**Intern Details**

**Name:** Sharyu Suhas Dole  
**Intern ID:** IJ250006  
**Project Link:**

**Project Title**

EV User Profile Manager Using Python

**1. Introduction**

This project focuses on building a simple Python-based console application called EV User Profile Manager. The system allows for the efficient management of electric vehicle (EV) users, including their profiles and battery swap history. It simulates real-world EV battery swap tracking using JSON file storage, demonstrating core Python skills.

**2. Objective**

* To create a console-based user management system for EV users.
* To simulate and log EV battery swap events.
* To implement persistent storage using JSON files.
* To apply Python concepts such as file handling, dictionaries, conditionals, and loops.

**3. Tools and Technologies Used**

* Programming Language: Python 3.x
* Libraries/Modules: json, datetime
* IDE: VS Code / PyCharm
* Database: JSON File-based storage

**4. System Requirements**

* Hardware: Minimum 2 GB RAM, 1 GHz Processor
* **S**oftware: Python 3.x installed
* Dependencies: Standard Python libraries (json, datetime)

**5. Project Modules / Features**

* Add User: Allows entry of new EV user profiles with name and vehicle number.
* Edit User: Modify existing user information.
* Delete User: Remove a user record from the system.
* View Swap History: Display a user’s battery swap log.
* Simulate Swap: Record the date and time of a simulated battery swap event.
* Persistent Storage: All data is saved in a local JSON file.

**6. Project Flow / Working**

1. The user runs the Python script, which launches a menu-driven CLI (command-line interface).
2. The main menu offers six options: Add, Edit, Delete, View History, Simulate Swap, Exit.
3. Each function reads from or writes to users.json, ensuring persistent user and history data.
4. The Simulate Swap function appends the current date and time to a user’s history list.
5. All user data, including swap history, is stored in structured JSON format.

**7. Output Screenshots**

Output

--- EV User Profile Manager ---  
1. Add User  
2. Edit User  
3. Delete User  
4. View Swap History  
5. Simulate Swap  
6. Exit  
Choose an option (1-6): 1  
Enter new User ID: 30  
Enter Name: Mohan  
Enter Vehicle Number: 5348  
User added successfully.  
  
--- EV User Profile Manager ---  
1. Add User  
2. Edit User  
3. Delete User  
4. View Swap History  
5. Simulate Swap  
6. Exit

JSON File

{  
 "143": {  
 "name": "Sharyu Suhas Dole",  
 "vehicle\_no": "7007",  
 "history": [  
 "Swapped on 2025-06-27 15:24"  
 ]  
 },  
 "144": {  
 "name": "Shreya ",  
 "vehicle\_no": "6080",  
 "history": []  
 },  
 "145": {  
 "name": "Shruti",  
 "vehicle\_no": "7683",  
 "history": []  
 },  
 "146": {  
 "name": "Shriya Patil",  
 "vehicle\_no": "(876",  
 "history": []  
 },  
 "30": {  
 "name": "Mohan",  
 "vehicle\_no": "5348",  
 "history": []  
 }  
}

**8. Testing and Evaluation**

* Manual Testing**:** Each function was tested with valid and invalid inputs to check error handling.
* Observations:
  + Adding and editing users reflects accurately in the JSON file.
  + Swap events are correctly timestamped.
  + Data integrity is maintained even after multiple operations.

**9. Conclusion**

This project was an excellent exercise in implementing a file-based data management system using Python. It helped strengthen concepts like user input handling, dictionary manipulation, JSON file operations, and creating real-world command-line interfaces.

**10. Acknowledgment**

I would like to thank my mentor and institution for their support and encouragement throughout this project. Their feedback greatly enhanced the application’s features and usability.

**11. References**

* <https://docs.python.org/3/>
* Stack Overflow
* GeeksforGeeks
* W3Schools

Code

import json  
from datetime import datetime  
  
# File to store user data  
USER\_FILE = "users.json"  
  
# Load users from JSON file  
def load\_users():  
 try:  
 with open(USER\_FILE, 'r') as file:  
 return json.load(file)  
 except FileNotFoundError:  
 return {}  
  
# Save users to JSON file  
def save\_users(users):  
 with open(USER\_FILE, 'w') as file:  
 json.dump(users, file, indent=4)  
  
# Add a new user  
def add\_user():  
 users = load\_users()  
 user\_id = input("Enter new User ID: ")  
 if user\_id in users:  
 print("User ID already exists.")  
 return  
 name = input("Enter Name: ")  
 vehicle\_number = input("Enter Vehicle Number: ")  
 users[user\_id] = {  
 "name": name,  
 "vehicle\_no": vehicle\_number,  
 "history": []  
 }  
 save\_users(users)  
 print("User added successfully.")  
  
# Edit an existing user  
def edit\_user():  
 users = load\_users()  
 user\_id = input("Enter User ID to edit: ")  
 if user\_id not in users:  
 print("User not found.")  
 return  
 name = input("Enter new Name: ")  
 vehicle\_number = input("Enter new Vehicle Number: ")  
 users[user\_id]["name"] = name  
 users[user\_id]["vehicle\_no"] = vehicle\_number  
 save\_users(users)  
 print("User updated successfully.")  
  
# Delete a user  
def delete\_user():  
 users = load\_users()  
 user\_id = input("Enter User ID to delete: ")  
 if user\_id in users:  
 del users[user\_id]  
 save\_users(users)  
 print("User deleted successfully.")  
 else:  
 print("User not found.")  
  
# View a user's swap history  
def view\_history():  
 users = load\_users()  
 user\_id = input("Enter User ID to view history: ")  
 if user\_id in users:  
 history = users[user\_id]["history"]  
 if history:  
 print(f"\nSwap History for {user\_id}:")  
 for event in history:  
 print(f"- {event}")  
 else:  
 print("No swap history available.")  
 else:  
 print("User not found.")  
  
# Simulate a battery swap and record it  
def simulate\_swap():  
 users = load\_users()  
 user\_id = input("Enter User ID to simulate swap: ")  
 if user\_id in users:  
 timestamp = datetime.now().strftime("%Y-%m-%d %H:%M")  
 users[user\_id]["history"].append(f"Swapped on {timestamp}")  
 save\_users(users)  
 print("Swap recorded.")  
 else:  
 print("User not found.")  
  
# Main program loop  
def main():  
 while True:  
 print("\n--- EV User Profile Manager ---")  
 print("1. Add User")  
 print("2. Edit User")  
 print("3. Delete User")  
 print("4. View Swap History")  
 print("5. Simulate Swap")  
 print("6. Exit")  
  
 choice = input("Choose an option (1-6): ")  
  
 if choice == '1':  
 add\_user()  
 elif choice == '2':  
 edit\_user()  
 elif choice == '3':  
 delete\_user()  
 elif choice == '4':  
 view\_history()  
 elif choice == '5':  
 simulate\_swap()  
 elif choice == '6':  
 print("Exiting program.")  
 break  
 else:  
 print("Invalid choice. Please enter a number between 1 and 6.")  
  
# Run the program  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()