**NationGuard: Detailed Flow**

**🔓 Login System**

* User roles: BSF Border Office Admin, BSF Officer, Home Ministry Official, State Admin, Police Station Officer.
* Each role sees its specific dashboard after login.

**🚪 BSF Border Office Admin Role**

1. Border Office Admin creates accounts for BSF Officers.
2. Registers each BSF Officer's:
   * Full Name
   * Unique Officer ID (system-generated)
   * Face image (for facial authentication)
   * Border Point assigned

**🚪 Visitor Entry by BSF Officer**

1. BSF Officer logs into NationGuard with **User ID and Password**. / **User ID and Face Authentication.**
2. Officer selects **"New Visitor Entry"**.
3. Officer **scans QR code** on visitor's visa (using JS QR) → auto-fills:
   * Visitor Name
   * Passport Number
   * Country
   * Visa Type
   * Visa Expiry Date
4. Officer **uploads face image** of the visitor.
5. App **captures current geolocation** (border point) automatically or lets officer select from predefined border points.
6. Officer **re-verifies own face** before submitting visitor entry (for security).
7. Officer **submits** all details.
8. Visitor data saved in **"Active Visitors"** database with:
   * Officer ID (who entered the visitor)
   * Border Location
   * Timestamp

**Flow of Visitor Entry**

1. **Scans Visa QR for capture Details of Immigrant**
2. **Scans Immigrant Face**
3. **Post Immigrant Details with Face image in server**
4. **Server Makes Vector of Face representation and saves the details with vector in server cache and generate a session key attached that key in a array of object with**

**Immigrant details .**

1. **Sends back the session key in client side .**
2. **BSF Officer Scans His/Her face with His/Her UserID .**

**POST His/Her Face in Server , Server Find that BSF officer by using UserID**

1. **Server Compares the BSF officer face with Pre Registered Face .**
2. **If ok then server use the session key and retrive the Immigrant Details store in Cache and also retriever BSF officer Details like (Btn no, Name , Border Office ID , Timestamp)**

**📊 BSF Dashboard**

* View today's entries.
* Search visitors by Passport No., Country, Officer ID.
* Monitor visa expiry dates.

**👩‍👩‍👩 Home Ministry Dashboard**

* See list of all visitors inside India.
* **Highlight visitors** whose **visa has expired** but **no exit recorded**.
* **Forward visitor cases** to respective states where they were last located.

**🌏 State Dashboard**

* Receive forwarded cases from Home Ministry.
* View list of overstayed visitors in their state.
* Assign visitors to police stations under that state.

**🛍️ Police Station Dashboard**

* View detailed visitor profiles assigned to their jurisdiction.
* Options available:
  + **Locate** (Begin search for visitor)
  + **Investigate** (Collect local information)
  + **Report** (Update findings back to State/Home Ministry)

**🌐 Technology Stack**

* Frontend: React.js (with JS-QR for scanning)
* Backend: Node.js + Express.js
* Database: MongoDB (with collections for Visitors, Users, Officers, Locations)
* Location capture: Geolocation API
* Face recognition: Real-time verification using camera API.

**📊 Data Stored for Each Visitor**

* Name
* Passport Number
* Country
* Visa Type
* Visa Expiry Date
* Face Image URL
* Entry Location (Border name, lat-long)
* Entry Timestamp
* Officer ID (who entered)
* Status (Active / Overstayed / Cleared)

**📊 Data Stored for Each BSF Officer**

* Officer Name
* Officer Unique ID
* Face Image (for login and verification)
* Assigned Border Point
* Contact Details (Optional)

**✨ Future Scope**

* Integration with Airport Immigration.
* Real-time alerts to BSF and Police.
* Automatic Blacklist generation for serious overstayers.
* Integration with Aadhar/Passport databases.

**End of Flow** 🔄

Would you also like me to update a quick login/visitor-entry visual flow too?