**Workflow for Student Feedback System**

**1. System Initialization**

* **Faculty logs in:** Faculty logs into the system using their credentials.
* **Session setup:** Faculty sets up the session details (course, year, semester) and generates a unique **faculty\_id**.
* **Generate Attendance List:** Faculty ensures the list of students physically present in the class is accurate.
* **Virtual Attendance Matching:** The system will then generate a virtual attendance list that mirrors the confirmed physical attendance, ensuring that only students who are physically present are allowed to participate in the voting process.

**2. Student Participation Setup**

* **Faculty shares faculty\_id:** Faculty shares the **faculty\_id** with the students in the session.
* **Student app setup:** Students open the feedback app on their mobile devices.

**3. Voting Process**

* **Student checks in:**
  + **App startup:** Students open the app and enter the **faculty\_id**.
  + **Check Local Storage:** The app checks local storage to ensure no duplicate voting (i.e., the student hasn't already voted for this session).
  + **Verify Identity:** The app may prompt students for verification (e.g., student ID) to match against the physical attendance list.
  + **Register Vote Permission:** The app registers the student’s device for the session.

**4. Voting Commencement**

* **Faculty starts voting:**
  + **Initiate voting:** Faculty presses the "Start Voting" button on their interface.
  + **Voting period:** Students can now submit their feedback through the app.
  + **Anonymity ensured:** Feedback submitted by students is anonymous and stored in the PostgreSQL database.

**5. Vote Submission**

* **Feedback Submission:** Students submit their feedback.
  + **Data storage:** The feedback is stored on the server with unique identifiers ensuring anonymity.
  + **Prevent duplicates:** Once feedback is submitted, the app updates local storage to prevent duplicate votes.

**6. Voting Termination**

* **End voting:** Faculty presses the "Stop Voting" button once satisfied with the number of votes.
  + **Disable submission:** Voting for that session is closed, and no further feedback is accepted.

**7. Post-Voting**

* **Data Review:** Faculty can view the count of submitted feedbacks and compare it with the physical attendance.
* **Data Analysis:** The feedback data can be analyzed and reviewed as needed.

**System Components and Interactions**

1. **Frontend (Mobile App for Students):**
   * Interface for entering **faculty\_id**.
   * Feedback submission form.
   * Local storage to track voting status.
2. **Frontend (Web Interface for Faculty):**
   * Interface for session setup and generating **faculty\_id**.
   * Controls for starting and stopping the voting process.
   * Dashboard for viewing voting status and feedback count.
3. **Backend (Django Server):**
   * API endpoints for student verification, feedback submission, and vote count retrieval.
   * Database interactions with PostgreSQL to store session details, feedback data, and voting status.
4. **Database (PostgreSQL):**
   * Tables for session information, student records, feedback submissions, and voting status.

**Database Schema (Simplified)**

* **sessions:**
  + id (Primary Key)
  + faculty\_id
  + course
  + year
  + semester
  + start\_time
  + end\_time
  + status (active/inactive)
* **students:**
  + id (Primary Key)
  + student\_id
  + name
  + course
  + year
  + semester
  + attendance\_status (present/absent)
* **feedbacks:**
  + id (Primary Key)
  + session\_id (Foreign Key)
  + feedback\_text
  + submitted\_at
* **votes:**
  + id (Primary Key)
  + session\_id (Foreign Key)
  + student\_id (Foreign Key)
  + voted\_at

**Summary**

This workflow ensures an anonymous, secure, and organized process for collecting student feedback, aligned with the given considerations. It leverages Django for the backend and PostgreSQL for data management, with a clear separation of responsibilities between faculty and students, while preventing duplicate votes and ensuring only present students can participate.