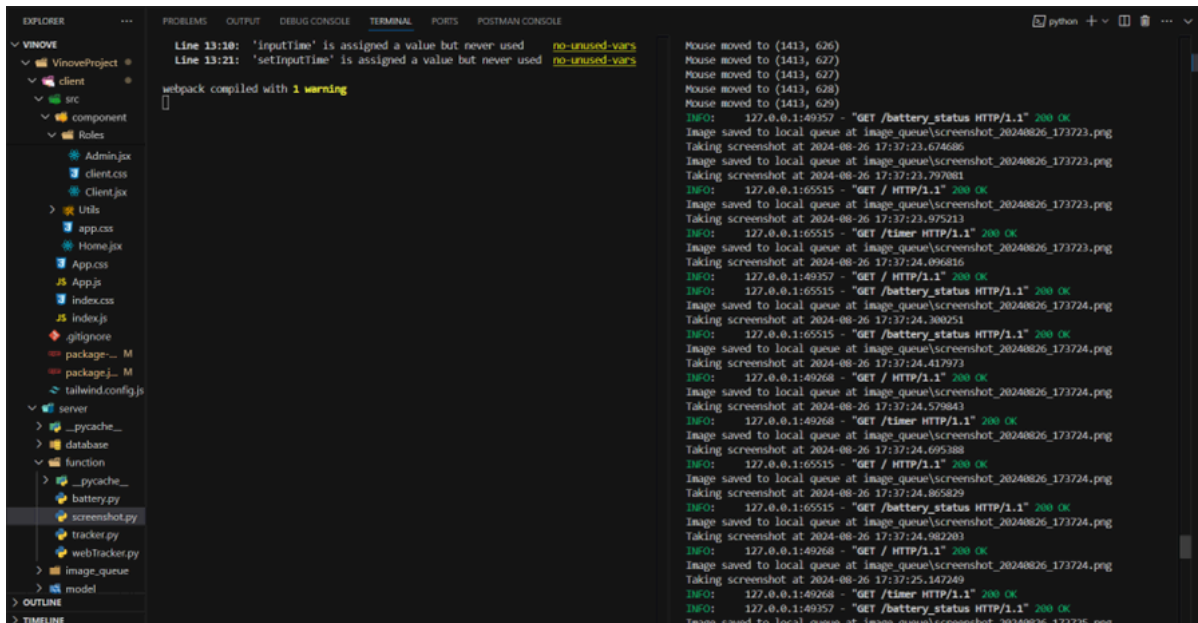
Captured screenshots and activity logs are automatically uploaded to Amazon S3 cloud storage service. The application implements mechanisms for efficient handling of large file uploads, such as chunked uploads or compression. Data is securely uploaded using encryption to ensure privacy and data integrity.



Error Handling and Resilience:

The application includes robust error handling for scenarios such as:

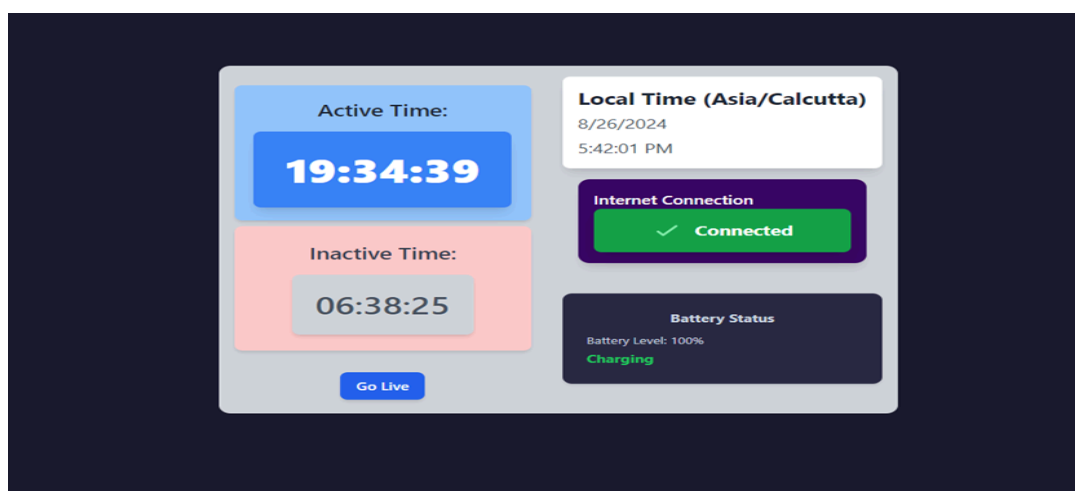
- **No Internet Connection:** Uploads are queued and retried when the connection is restored.

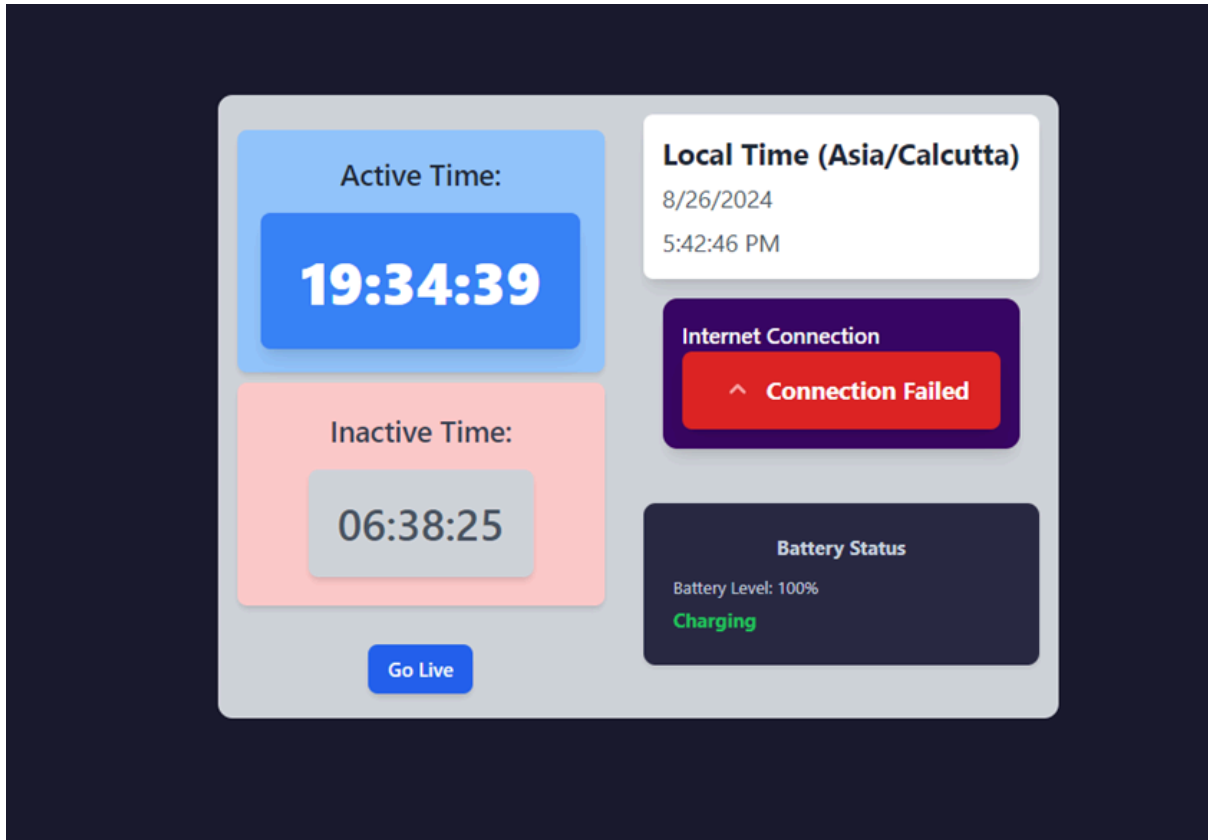


3. Additional Features

Internet Connection:

The user will be informed about their internet connection status. If the internet is properly connected, they will see 'Internet Connection (Connected)' displayed on their screen. If the connection is lost, 'Internet Connection (Not Connected)' will be shown. This ensures that users are always aware of their network status, allowing them to troubleshoot connectivity issues promptly. Additionally, the system could periodically check the connection and update the status in real-time.





Battery Status:

The user will be informed about their battery status. If the device is charging, they will see 'Battery Status (Charging)' displayed on their screen. If the device is not charging, 'Battery Status (Not Charging)' will be shown. This ensures that users are always aware of their battery condition, allowing them to manage power usage and charging needs effectively.

Active Time:

19:34:39

Inactive Time:

06:38:25

Go Live

Local Time (Asia/Calcutta)

8/26/2024
5:42:01 PM

Internet Connection

✓ Connected

Battery Status

Battery Level: 100%
Charging

Active Time:

19:34:39

Inactive Time:

06:38:25

Go Live

Local Time (Asia/Calcutta)

8/26/2024
5:45:00 PM

Internet Connection

✓ Connected

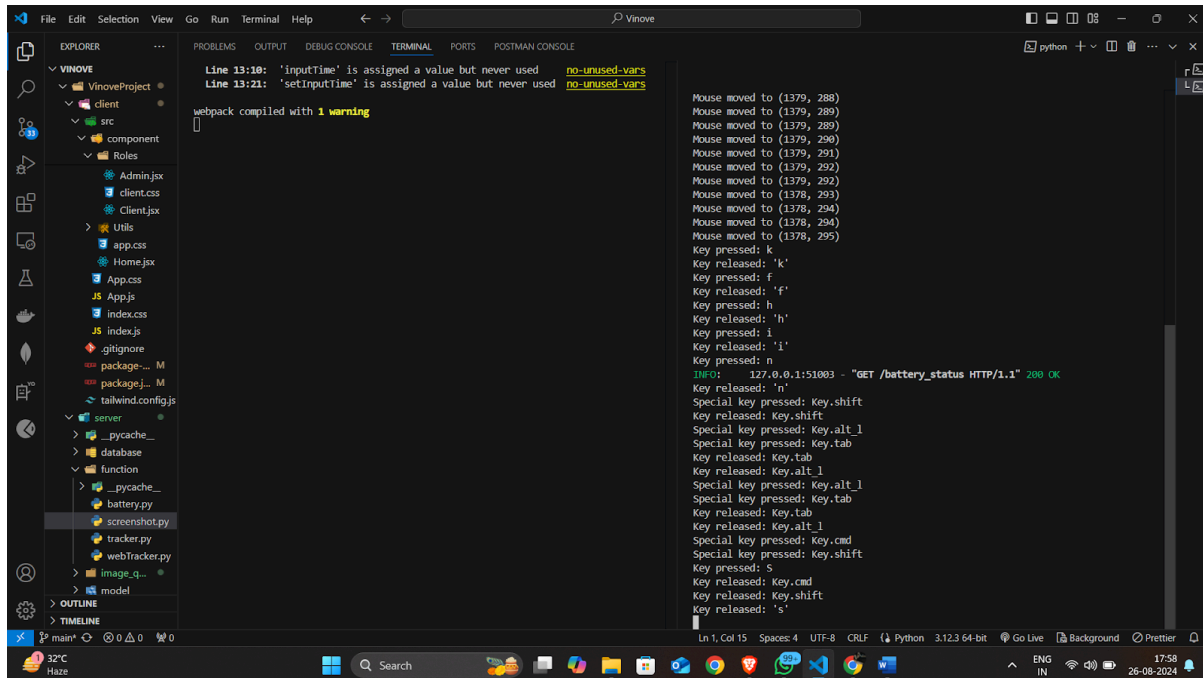
Battery Status

Battery Level: 100%
Not Charging

Mouse / Keyboard Detection:

The system will detect any movement of the mouse and any key pressed on the keyboard. If the user moves the mouse, a notification will display 'Mouse Moved' to indicate the movement.

Additionally, if a key is pressed on the keyboard, the specific key pressed will be shown on the screen, for example, 'Key Pressed: A'. This functionality allows for real-time monitoring of user input, ensuring that every mouse movement and keystroke is accurately recorded and displayed.



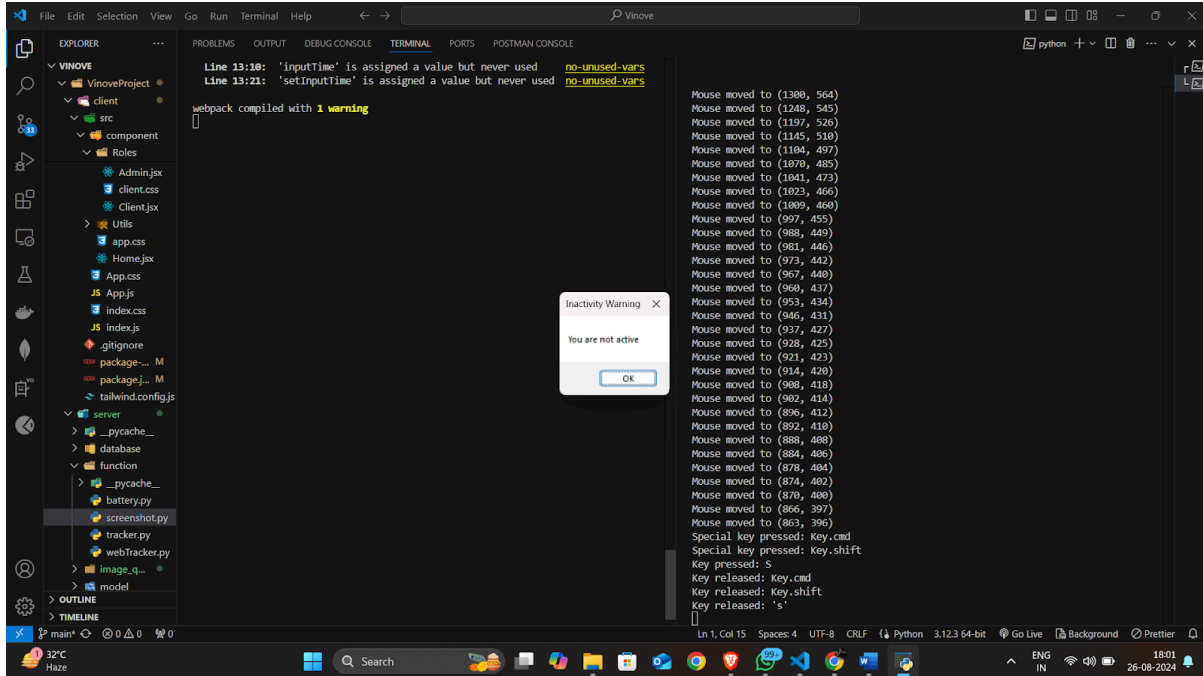
```
File Edit Selection View Go Run Terminal Help
VINOVE
VinoProject
  client
  src
  component
  Roles
    Admin.jsx
    client.css
    Client.jsx
  Utils
    app.css
    Home.jsx
    App.js
    index.css
    index.js
  .gitignore
  package.json
  package-lock.json
  tailwind.config.js
  server
    __pycache__
    database
    function
    __pycache__
    battery.py
    screenshot.py
    tracker.py
    webTracker.py
    image_q...
  model
  OUTLINE
  TIMELINE
  main*

Line 13:10: 'inputTime' is assigned a value but never used no-unused-vars
Line 13:21: 'setInputTime' is assigned a value but never used no-unused-vars
webpack compiled with 1 warning

Mouse moved to (1379, 288)
Mouse moved to (1379, 289)
Mouse moved to (1379, 289)
Mouse moved to (1379, 290)
Mouse moved to (1379, 291)
Mouse moved to (1379, 292)
Mouse moved to (1379, 292)
Mouse moved to (1378, 293)
Mouse moved to (1378, 294)
Mouse moved to (1378, 294)
Mouse moved to (1378, 295)
Key pressed: k
Key released: 'k'
Key pressed: f
Key released: 'f'
Key pressed: h
Key released: 'h'
Key pressed: i
Key released: 'i'
Key pressed: n
INFO: 127.0.0.1:51003 - "GET /battery_status HTTP/1.1" 200 OK
Key released: 'n'
Special key pressed: Key.shift
Key released: Key.shift
Special key pressed: Key.alt_1
Special key pressed: Key.tab
Key released: Key.tab
Key released: Key.alt_1
Special key pressed: Key.alt_1
Special key pressed: Key.tab
Key released: Key.tab
Key released: Key.alt_1
Special key pressed: Key.cmd
Special key pressed: Key.shift
Key pressed: s
Key released: Key.cmd
Key released: Key.shift
Key released: 's'
```

Activity Status:

If a user remains inactive for a couple of minutes—without moving the mouse or pressing any keys—a pop-up notification will appear on the screen, stating 'You are not active.' This feature ensures that any periods of inactivity are promptly highlighted, encouraging the user to remain engaged or take necessary actions to avoid being marked as idle.



4. Conclusion

The Python-based desktop agent application was successfully developed with all the core features implemented as per the requirements. The application is robust, secure, and user-friendly, ensuring that employee activity is monitored accurately while maintaining data privacy and integrity.