CS-207
FixTitute

Abhinav Dixit (B16003) *, Aman Jain (B16044)†, Lokesh Kumar (B16061) ‡ and Anand Ramrakhyani (B16124)§
* b16003@students.iitmandi.ac.in
† b16044@students.iitmandi.ac.in
‡ b16061@students.iitmandi.ac.in
§ b16124@students.iitmandi.ac.in

I. Project Statement

With the schedule of the community members getting hectic day-by-day, clashes in the arrangement of events in institute like ours are getting more and more frequent. In many cases, such clashes result in the waste of time of the members. Hence, there is a need for some event management portal, which can be used to manage the events of the institute.

FixTitute fills this need. It allows the organiser of any event (which can be some extra class, arranged by some faculty or, can be some club related activity) to avoid scheduling the event at that time when many (or all) of its participants would not be able to attend the event, by creating certain groups of participants. *FixTitute* works via managing different events by grouping the members into several groups.

II. ER Diagram

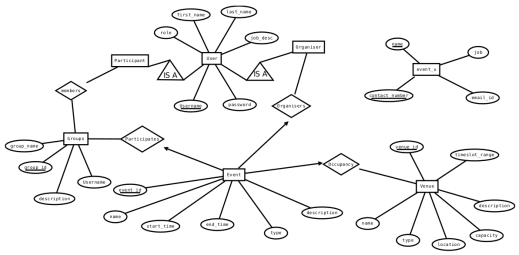

Fig. 1. The Proposed ER Diagram for the application.

III. Relational Schema

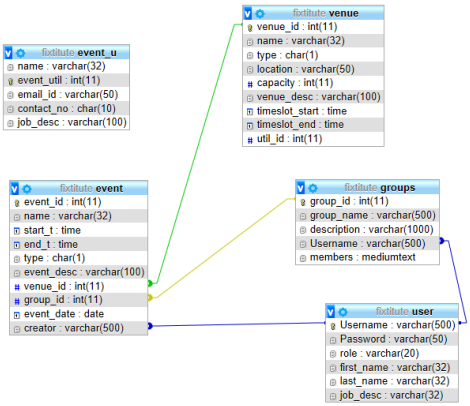

Fig. 2. The Relational Schema for the application.

A. *event table*

Stores the details of all events going to take place in the institute.

B. *user table*

Stores the details of all the user accounts along with the privileges.

C. *venue*

Predefined table, with details of possible venues of the events.

D. *groups*

Created by organizer, having collection of people (club members, etc.) for frequent creation of similar events.

E. event_u

Stores info about people relevant to maintenance and housekeeping of a venue (ex. lab asssistants,etc.)

IV. Intended User Groups

- Event organizers (Faculty, Dean, Club Coordinators)
- Attendees of the event (Students, Staff, Outsiders)

V. Features

- Front-desk booking of events
- Front-desk cancellation of events
- Searching for names of attendees of the event
- Notifying the participants through one click
- Searching for upcoming events
- Searching for the venue of events
- Sorting events by different group

VI. Front end design - Technology Used

HTML5 and CSS3 in conjugation with JavaScript and jQuery 3.1.1.

VII. Few Complex SQL Queries

Some of the SQL queries which we have implemented in our project are :

- ```
  CREATE TRIGGER group_maintain
  AFTER DELETE ON groups
  FOR EACH ROW
  DELETE FROM event
  WHERE event.group_id old.group_id ;
  ```

- ```
  CREATE TRIGGER venue_maintain
  AFTER DELETE ON venue
  FOR EACH ROW
  DELETE FROM event
  WHERE event.venue_id old.venue_id ;
  ```

- ```
  SELECT name FROM venue
  WHERE
  venue_id =  (SELECT venue_id FROM event
  WHERE creator = '$user_id';)
  ```

- ```
  SELECT * FROM event
  WHERE event_date = '$evedate'
  AND
  start_t = '$starttime' AND group_id = '$gid'
  AND end_t = '$endtime';
  ```

- ```
  SELECT * FROM user
  WHERE Username = '$username'
  ```

- ```
  DELETE FROM event
  WHERE event_id = '$eventId'
  ```

- ```
  DELETE FROM groups
  WHERE group_id = '$groupId'
  ```

- ```
  SELECT * FROM venue;
  ```

- ```
  INSERT INTO event
  (name, start_t, end_t, venue_id, event_date,
  creator, group_id, type, event_desc)
  VALUES ('$evename','$starttime','$endtime',
  '$venueid','$evedate','$creator','$gid',
  'N','$descrip');
  ```

- ```
  UPDATE groups
  SET group_name = '$groupName', description = '$description',
  members='$memberId')
  WHERE group_id = '$gid';
  ```

VIII. Performance Checks

For databases as large as ours, the application runs fine.

The insertion time for inserting 1 query is 0.07 seconds. While, the time taken to delete the complete table having 1000 entries is 0.395 seconds.

IX. Work Planned but could not do

- Integration of user database with LDAP : After making the project more efficient and accurate, we could have linked this portal with the IIT Mandi Intranet (similarly to OAS, MOODLE, etc.)

- Authentication of accounts and privileges with institute email-ID : In order to restrict non-authenticated users to create and/or delete events, we could have validated the different accounts.

- Feedback : Provision for addition of feedback for events attended could have been added.

- Attendance : Provision for maintenance of attendance for the events attended could have been added.

- Integration with IIT Mandi cloud : It could have been done for archiving the photos, videos, presentation materials, etc. of major events (like Exodia) for future references.

X. Our Experience of doing this project

It was quite enriching and rewarding while exploring the various technologies involved in front-end and back-end development and how they can be integrated with databases to create scalable and responsive websites,

which can be used to solve day-to-day problems.

We had great experience in learning about the MySQL database management system, as we implemented different queries of SQL including triggers.