SOFTWARE REQUIREMENT SPECIFICATION FOR FACULTY WORKLOG PORTAL

STUDENT NAME: AMBUJA S

PROJECT ID: 19

PROJECT TITLE: FACULTY WORKLOG

TECHNICAL COMPONENTS:

COMPONENT	**TECH STACK**
BACKEND	NODE JS, EXPRESS JS
FRONT END	NODE JS
DATA BASE	MONGODB
API	OPEN API

PROBLEM STATEMENT:

Institutions need an efficient system to manage and monitor the daily activities of their faculty members across multiple clusters, including Academics, Special Labs, Skill Training, Training and Placement, and ELCC. Currently, there is no streamlined way for faculties to submit their daily worklogs, for cluster heads to review and approve or reject them, and for an Admin to oversee the entire process. This results in disorganized record-keeping, inefficient tracking of faculty activities, and difficulties in generating comprehensive reports. The Faculty Worklog Portal aims to address these issues by providing a centralized platform for worklog submission, approval, and reporting.

PROJECT FLOW:

Purpose

The purpose of creating the Faculty Worklog Portal is to streamline and enhance the management of daily faculty activities across various clusters. It facilitates efficient submission, review, and approval of worklogs by faculty members, cluster heads, and the Admin. This system aims to improve organization, accountability, and transparency, ensuring that all faculty activities are accurately tracked and reported. Additionally, it enables the generation of comprehensive reports, aiding in performance evaluation and administrative decision-making.

Scope

The scope of the Faculty Worklog Portal includes developing a centralized platform where faculties in six clusters can submit daily worklogs, cluster heads can approve or reject them, and an Admin can oversee, review, and generate reports on all activities, ensuring streamlined management and accountability.

Business context

The Faculty Worklog Portal is designed to enhance the operational efficiency of educational institutions. By providing a centralized platform for faculty members to submit daily worklogs and for cluster heads and an Admin to review and manage these submissions, the portal ensures organized record-keeping and efficient monitoring of faculty activities. This system supports better oversight, accountability, and streamlined reporting processes, ultimately contributing to improved administrative management and academic performance.

Consideration

- **Efficiency in Tracking:** Streamlining the process of daily worklog submission and approval to ensure all faculty activities are recorded accurately and reviewed promptly, reducing administrative overhead.
- **Transparency and Accountability:** Providing a clear approval workflow for heads and the Admin to enhance transparency in faculty performance evaluations and accountability for work completed.
- **Comprehensive Reporting:** Enabling the Admin to generate detailed reports on faculty activities, including approved and rejected worklogs, to support data-driven decision-making and performance assessments.

Dependencies

- **Database System:** A robust and scalable database to store and manage faculty worklogs, user profiles, approval statuses, and generated reports efficiently and securely.

- **User Authentication and Authorization:** Implement secure login mechanisms and role-based access control to ensure that only authorized users (faculties, heads, admin) can access specific functionalities.
- **Web Development Framework:** Utilize a reliable web development framework, such as MongoDB, expressJS or NodeJS to build the backend and frontend components of the portal.
- **Reporting Tools:** Integrate advanced reporting tools to enable the generation of detailed and customizable reports on faculty activities, approvals, and overall cluster performance.

User Personas

1. **Faculty:**
   - Uploads work logs with details and documents.
   - Tracks deadlines and manages workload.
   - Views feedback on submitted work logs.
2. **Head:**
   - Reviews and approves/rejects faculty work logs.
   - Provides feedback on work logs.
   - Generates reports on faculty workload and performance.
3. **Admin:**
   - Manages user accounts (faculty and Heads).
   - Monitors overall faculty work log activity of both faculties and heads.
   - Generates reports on faculty workload and performance of all the faculties.

User Stories

- As a Faculty, I have to upload my daily worklogs, which I've done with desired timings.
- As a Head, I've to Approve or Rejects the faculty's worklogs and notify the balance works to all the faculties.
- As a Admin, I've to view all the works of both Faculties and Admins and also generate the reports.

Functional Requirements

1) User Registration and Authentication:

- Users, including faculty members, heads and admin, must be able to register and authenticate themselves on the platform.
- The Admin will manage user accounts, including faculty and head accounts, ensuring proper role-based access.

2) Worklog Upload Portal

- Faculty members will upload work logs with details and supporting documents.
- Faculty members can track deadlines, manage their workload, and view feedback on their submitted work logs.

3) Worklog Approval/Rejection mechanism

- The Head of each cluster will review required worklogs submitted by faculty members.
- Heads have the authority to approve or reject the worklogs based on predefined criteria and suggestions from domain experts.
- Admins can view a list of approved worklogs on the portal.

4) Head responsibilities

- Heads will review, approve, or reject faculty work logs.
- Heads will provide feedback on submitted work logs and notify the balanced worklog to be uploaded.

5) Admin responsibilities

- Admin will manage user accounts for faculty and heads.
- Admin will monitor overall faculty work log activity.
- Admin will generate comprehensive reports across the entire system.

Non-functional Requirements

1) Scalability:

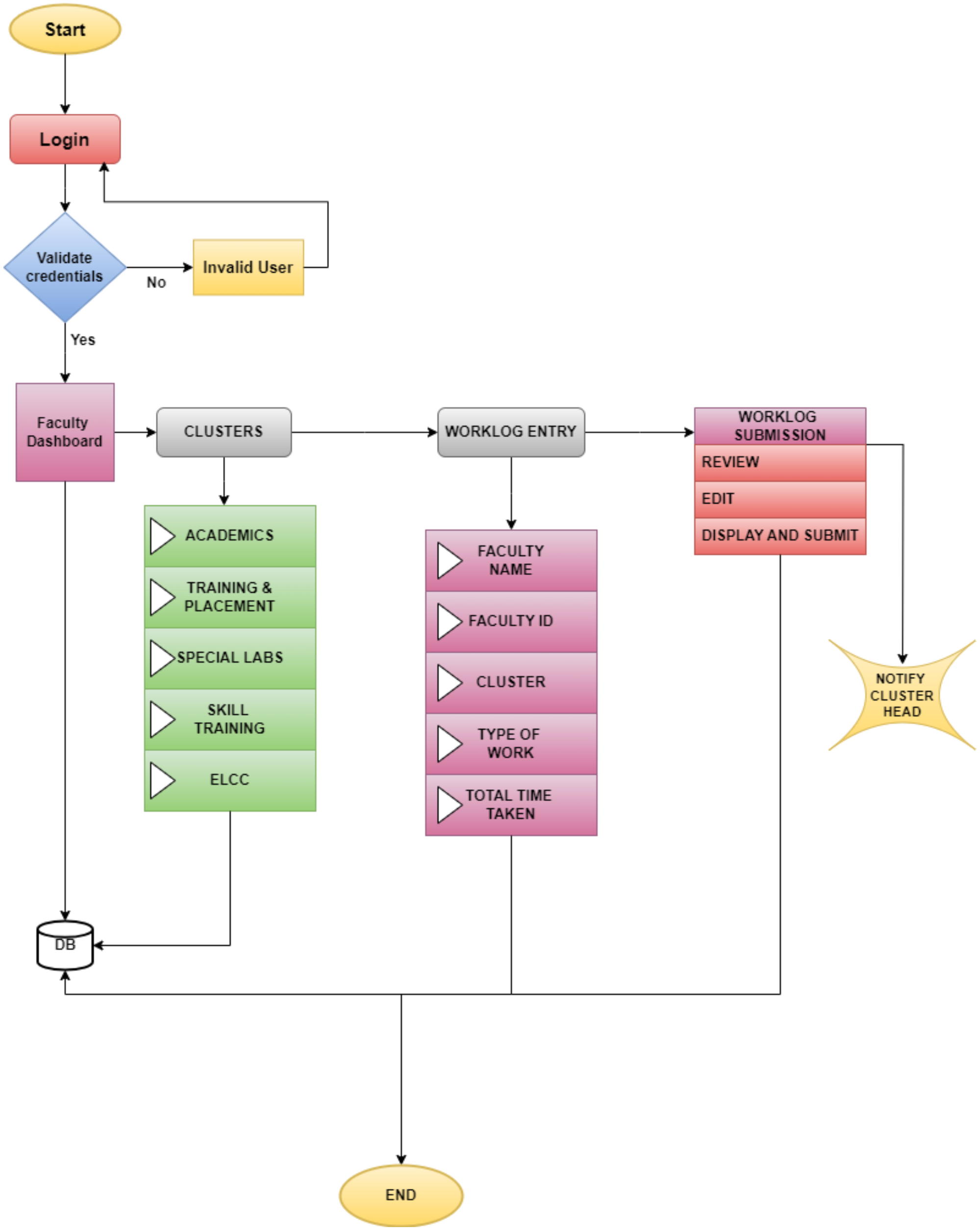
- The platform must be designed to efficiently handle a growing number of users, including Faculty, Heads and Admin roles.
- It should be capable of managing an increasing volume of work logs without compromising performance.
- Heads should have the capability to add, edit, and manage approved events seamlessly, even as the number of worklogs grows.

2) Privacy and Data Security:

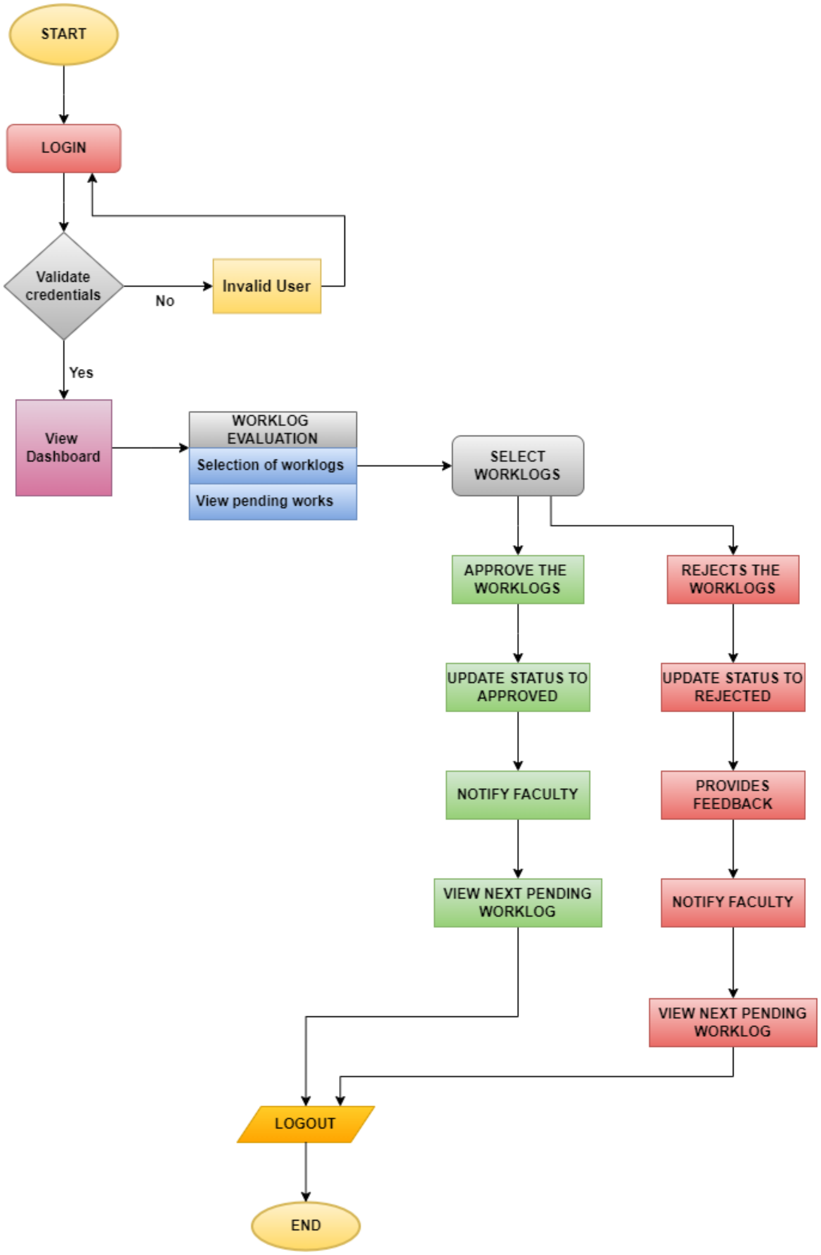
- The platform must ensure the confidentiality and integrity of user data, worklog submissions and status of approvals.
- Robust security measures should be implemented to protect the privacy of Faculty, Heads and Admin users.
- Data encryption, secure access controls, and regular security audits should be employed to safeguard sensitive information.

FLOW CHART

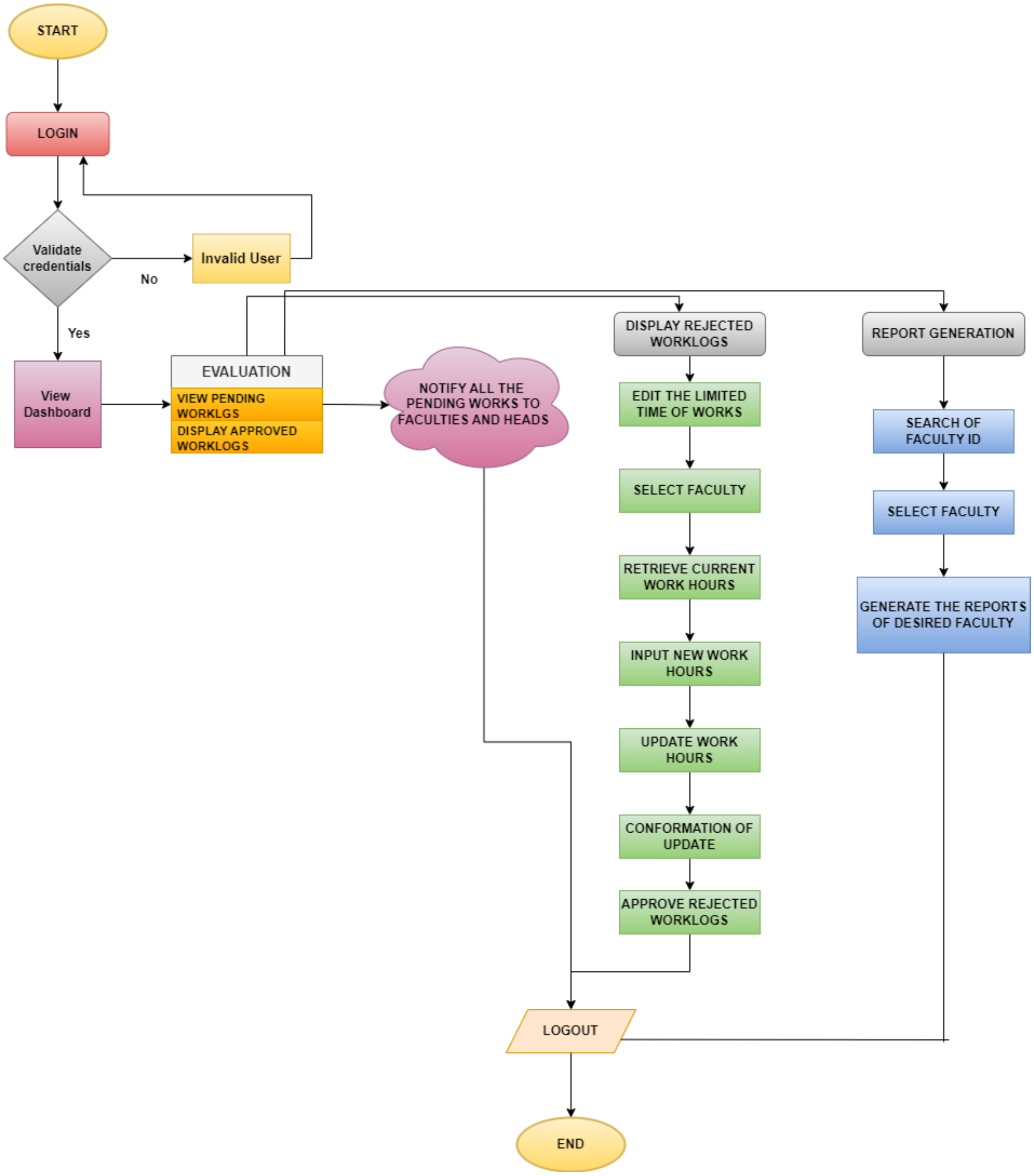
FACULTY:



HEAD:


START
LOGIN
Validate credentials
No
Invalid User
Yes
View Dashboard
WORKLOG EVALUATION
Selection of worklogs
View pending works
SELECT WORKLOGS
APPROVE THE WORKLOGS
REJECTS THE WORKLOGS
UPDATE STATUS TO APPROVED
UPDATE STATUS TO REJECTED
NOTIFY FACULTY
PROVIDES FEEDBACK
VIEW NEXT PENDING WORKLOG
NOTIFY FACULTY
VIEW NEXT PENDING WORKLOG
LOGOUT
END

ADMIN:


START
LOGIN
Validate credentials
No
Invalid User
Yes
View Dashboard
EVALUATION
VIEW PENDING WORKLGS
DISPLAY APPROVED WORKLOGS
NOTIFY ALL THE PENDING WORKS TO FACULTIES AND HEADS
DISPLAY REJECTED WORKLOGS
EDIT THE LIMITED TIME OF WORKS
SELECT FACULTY
RETRIEVE CURRENT WORK HOURS
INPUT NEW WORK HOURS
UPDATE WORK HOURS
CONFORMATION OF UPDATE
APPROVE REJECTED WORKLOGS
REPORT GENERATION
SEARCH OF FACULTY ID
SELECT FACULTY
GENERATE THE REPORTS OF DESIRED FACULTY
LOGOUT
END