Explanation of different roles:

SCHOLAR: Submit his/her reports (M/Q) and also search previous reports submitted by him/her.

SUPERVISOR: Gets all the reports (M/Q), he/she could approve/review the reports and also search the reports for all the scholars under his/her supervision. Besides, he/she could also delete the reports.

HEAD: Create, update, retrieve and delete supervisor within the department.

DEAN: Has all the functionality of head and could also create department(s).

ADMIN: Has all the rights of Dean plus he/she could create faculty(ies).

SUPERADMIN: Has all the possible permissions for administrative purpose. Only us as a service provider has this role.

=====================================================================================

Noteworthy Points:

* The whole page would be loaded only once for all users, subsequent pages/sections would be updated with the help of ajax request/response.
* The role of any supervisor populates the menu for the supervisor in the active session once authenticated with credentials i.e. for each page the left pane menu has to be populated on the basis of the allotted roles to the user.
* There has to be a master jsp for which the background image and color could be set according to the user preferences which are persistent in the database (optional).
* When any supervisor is created/updated, the form will be having a role field which would be a select dropdown having all the roles smaller than the current user role. There would be default role associated as SCHOLAR/SUPERVISOR to scholar/supervisor resp. created in the system.
* Each supervisor must have a “supervisor/manager” under whom it falls e.g. all heads must have dean as manager. Based on this approach when a head/dean/admin/ manages supervisors, he/she could update/delete/activate/deactivate only the supervisors which comes under him/her.
* When any supervisor (be it ADMIN, DEAN etc.,) logs in, it would be in supervisor view. If the supervisor has role like ADMIN, DEAN etc., he/she could login as that particular role too and this would change the menu with all the additional operations.
* Since we want to keep it as an open web application, any user could be registered in two modes. We could also engage supervisors (lecturers, readers, professors etc.) using same technique.
  + Independent – not associated with any institute
  + Associated – with any institution.

The advantage of this approach is that – an independent user can also register as associated user under a particular supervisor but since the supervisor has the scholar management capability, only he/she could activate that scholar and even delete that scholar. So, any attempt for unauthorized user is futile.

* For increased popularity, we could add forums and journals as live feed to the main page for guest visitors and certainly for associated users.

In code we could define below permissions but may be working internally:

VIEW\_REPORT

SUBMIT\_REPORT

REVIEW\_ REPORT

APPROVE\_REPORT

DELETE\_REPORT

CREATE\_SCHOLAR

VIEW\_SCHOLAR

UPDATE\_SCHOLAR

DELETE\_SCHOLAR

ACTIVATE\_SCHOLAR

DEACTIVATE\_SCHOLAR

CREATE\_SUPERVISOR

VIEW\_SUPERVISOR

UPDATE\_SUPERVISOR

DELETE\_SUPERVISOR

ACTIVATE\_SUPERVISOR

DEACTIVATE\_SUPERVISOR

CREATE\_DEPARTMENT

VIEW\_DEPARTMENT

DELETE\_DEPARTMENT

CREATE\_FACULTY

VIEW\_FACULTY

DELETE\_FACULTY

CREATE\_INSTITUTION

VIEW\_INSTITUTION

DELETE\_INSTITUTION