Event Management System – Documentation Report

---

1. Document Your Approach

Assumptions and Decisions

- Each student can register for multiple events but only once per event.
- Events are associated with a specific college.
- Attendance can only be marked for registered students.
- Feedback is optional and rated on a scale of 1–5.

AI Tool Usage

- Used ChatGPT to brainstorm database schema, API endpoints, and workflows.
- Adapted suggestions to match project requirements, such as using SQLite for simplicity.

AI Conversation Log

- Chat logs from ChatGPT brainstorming were saved as screenshots and links.
- Some suggestions were followed (ER diagram, sequence diagrams), others adapted (frontend interactivity, table designs).

---

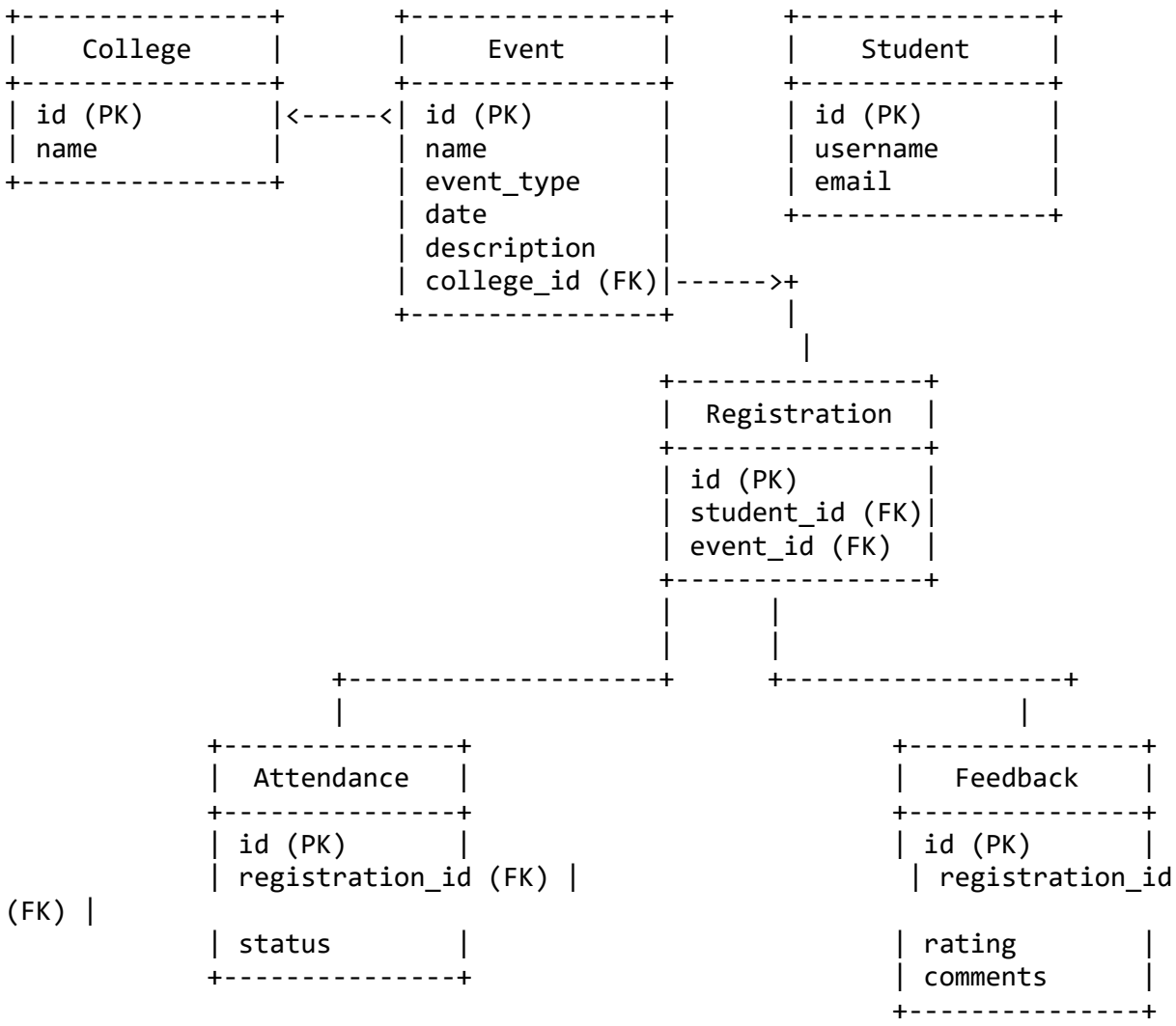
2. Design Document

Data to Track

- Event creation (name, type, date, description, college).
- Student registration (student, event).
- Attendance (present/absent).

- Feedback (rating 1–5, comments).

Database Schema (ER Diagram)

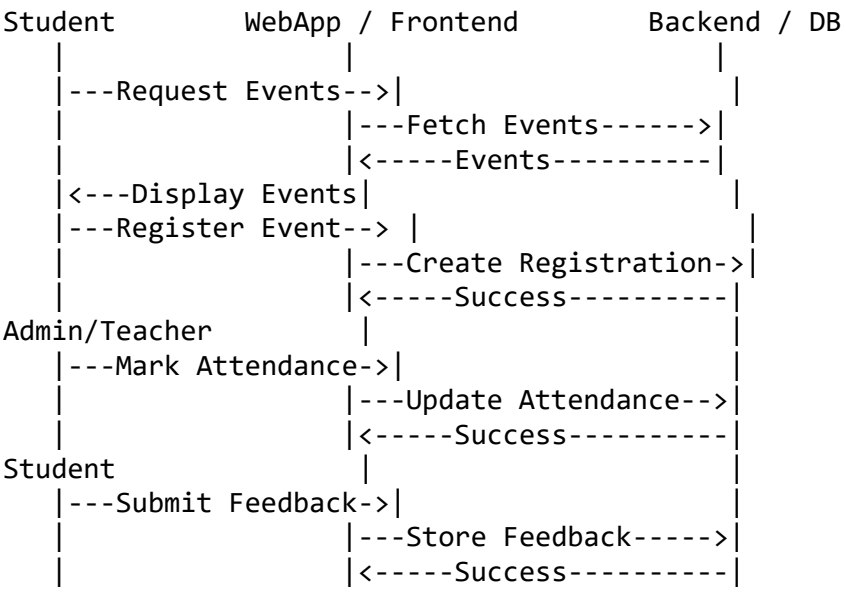


API Design

- `POST /student/register_event` → Register student to an event.
- `POST /admin/mark_attendance` → Mark attendance for students.
- `POST /student/feedback` → Submit feedback.
- `GET /admin/dashboard_data` → Fetch all data for admin dashboard.
- `GET /report/registrations` → Get total registrations per event.
- `GET /report/attendance` → Get attendance percentage.

- `GET /report/feedback` → Get average feedback score.

Workflows (Sequence Diagram)



Assumptions & Edge Cases

- Duplicate registrations are prevented.
- Cancelled events remove pending registrations.
- Feedback is optional and defaults to null if not provided.

---

3. Prototype Implementation

Backend: Flask (Python)

- Database: SQLite
- Models: College, Event, Student, Registration, Attendance, Feedback
- Example endpoint: `/admin/dashboard_data` returns JSON for frontend.

Frontend: HTML + CSS + JS

- Dynamic tables for colleges, events, students, registrations, top 3 active students.

- Interactive buttons and hover effects.
- Fetch API used to load data from backend.

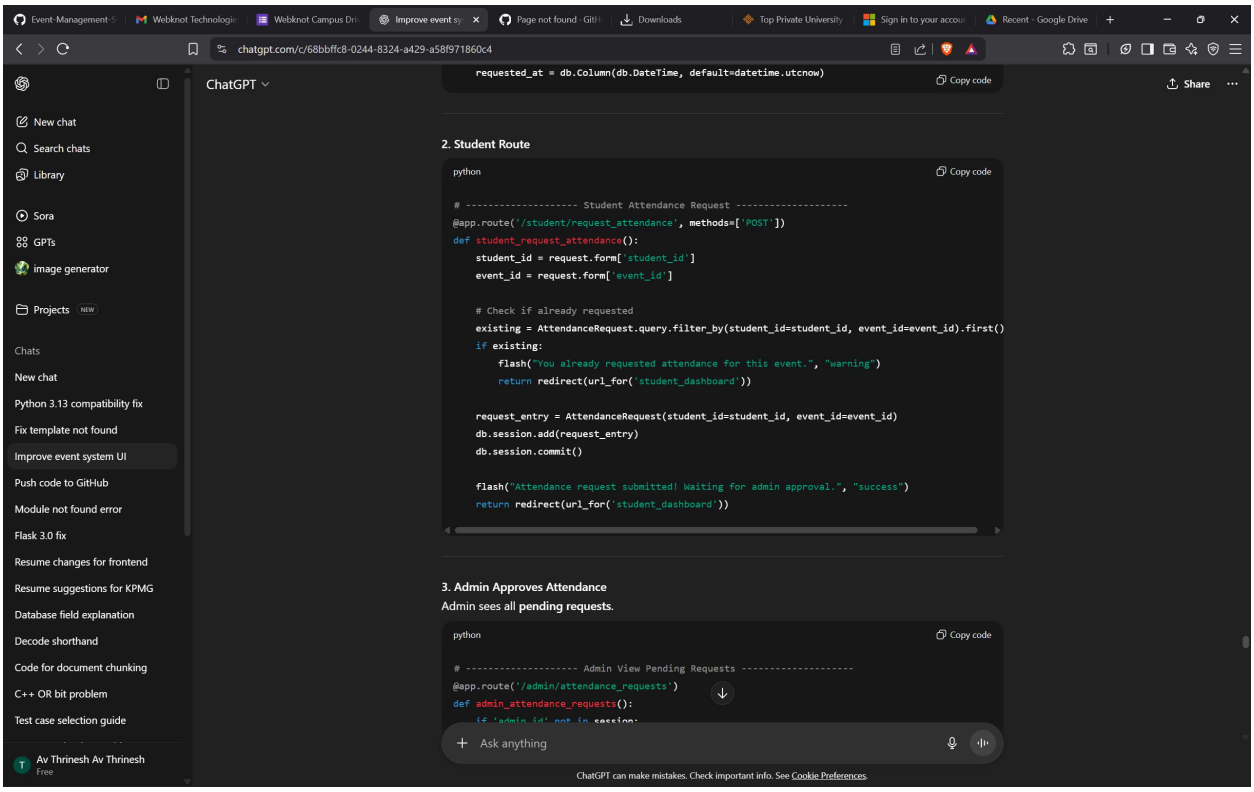
```
<!-- Example: Admin Dashboard Table -->
<table id="registrations-event-table">
    <caption>Registrations per Event</caption>
    <thead>
        <tr>
            <th>Event ID</th>
            <th>Event Name</th>
            <th>Registrations Count</th>
        </tr>
    </thead>
    <tbody></tbody>
</table>

// Fetch dashboard data
async function loadDashboard() {
    const response = await fetch("/admin/dashboard_data");
    const data = await response.json();


    // Populate tables dynamically
    const tableBody = document.querySelector("#registrations-event-table tbod
y");
    data.registrations_per_event.forEach(ev => {
        const row = document.createElement("tr");
        row.innerHTML = `<td>${ev.event_id}</td><td>${ev.event_name}</td><td>
${ev.count}</td>`;
        tableBody.appendChild(row);
    });
}
loadDashboard();
```

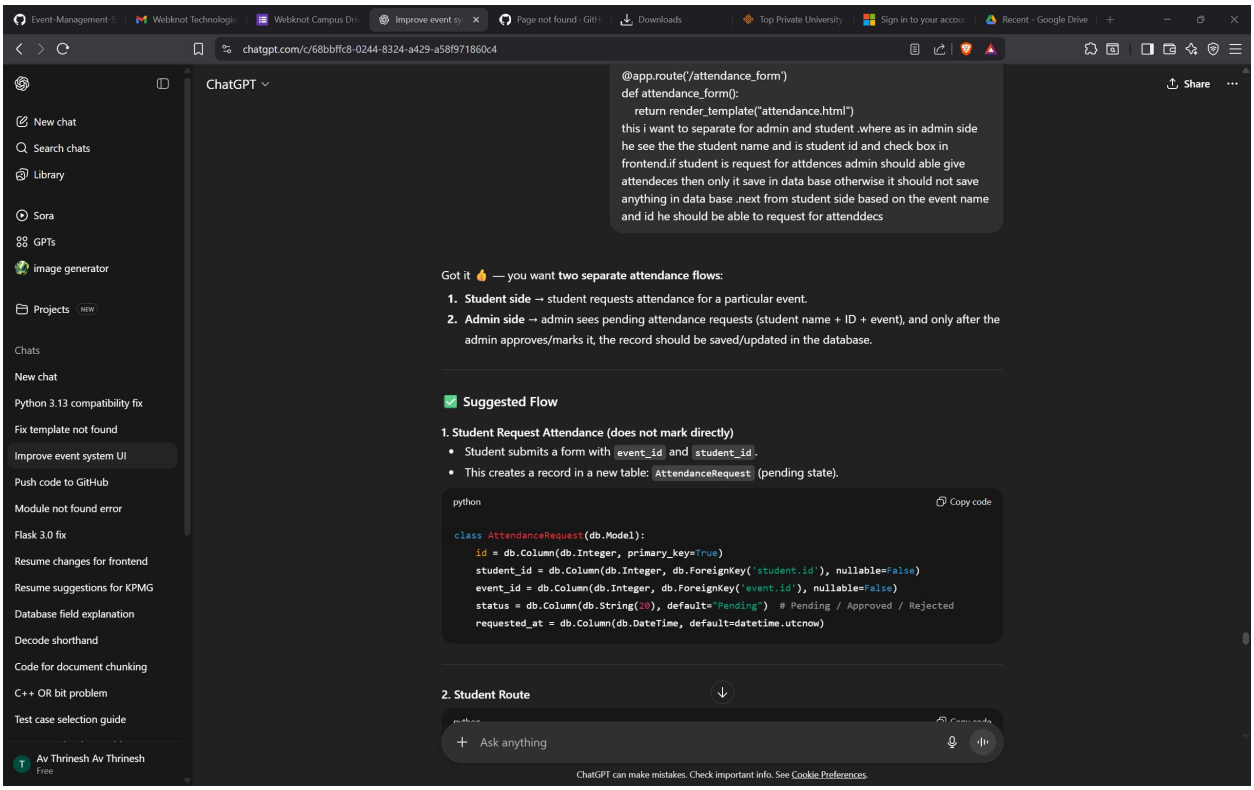
Sample Queries / Reports

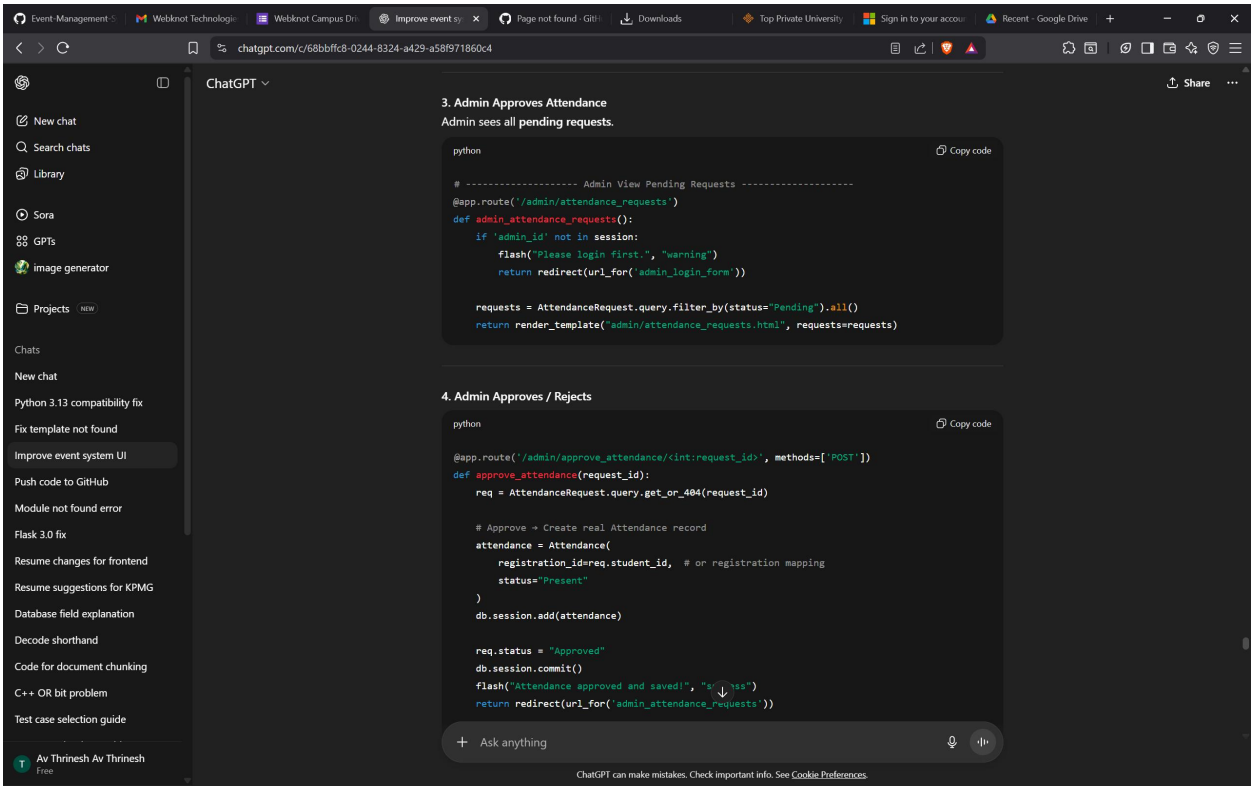
- Total registrations per event: `SELECT event_id, COUNT(*) FROM registration GROUP BY event_id;`
- Attendance percentage: `SELECT student_id, (SUM(CASE WHEN status='Present' THEN 1 ELSE 0 END)/COUNT(*))*100 FROM attendance GROUP BY student_id;`
- Average feedback score: `SELECT event_id, AVG(rating) FROM feedback GROUP BY event_id;`

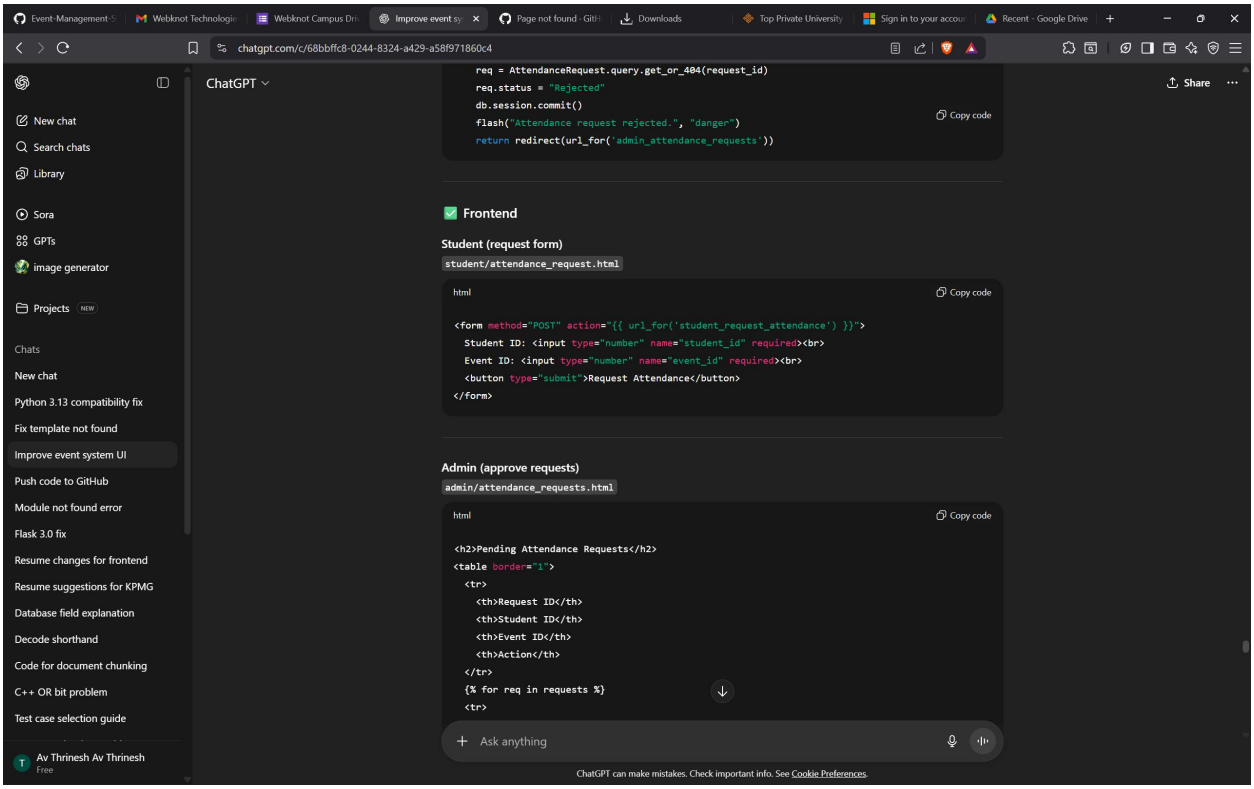
Some screenshots of chat with Ai:

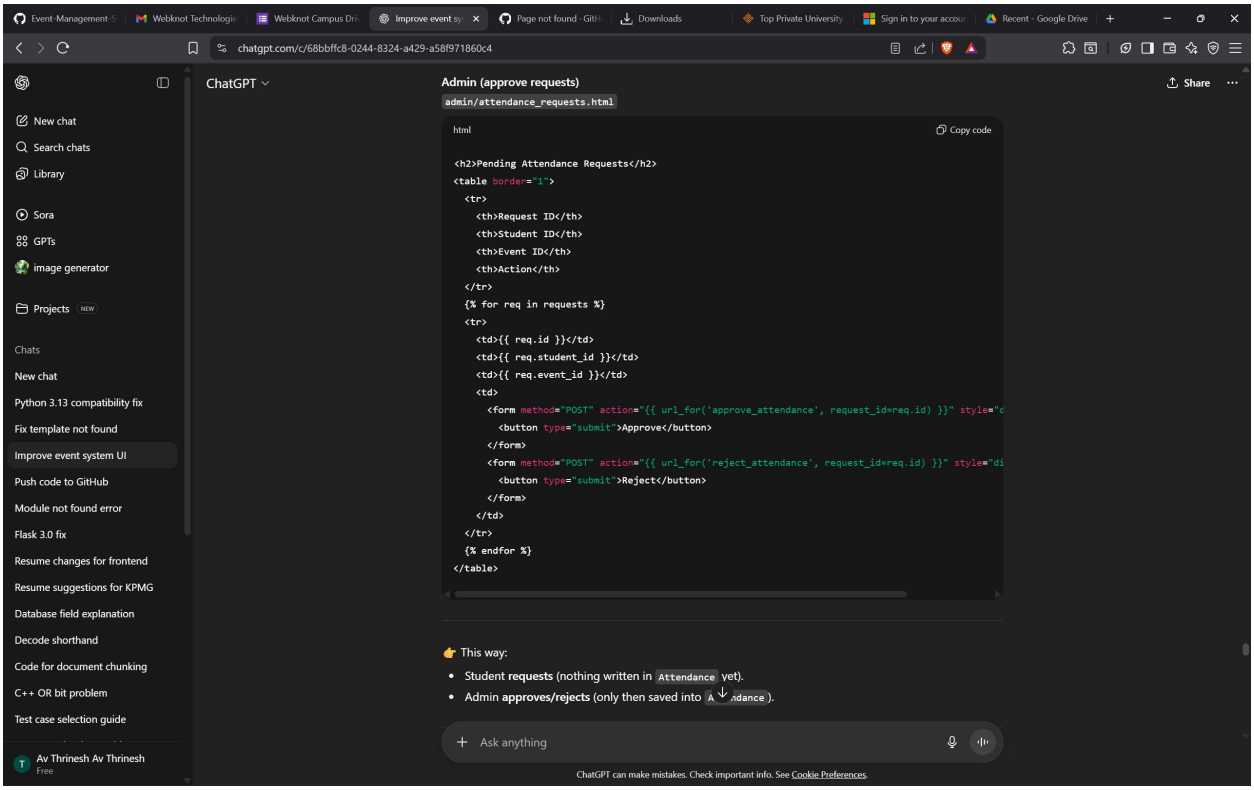
Link: https://chatgpt.com/share/68bcdf71-e64c-8013-80c8-7e4e7ac2dc55






chatgpt.com/c/68bbffc8-0244-8324-a429-a58f971860c4
ChatGPT
Share
New chat
Search chats
Library
Sora
GPTs
image generator
Projects
Chats
New chat
Python 3.13 compatibility fix
Fix template not found
Improve event system UI
Push code to GitHub
Module not found error
Flask 3.0 fix
Resume changes for frontend
Resume suggestions for KPMG
Database field explanation
Decode shorthand
Code for document chunking
C++ OR bit problem
Test case selection guide
Av Thrinesh Av Thrinesh
Free
3. Admin Approves Attendance
Admin sees all pending requests.
python
Copy code
------------------- Admin View Pending Requests -------------------
@app.route('/admin/attendance_requests')
def admin_attendance_requests():
if 'admin_id' not in session:
flash("Please login first.", "warning")
return redirect(url_for('admin_login_form'))
requests = AttendanceRequest.query.filter_by(status="Pending").all()
return render_template("admin/attendance_requests.html", requests=requests)
4. Admin Approves / Rejects
python
Copy code
@app.route('/admin/approve_attendance/<int:request_id>', methods=['POST'])
def approve_attendance(request_id):
req = AttendanceRequest.query.get_or_404(request_id)
# Approve → Create real Attendance record
attendance = Attendance(
registration_id=req.student_id, # or registration mapping
status="Present"
)
db.session.add(attendance)
req.status = "Approved"
db.session.commit()
flash("Attendance approved and saved!", "success")
return redirect(url_for('admin_attendance_requests'))
Ask anything
ChatGPT can make mistakes. Check important info. See Cookie Preferences.


chatgpt.com/c/68bbffc8-0244-8324-a429-a58f971860c4
ChatGPT
Share
req = AttendanceRequest.query.get_or_404(request_id)
req.status = "Rejected"
db.session.commit()
flash("Attendance request rejected.", "danger")
return redirect(url_for('admin_attendance_requests'))
Copy code
Frontend
Student (request form)
student/attendance_request.html
html
Copy code
<form method="POST" action="{{ url_for('student_request_attendance') }}">
Student ID: <input type="number" name="student_id" required>
Event ID: <input type="number" name="event_id" required>
<button type="submit">Request Attendance</button>
</form>
Admin (approve requests)
admin/attendance_requests.html
html
Copy code
<h2>Pending Attendance Requests</h2>
<table border="1">
<tr>
<th>Request ID</th>
<th>Student ID</th>
<th>Event ID</th>
<th>Action</th>
</tr>
{% for req in requests %}
<tr>
Ask anything
ChatGPT can make mistakes. Check important info. See Cookie Preferences.


Event-Management-S
Webknot Technologie
Webknot Campus Dri
Improve event sy
Page not found · GitH
Downloads
Top Private University
Sign in to your accour
Recent - Google Drive
chatgpt.com/c/68bbffc8-0244-8324-a429-a58f971860c4
ChatGPT
New chat
Search chats
Library
Sora
GPTs
image generator
Projects NEW
Chats
New chat
Python 3.13 compatibility fix
Fix template not found
Improve event system UI
Push code to GitHub
Module not found error
Flask 3.0 fix
Resume changes for frontend
Resume suggestions for KPMG
Database field explanation
Decode shorthand
Code for document chunking
C++ OR bit problem
Test case selection guide
Av Thrinesh Av Thrinesh
Free
Admin (approve requests)
admin/attendance_requests.html
Share
html
Copy code
<h2>Pending Attendance Requests</h2>
<table border="1">
<tr>
<th>Request ID</th>
<th>Student ID</th>
<th>Event ID</th>
<th>Action</th>
</tr>
{% for req in requests %}
<tr>
<td>{{ req.id }}</td>
<td>{{ req.student_id }}</td>
<td>{{ req.event_id }}</td>
<td>
<form method="POST" action="{{ url_for('approve_attendance', request_id=req.id) }}" style="d
<button type="submit">Approve</button>
</form>
<form method="POST" action="{{ url_for('reject_attendance', request_id=req.id) }}" style="di
<button type="submit">Reject</button>
</form>
</td>
</tr>
{% endfor %}
</table>
👉 This way:
Student requests (nothing written in Attendance yet).
Admin approves/rejects (only then saved into A ↓ ndance).
Ask anything
ChatGPT can make mistakes. Check important info. See Cookie Preferences.