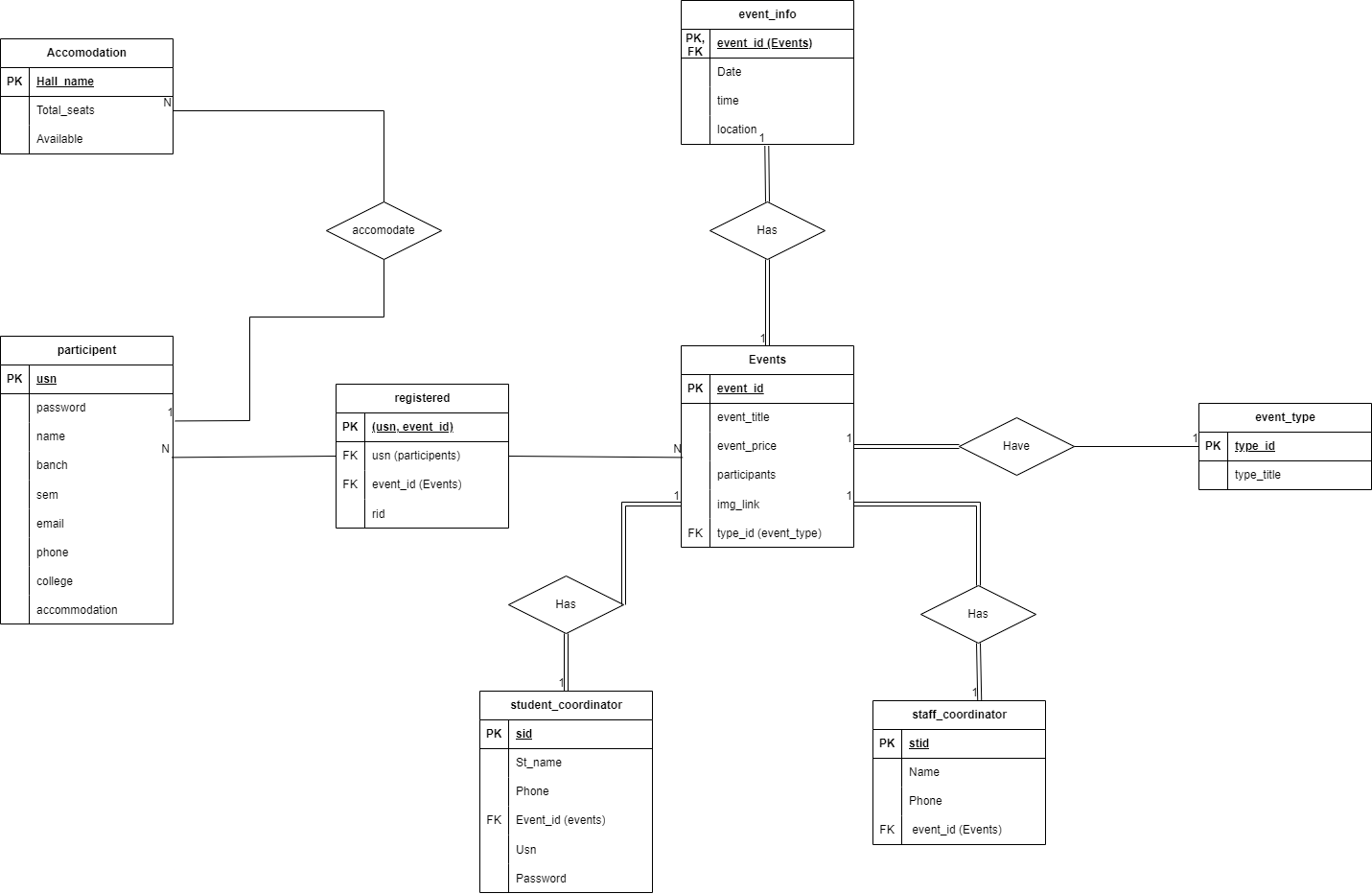
****

**Schema**

**Constraints:**

1. Foreign key constraint on EVENT: If an event type gets deleted or updated the corresponding changes happen in a cascading manner

ALTER TABLE `events`

  ADD CONSTRAINT `events\_ibfk\_1` FOREIGN KEY (`type\_id`) REFERENCES `event\_type` (`type\_id`) ON DELETE CASCADE ON UPDATE CASCADE;

1. Foreign key constraint on EVENT\_INFO: If an event is deleted or updated the changes occur in event\_info as well

-- Constraints for table `event\_info`

--

ALTER TABLE `event\_info`

  ADD CONSTRAINT `event\_info\_ibfk\_1` FOREIGN KEY (`event\_id`) REFERENCES `events` (`event\_id`) ON DELETE CASCADE ON UPDATE CASCADE;

1. Foreign key constraint on PARTICIPANTS: If a participant gets deleted or updated, corresponding changes will happen in registered table

-- Constraints for table `registered`

--

ALTER TABLE `registered`

  ADD CONSTRAINT `registered\_ibfk\_1` FOREIGN KEY (`usn`) REFERENCES `participent` (`usn`) ON DELETE CASCADE ON UPDATE CASCADE;

1. Foreign key constraint on STAFF\_COORDINATOR: If an event gets deleted, the corresponding changes will reflect on staff coordinator

-- Constraints for table `staff\_coordinator`

--

ALTER TABLE `staff\_coordinator`

  ADD CONSTRAINT `staff\_coordinator\_ibfk\_1` FOREIGN KEY (`event\_id`) REFERENCES `events` (`event\_id`) ON DELETE CASCADE ON UPDATE CASCADE;

1. Foreign key constraint on STUDENT\_COORDINATOR: If an event gets deleted, the corresponding changes will reflect on Student coordinator

-- Constraints for table `Student\_coordinator`

--

ALTER TABLE `student\_coordinator`

  ADD CONSTRAINT `student\_coordinator\_ibfk\_1` FOREIGN KEY (`event\_id`) REFERENCES `events` (`event\_id`) ON DELETE CASCADE ON UPDATE CASCADE;

**Triggers**

1. Whenever a new participant registers the participant count increases by 1

CREATE TRIGGER `count` AFTER INSERT ON `registered` FOR EACH ROW update events

set events.participents=events.participents+1

WHERE events.event\_id=new.event\_id

**Queries**

1. Displays the event details of all events of a particular type

"SELECT \* FROM events,event\_info ef,student\_coordinator s,staff\_coordinator st WHERE type\_id = $id and ef.event\_id=events.event\_id and s.event\_id=events.event\_id and st.event\_id=events.event\_id  "

1. Insert new participants into database

"INSERT INTO participent (usn,password,name,branch,sem,email,phone,college) VALUES('$usn','$pwd','$name','$branch',$sem,'$email','$phone','$college')"

1. Check if the username and password are valid

"SELECT \* FROM participent WHERE usn='$myusername' AND password='$mypassword'"

1. Displays all the event details to the admin

"SELECT \* FROM staff\_coordinator s ,event\_info ef ,student\_coordinator st,events e where e.event\_id= ef.event\_id and e.event\_id= s.event\_id and e.event\_id= st.event\_id"

1. Allows admin to create a new event and modify details

$INSERT="INSERT INTO events(event\_id,event\_title,event\_price,img\_link,type\_id) VALUES($event\_id,'$event\_title', $event\_price,'$img\_link',$type\_id);";

            $INSERT.= "INSERT INTO event\_info (event\_id,Date,time,location) Values ($event\_id,'$Date','$time','$location');";

            $INSERT.="INSERT into student\_coordinator(sid,st\_name,phone,event\_id)  values($event\_id,'$st\_name',NULL,$event\_id);";

$INSERT.="INSERT into staff\_coordinator(stid,name,phone,event\_id)  values($event\_id,'$name',NULL,$event\_id)";

1. Allows admin to delete an event

"delete from events where event\_id='$id';";

1. Allows the participants to view the event details in which he or she has participated

"SELECT \* FROM registered r,staff\_coordinator s ,event\_info ef ,student\_coordinator st,events e where e.event\_id= ef.event\_id and e.event\_id= s.event\_id and e.event\_id= st.event\_id and r.usn= '$usn' and r.event\_id=e.event\_id"

1. Display staff coordinator events and their corresponding details to admin

"SELECT \* FROM staff\_coordinator s ,events e where e.event\_id= s.event\_id"

1. Display student coordinator events and their corresponding details to admin

"SELECT \* FROM student\_coordinator s ,events e where e.event\_id= s.event\_id"

1. Allows admin to view participant details

"SELECT \* FROM events,registered r ,participent p WHERE events.event\_id=r.event\_id and r.usn = p.usn order by event\_title"

1. Allows admin to update staff details

"UPDATE staff\_coordinator set phone='$phone',name='$name' where stid='$id'"

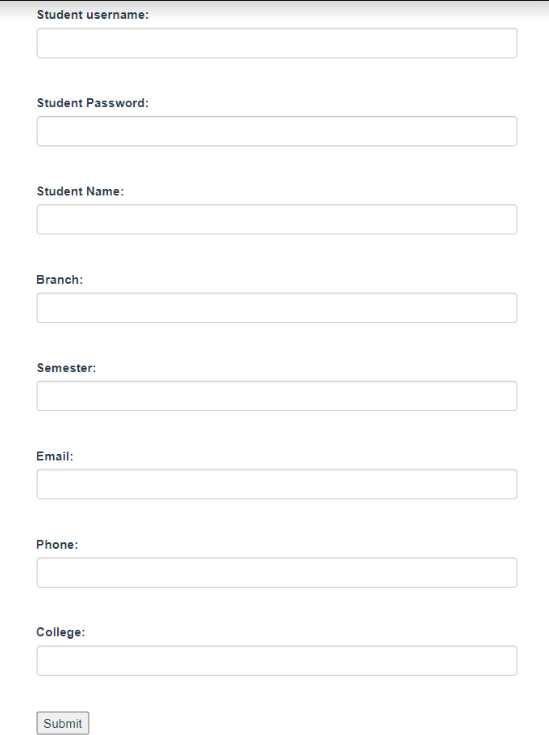
1. Allows admin to update participant details including username and password

"UPDATE student\_coordinator set phone='$phone',st\_name='$name' where sid='$id'"

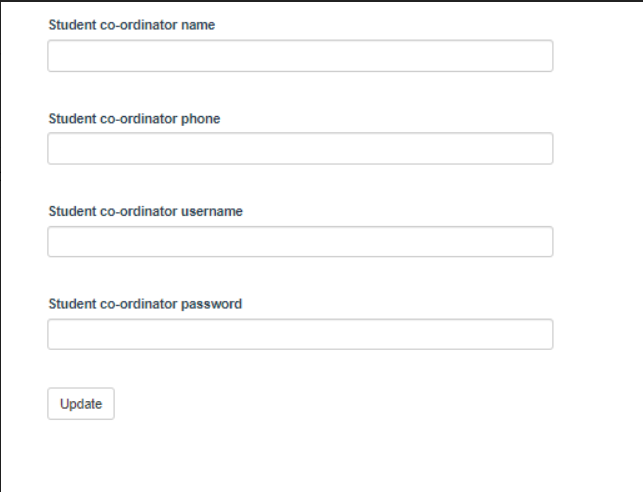
**Forms**

1. Registration Form:

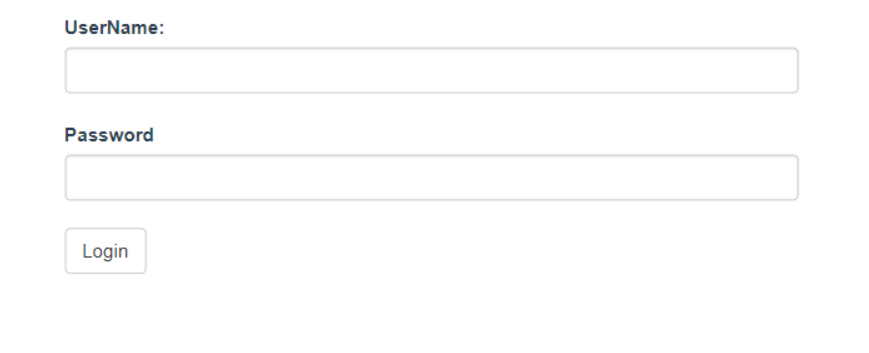
Used to input for registration of new participants



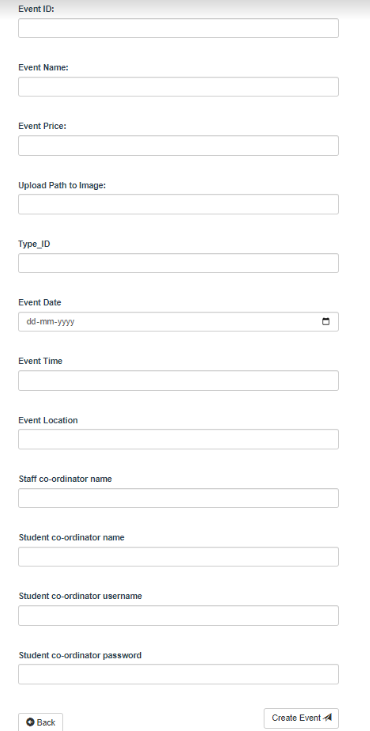
1. Update details of student coordinators, only accessible by admin



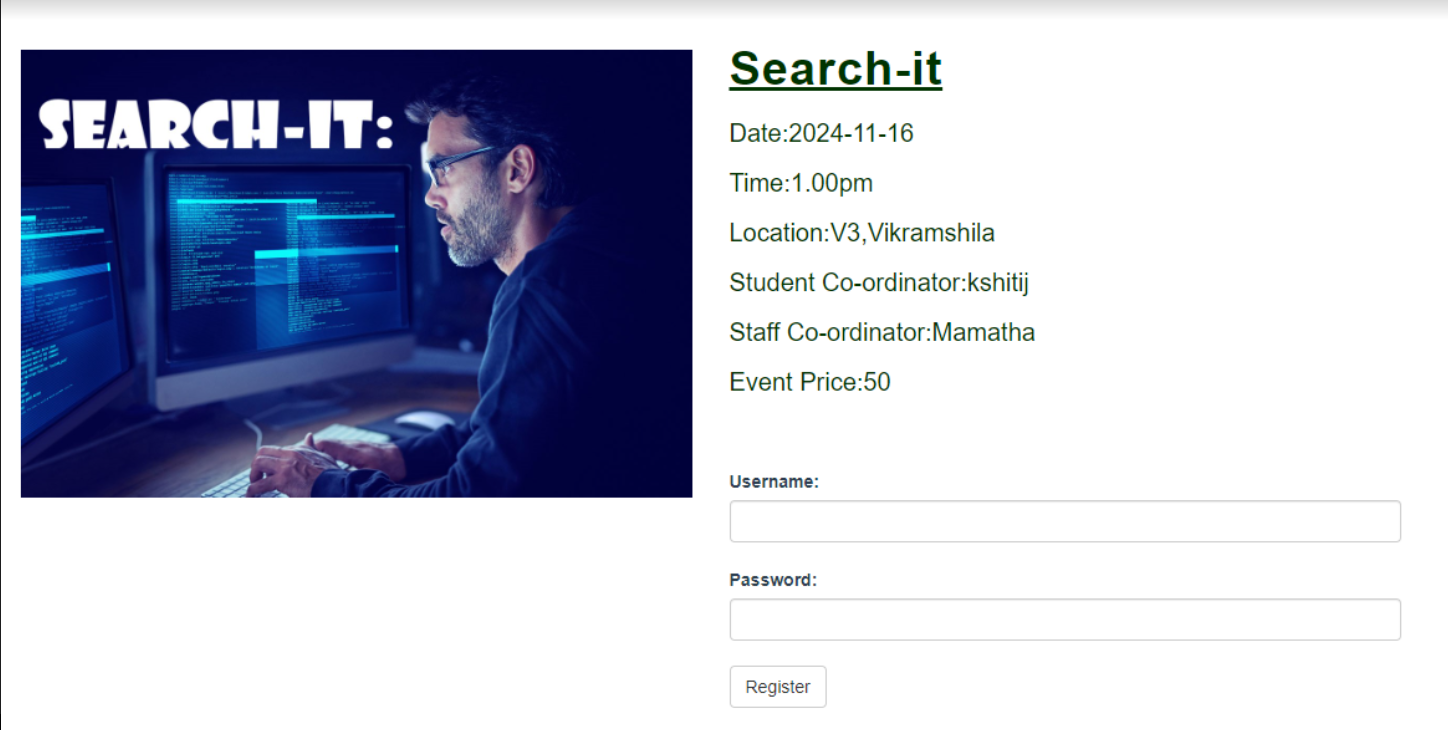
1. Login page for participants, admin and student coordinators



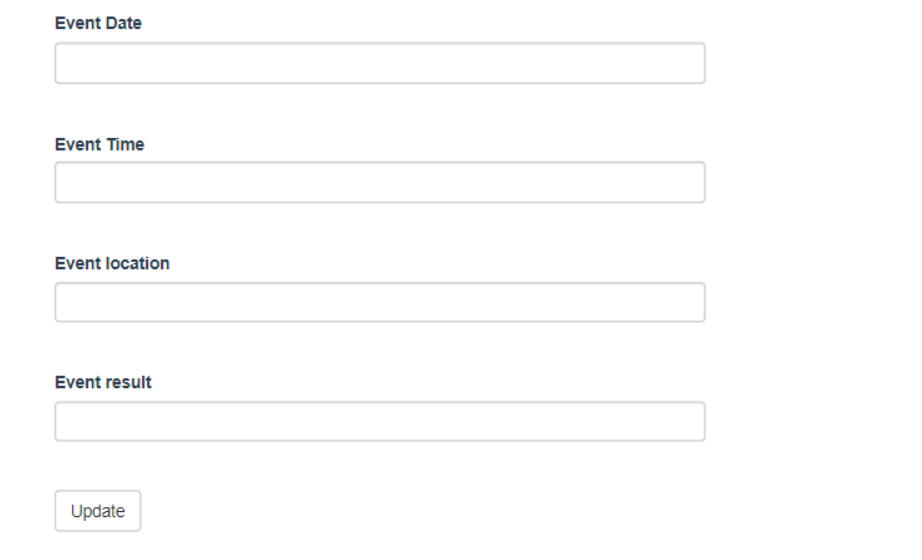
1. Allows admin to Create New events and assign event management to a Student Coordinator and a Staff Coordinator



1. Event Registration for participants



1. Update event details and result used by student coordinator



1. Update details of staff coordinator by admin

