Your proposal for maintaining a virtual student attendance register anonymously and ensuring that only approved students can cast their votes is well thought out. Here’s a detailed plan for implementing this system:

**System Components**

1. **Frontend (Student App):**
   * Generate and store a unique student\_id and faculty\_id in local storage when the student clicks the "proceed" button.
   * Submit feedback and update the timestamp in the virtual register.
2. **Backend (Django Server):**
   * API endpoints for registering attendance, submitting feedback, and approving students.
   * Middleware to check if a student is approved before allowing feedback submission.
3. **Database (PostgreSQL):**
   * Table virtual\_register to store student attendance records.

**Summary**

1. **Student App:** Generates a random student\_id and stores it locally along with faculty\_id.
2. **Register Student:** Adds the student to the virtual register when they click "Proceed".
3. **Submit Feedback:** Updates the timestamp when feedback is submitted, only if the student is approved.
4. **Start Voting Session:** Faculty approves all registered students and starts the voting session.
5. **Clear Virtual Register:** Faculty can clear the virtual register to start a new session.

This implementation ensures anonymity, prevents duplicate votes, and restricts participation to approved students only.