

User Access Management System

Final Project Report

Yash Patkar- 22110296

12 th November 2025

Abstract

The **User Access Management System (UAM)** provides administrators the ability to control user logins based on defined time schedules. Developed using **Python, Flask, and SQLite**, it allows defining, checking, and logging access permissions automatically. Each user's access is verified periodically, and outcomes are displayed in a web-based dashboard. The project demonstrates a functional prototype of time-based access control for shared computing environments.

1 Introduction

Unrestricted access to shared computing systems can lead to misuse and inefficiency. This project aims to build a simple, effective solution to regulate access based on user-specific time slots. The implemented tool automates permission checking and maintains detailed logs. The combination of Flask for the dashboard, SQLite for storage, and Python for automation creates a cohesive prototype suitable for institutional or lab setups.

2 Objectives and Achievements

- **Admin control over time-based access:** Implemented using a web interface for schedule management.
- **Simulated SSH/RDP enforcement:** Achieved through schedule-driven permission logic.
- **Database-based record management:** Achieved via SQLite tables for users, schedules, and logs.

- **Automated verification:** Periodic script execution through Task Scheduler or PowerShell loop.
- **Dashboard visualization:** Web dashboard displays all access history clearly.

3 System Architecture

The system has three main components:

1. **Frontend (Flask Dashboard):** Interface for adding users, schedules, and reviewing logs.
2. **Backend (SQLite Database):** Stores all user, schedule, and log data persistently.
3. **Scheduler (Python Script):** Periodically verifies time-based permissions and updates results.

The screenshot shows a web-based dashboard titled "User Access Management Dashboard". It includes sections for "Add New User" (with fields for Username and Full name), "Users" (a table with one entry: ID 1, Username Yash, Full name Yash Patkar), "Add Schedule" (with dropdowns for User (Yash), Day (0-Mon. 6-Sun) (0), Start (HH:MM) (09:00), End (HH:MM) (17:00), and a "Add Schedule" button), and "Schedules" (a table showing three entries for user Yash with start times 00:08, 00:11, and 09:00, and end times 00:10, 00:13, and 10:00 respectively). The interface uses a light blue color scheme and standard HTML/CSS styling.

Figure 1: Main dashboard showing user addition, schedule inputs, and data tables.

4 Implementation Details

Core Files

- `init_db.py`: Initializes the SQLite database.
- `app.py`: Provides the Flask-based admin dashboard.
- `access_checker.py`: Checks and logs whether each user is allowed or denied based on time.

Automation was implemented using Windows Task Scheduler and PowerShell loops to run the checker script at intervals.

Schedules			
User	Day	Start	End
Yash	3	00:08	00:10
Yash	3	00:11	00:13

Figure 2: Schedule entries in the dashboard for user Yash.

```

while ($true) {
    python .\access_checker.py
    Start-Sleep -Seconds 60
}

```

5 Results and Demonstration

Testing involved two users, **Yash** and **Madhu**. Multiple test runs were conducted to verify automatic logging and status accuracy.

```

PS C:\Users\Student> cd "$env:USERPROFILE\uam_demo"
PS C:\Users\Student\uam_demo> .\venv\Scripts\Activate.ps1
(venv) PS C:\Users\Student\uam_demo> python .\access_checker.py
    ✓ Check completed and logs updated.
(venv) PS C:\Users\Student\uam_demo> python -c "import datetime; print(datetime.datetime.now().weekday())"
3
(venv) PS C:\Users\Student\uam_demo> python .\access_checker.py
Yash: ALLOWED at 00:10
    ✓ Check completed and logs updated.
(venv) PS C:\Users\Student\uam_demo> python .\access_checker.py
Yash: ALLOWED at 00:13
    ✓ Check completed and logs updated.
(venv) PS C:\Users\Student\uam_demo> python .\access_checker.py
Yash: DENIED at 00:14
    ✓ Check completed and logs updated.
(venv) PS C:\Users\Student\uam_demo> python .\access_checker.py
Yash: DENIED at 00:27
Madhu: ALLOWED at 00:27

```

Figure 3: Terminal log showing Yash as ALLOWED at scheduled times.

2025-11-13T00:13:16	Yash	ALLOWED
2025-11-13T00:10:22	Yash	ALLOWED

Figure 4: Showing Yash's successful ALLOWED entries.

Yash	3	09:00	10:00
Timestamp	User	Action	
2025-11-13T00:14:13	Yash	DENIED	

Figure 5: Dashboard showing an entry where Yash was DENIED outside the allowed time.

Yash	3	09:00	10:00
Madhu	3	00:26	00:28

Figure 6: Schedules configured for Yash and Madhu on the same weekday.

The system correctly handled multiple users and logged events for each according to their respective schedules.

2025-11-13T00:40:19	Yash	DENIED
2025-11-13T00:27:36	Madhu	ALLOWED

Figure 7: Yash ALLOWED and Madhu DENIED, demonstrating multi-user verification.

Recent Logs		
Timestamp	User	Action
2025-11-13T01:08:24	Madhu	ALLOWED
2025-11-13T01:08:24	Yash	DENIED
2025-11-13T01:07:21	Madhu	DENIED
2025-11-13T01:07:21	Yash	DENIED
2025-11-13T01:00:33	Madhu	DENIED
2025-11-13T00:58:01	Yash	ALLOWED
2025-11-13T00:58:01	Madhu	DENIED
2025-11-13T00:40:19	Madhu	ALLOWED
2025-11-13T00:40:19	Yash	DENIED
2025-11-13T00:27:36	Madhu	ALLOWED
2025-11-13T00:27:36	Yash	DENIED
2025-11-13T00:14:13	Yash	DENIED
2025-11-13T00:13:16	Yash	ALLOWED
2025-11-13T00:10:22	Yash	ALLOWED

⚠ Note: For safety, this demo only simulates access enforcement — it does not actually lock Windows accounts.

Figure 8: Consolidated log table showing actions with timestamps and color-coded status.

User Access Management Dashboard					
+ Add New User					
Users					
ID	Username	Full name			
1	Yash	Yash Patel			
2	Madhu	Madhu Malhotra			
Add Schedule					
User	Day	Day (0-Mon-8-Sun)	Start (HH:MM)	End (HH:MM)	Save
Schedules					
User	Day	Day	Start	End	
Yash	0	0	00:00	00:00	
Yash	1	1	00:00	00:10	
Yash	2	2	00:00	00:20	
Madhu	3	3	00:00	00:30	
Madhu	4	4	00:00	00:40	
Madhu	5	5	00:00	00:50	
Yash	6	6	00:00	01:00	
Yash	7	7	00:00	01:00	
Madhu	8	8	00:00	01:00	
Madhu	9	9	00:00	01:00	
Yash	10	10	00:00	01:00	
Yash	11	11	00:00	01:00	
Madhu	12	12	00:00	01:00	
Yash	13	13	00:00	01:00	
Yash	14	14	00:00	01:00	
Madhu	15	15	00:00	01:00	
Yash	16	16	00:00	01:00	
Yash	17	17	00:00	01:00	
Madhu	18	18	00:00	01:00	
Yash	19	19	00:00	01:00	
Yash	20	20	00:00	01:00	
Madhu	21	21	00:00	01:00	
Yash	22	22	00:00	01:00	
Yash	23	23	00:00	01:00	
Madhu	24	24	00:00	01:00	
Yash	25	25	00:00	01:00	
Yash	26	26	00:00	01:00	
Madhu	27	27	00:00	01:00	
Yash	28	28	00:00	01:00	
Yash	29	29	00:00	01:00	
Madhu	30	30	00:00	01:00	
Yash	31	31	00:00	01:00	
Yash	32	32	00:00	01:00	
Madhu	33	33	00:00	01:00	
Yash	34	34	00:00	01:00	
Yash	35	35	00:00	01:00	
Madhu	36	36	00:00	01:00	
Yash	37	37	00:00	01:00	
Yash	38	38	00:00	01:00	
Madhu	39	39	00:00	01:00	
Yash	40	40	00:00	01:00	
Yash	41	41	00:00	01:00	
Madhu	42	42	00:00	01:00	
Yash	43	43	00:00	01:00	
Yash	44	44	00:00	01:00	
Madhu	45	45	00:00	01:00	
Yash	46	46	00:00	01:00	
Yash	47	47	00:00	01:00	
Madhu	48	48	00:00	01:00	
Yash	49	49	00:00	01:00	
Yash	50	50	00:00	01:00	
Madhu	51	51	00:00	01:00	
Yash	52	52	00:00	01:00	
Yash	53	53	00:00	01:00	
Madhu	54	54	00:00	01:00	
Yash	55	55	00:00	01:00	
Yash	56	56	00:00	01:00	
Madhu	57	57	00:00	01:00	
Yash	58	58	00:00	01:00	
Yash	59	59	00:00	01:00	
Madhu	60	60	00:00	01:00	
Yash	61	61	00:00	01:00	
Yash	62	62	00:00	01:00	
Madhu	63	63	00:00	01:00	
Yash	64	64	00:00	01:00	
Yash	65	65	00:00	01:00	
Madhu	66	66	00:00	01:00	
Yash	67	67	00:00	01:00	
Yash	68	68	00:00	01:00	
Madhu	69	69	00:00	01:00	
Yash	70	70	00:00	01:00	
Yash	71	71	00:00	01:00	
Madhu	72	72	00:00	01:00	
Yash	73	73	00:00	01:00	
Yash	74	74	00:00	01:00	
Madhu	75	75	00:00	01:00	
Yash	76	76	00:00	01:00	
Yash	77	77	00:00	01:00	
Madhu	78	78	00:00	01:00	
Yash	79	79	00:00	01:00	
Yash	80	80	00:00	01:00	
Madhu	81	81	00:00	01:00	
Yash	82	82	00:00	01:00	
Yash	83	83	00:00	01:00	
Madhu	84	84	00:00	01:00	
Yash	85	85	00:00	01:00	
Yash	86	86	00:00	01:00	
Madhu	87	87	00:00	01:00	
Yash	88	88	00:00	01:00	
Yash	89	89	00:00	01:00	
Madhu	90	90	00:00	01:00	
Yash	91	91	00:00	01:00	
Yash	92	92	00:00	01:00	
Madhu	93	93	00:00	01:00	
Yash	94	94	00:00	01:00	
Yash	95	95	00:00	01:00	
Madhu	96	96	00:00	01:00	
Yash	97	97	00:00	01:00	
Yash	98	98	00:00	01:00	
Madhu	99	99	00:00	01:00	
Yash	100	100	00:00	01:00	
Yash	101	101	00:00	01:00	
Madhu	102	102	00:00	01:00	
Yash	103	103	00:00	01:00	
Yash	104	104	00:00	01:00	
Madhu	105	105	00:00	01:00	
Yash	106	106	00:00	01:00	
Yash	107	107	00:00	01:00	
Madhu	108	108	00:00	01:00	
Yash	109	109	00:00	01:00	
Yash	110	110	00:00	01:00	
Madhu	111	111	00:00	01:00	
Yash	112	112	00:00	01:00	
Yash	113	113	00:00	01:00	
Madhu	114	114	00:00	01:00	
Yash	115	115	00:00	01:00	
Yash	116	116	00:00	01:00	
Madhu	117	117	00:00	01:00	
Yash	118	118	00:00	01:00	
Yash	119	119	00:00	01:00	
Madhu	120	120	00:00	01:00	
Yash	121	121	00:00	01:00	
Yash	122	122	00:00	01:00	
Madhu	123	123	00:00	01:00	
Yash	124	124	00:00	01:00	
Yash	125	125	00:00	01:00	
Madhu	126	126	00:00	01:00	
Yash	127	127	00:00	01:00	
Yash	128	128	00:00	01:00	
Madhu	129	129	00:00	01:00	
Yash	130	130	00:00	01:00	
Yash	131	131	00:00	01:00	
Madhu	132	132	00:00	01:00	
Yash	133	133	00:00	01:00	
Yash	134	134	00:00	01:00	
Madhu	135	135	00:00	01:00	
Yash	136	136	00:00	01:00	
Yash	137	137	00:00	01:00	
Madhu	138	138	00:00	01:00	
Yash	139	139	00:00	01:00	
Yash	140	140	00:00	01:00	
Madhu	141	141	00:00	01:00	
Yash	142	142	00:00	01:00	
Yash	143	143	00:00	01:00	
Madhu	144	144	00:00	01:00	
Yash	145	145	00:00	01:00	
Yash	146	146	00:00	01:00	
Madhu	147	147	00:00	01:00	
Yash	148	148	00:00	01:00	
Yash	149	149	00:00	01:00	
Madhu	150	150	00:00	01:00	
Yash	151	151	00:00	01:00	
Yash	152	152	00:00	01:00	
Madhu	153	153	00:00	01:00	
Yash	154	154	00:00	01:00	
Yash	155	155	00:00	01:00	
Madhu	156	156	00:00	01:00	
Yash	157	157	00:00	01:00	
Yash	158	158	00:00	01:00	
Madhu	159	159	00:00	01:00	
Yash	160	160	00:00	01:00	
Yash	161	161	00:00	01:00	
Madhu	162	162	00:00	01:00	
Yash	163	163	00:00	01:00	
Yash	164	164	00:00	01:00	
Madhu	165	165	00:00	01:00	
Yash	166	166	00:00	01:00	
Yash	167	167	00:00	01:00	
Madhu	168	168	00:00	01:00	
Yash	169	169	00:00	01:00	
Yash	170	170	00:00	01:00	
Madhu	171	171	00:00	01:00	
Yash	172	172	00:00	01:00	
Yash	173	173	00:00	01:00	
Madhu	174	174	00:00	01:00	
Yash	175	175	00:00	01:00	
Yash	176	176	00:00	01:00	
Madhu	177	177	00:00	01:00	
Yash	178	178	00:00	01:00	
Yash	179	179	00:00	01:00	
Madhu	180	180	00:00	01:00	
Yash	181	181	00:00	01:00	
Yash	182	182	00:00	01:00	
Madhu	183	183	00:00	01:00	
Yash	184	184			

- Centralized database for multiple computers.
- Alerts or notifications for denied attempts.

References

1. Microsoft Learn. (2024). *Task Scheduler Overview*. Retrieved from <https://learn.microsoft.com/en-us/windows/win32/taskschd/task-scheduler-start-page>
2. MongoDB Documentation. (2023). *Managing Time-Based Records*. Retrieved from <https://www.mongodb.com/docs/manual/tutorial/manage-timestamps/>
3. Linux Manual Pages. (2023). *cron(8) and systemd.timer(5)*. Retrieved from <https://man7.org/linux/man-pages/man8/cron.8.html>
4. OpenSSH Project. (2024). *OpenSSH Manual – Secure Shell (SSH)*. Retrieved from <https://www.openssh.com/manual.html>
5. SQLite Consortium. (2023). *SQLite Official Documentation*. Retrieved from <https://www.sqlite.org/docs.html>
6. Pallets Projects. (2024). *Flask Framework Documentation*. Retrieved from <https://flask.palletsprojects.com/>
7. Python Software Foundation. (2024). *Python 3.12 Documentation*. Retrieved from <https://docs.python.org/3/>
8. Saini, M., & Kumar, A. (2020). *Design and Implementation of Automated Access Control System Using Python and SQLite*. International Journal of Computer Applications, 176(35), 12–18.
9. Pal, S., & Banerjee, D. (2022). *Time-Based Authentication for Secure Multi-User Systems*. Journal of Information Security Research, 13(4), 267–275.