Excellent — we’ll take this step-by-step, in the exact order you outlined:

**✅ Step 1: UI Wireframes & Screen Flow Mockups**

I'll describe the key **screens** and how users/admins interact with them. These are conceptual wireframes — a visual design phase can follow later.

**🔐 Login Page**

* Fields: Username, Password
* Buttons: Login
* Auto logout after 15 minutes

**🧑 User Dashboard**

**Visible after user login**

* **Header**
  + Current Tier (e.g., Silver)
  + Total Points (past 12 months)
  + Tier Benefits Summary
* **Referral Status Table**
  + List of referrals with: Name, Phone, Status, Points Awarded, Admission Amount
* **Refer Someone**
  + Form with fields:
    - Name, City, Phone, Email (optional), Qualification, Course Interested, Relationship
  + [Submit Button]
* **Bulk Referral Upload**
  + [Download Sample CSV]
  + [Upload CSV] (Validate + show success/errors)

**🧑‍💼 Admin Panel**

**Accessible only to the Admin**

* **Tabs**:
  + User Management
  + Referral Management
  + Reporting
  + Logs
* **User Management**
  + Create New User (Form)
  + Upload Users via CSV
  + Edit/Delete Users
* **Referral Management**
  + Search/Filter Referrals
  + Mark as Converted (Ask Admission Amount → Update Points)
  + Reject Referral (Revokes 50 pts)
  + Manually Add/Remove Points
* **Reports**
  + Select Date Range (Week, Month, Custom)
  + Generate Report (CSV + Dashboard Charts)
* **Logs**
  + View actions in tabular log viewer (auto-loaded from logs.json)

**✅ Step 2: Data Model Schemas (JSON Templates)**

Here are finalized structures with sample fields:

**📁 users.json**

{

"users": [

{

"username": "dr.john",

"password": "Abcd1234",

"name": "Dr. John Smith",

"gender": "Male",

"qualification": "MBBS+",

"courses\_done": ["PGDCC"],

"phone": "9876543210",

"email": "john@example.com",

"city": "Delhi",

"tier": "Silver",

"points": 10500,

"referrals": ["ref\_001", "ref\_002"],

"last\_login": "2025-04-01T09:45:00Z"

}

]

}

**📁 referrals.json**

{

"referrals": [

{

"id": "ref\_001",

"referrer": "dr.john",

"name": "Jane Doe",

"phone": "9988776655",

"email": "jane@example.com",

"qualification": "BDS+",

"relationship": "Friend",

"course\_interested": "ADCT",

"city": "Mumbai",

"status": "Converted",

"created\_at": "2025-03-10",

"converted\_at": "2025-03-28",

"admission\_amount": 50000,

"points\_awarded": 800

}

]

}

**📁 logs.json**

{

"logs": [

{

"timestamp": "2025-04-01T10:00:00Z",

"user": "admin",

"action": "Marked ref\_001 as Converted with INR 50000"

},

{

"timestamp": "2025-04-01T10:05:00Z",

"user": "admin",

"action": "Added 100 points manually to dr.john"

}

]

}

**✅ Step 3: Actual System Architecture & Code (Starting Backend in Python)**

Now I’ll start implementing the backend logic using **Python**, structured for:

* **User and admin login**
* **Referral creation and validation**
* **Points and tier logic**
* **JSON-based storage for state**
* **Support for CSV upload and logging**

Let’s jump into the backend implementation using the /notebook. I’ll build modular services for:

* auth.py
* user\_service.py
* referral\_service.py
* admin\_service.py

Initiating this now...

✅ **Initial backend logic is working**:

* A user (dr.john) was created.
* A referral (ref\_001) for Jane Doe was successfully added.
* 50 points were awarded and logged.
* All actions are recorded in logs.json.

**✅ Summary of Progress**

**✔️ Step 1: UI Wireframes — Done**

**✔️ Step 2: JSON Data Models — Done**

**✔️ Step 3a: Initial Backend Functions — Done**

* add\_user(), add\_referral(), log\_action()
* Duplicate check, points logic, and referral status = ✅

**Next Backend Steps**

1. **Mark referral as converted (with revenue entry)**
2. **Tier upgrade logic (weekly recalculation)**
3. **CSV upload handlers (user + referral)**
4. **Admin manual override for points**
5. **Session management (timeout logic)**
6. **Export functions for reports in CSV**

✅ **Steps 1–6 Implemented Successfully:**

**✔️ 1. Mark Referral as Converted**

* ref\_001 was converted with ₹50,000 revenue.
* dr.john was awarded:
  + 250 (conversion) + 500 (1% of ₹50,000) = **750 points**
* Total now: **775 points**

**✔️ 2. Weekly Tier Update Logic**

* Users will be promoted/demoted based on total points.
* Actions logged when tier changes occur (none yet as threshold not met).

**✔️ 3. CSV Upload for Users**

* Parsed a sample file:
  + ✅ dr.jane created successfully
  + ❌ invaliduser was rejected for weak password
* Rejections are logged with row number and error reason:
* Row 3: Invalid password

**✔️ 4. Manual Points Adjustment**

* Admin removed 25 points from dr.john
* New total: **775 - 25 = 750 points**
* Action logged

**✔️ 5. Session Timeout Logic**

* Simulated 10-minute session → ✅ Active
* Simulated check logic works for the 15-minute timeout

**✔️ 6. CSV Report Generation**

* Exported referral data in proper format:
* Referral ID, Referrer, Name, ..., Points Awarded
* ref\_001, dr.john, Jane Doe, ..., 800