Sprint 3: Attendance Tracking and Reporting

Objective

Implement an automated attendance system with real-time tracking, reporting, and notifications.

Tasks

Attendance Tracking:

- Implement attendance marking through facial recognition.
- Integrate attendance data storage with the database.
- Ensure real-time updates for attendance records.

Automated Reports Generation:

- Develop a module to generate attendance reports.
- Enable Excel export functionality for reports.

Notification System:

- Send automated alerts to students and faculty for low attendance.
- Implement email notifications for attendance updates.

Boost Attendance with Rewards:

- To encourage students to maintain high attendance, introduce a reward-based system with badges and incentives.

Admin & Faculty Access:

- Provide role-based access to view, edit, and approve attendance records.
- Implement filtering and search features for attendance logs.

Testing & Optimization:

- Conduct unit and integration testing for attendance functionalities.

Success Criteria:

- Accurate attendance tracking and reporting.
- Automated notifications for students and faculty.

Technologies used:

- Implement face detection using OpenCV for real-time attendance marking.
- Store attendance records in MySQL using JDBC with efficient indexing.
- Reporting: Generate attendance reports in Excel using Apache POI.
- Send alerts via JavaMail API for low attendance.

Retrospective

Goal:

Develop a robust attendance tracking system with facial recognition, real-time reporting, and automated notifications.

What Went Well:

- Successfully integrated facial recognition for attendance marking.
- Automated report generation and real-time data updates were effective.
- Notifications for low attendance worked as expected.

Challenges Faced:

- Facial recognition struggled in low-light conditions and with occlusions (masks, glasses).
- High processing time when handling a large number of students simultaneously.

Improvements:

- Enhance the facial recognition model with better pre-processing techniques and dataset augmentation.
- Optimize algorithm efficiency to reduce processing time.