Software Requirement Specification for Project Portal

NAME:	SOUNDARYA K
ROLL NO:	7376221CD145
PROJECT ID:	19
PROJECT TITLE:	Manage and update faculty's all works

1. PROBLEM STATEMENT:

Build a portal to manage and update the faculty work.

2.INTRODUCTION:

2.1. PURPOSE:

The purpose of managing and updating the works of faculty members encompasses several key objectives that are vital for the efficient functioning and growth of educational institutions.

2.2.SCOPE OF PROJECT:

The scope of this project involves developing a system to manage and update faculty work, professional development activities. The system will feature user profiles, automated.It aims to enhance communication, streamline task management, ensure data security, and provide user training and technical support. This user-friendly, efficient, and secure system will significantly improve faculty work tracking and management, contributing to the institution's overall effectiveness.

3. FUNCTIONAL REQUIREMENTS:

3.1.User Authentication:

- Secure login for faculty, head, and admin.

3.2.Faculty Dashboard:

- Ability for faculty to create and update professional works.
- Faculty can view their schedules as approved or rejected.
- If they reject faculty work, they get a rejection notification with feedback.

3.3.Heads Dashboard:

- Heads check their schedules.
- Heads can approve or reject the faculty's works.

3.4.Admins Dashboard:

- Admins can view faculty work reports and check whether it's approved or rejected .

4. NON FUNCTIONAL REQUIREMENTS:

4.1.Performance: The system must respond to user actions within 2 seconds to ensure efficient usability and must handle a concurrent user load of at least 100 users without significant performance degradation.

4.2.Scalability: The system should be designed to accommodate an increasing number of users and data volume over time, and it should be scalable to support additional features and functionalities as per future requirements.

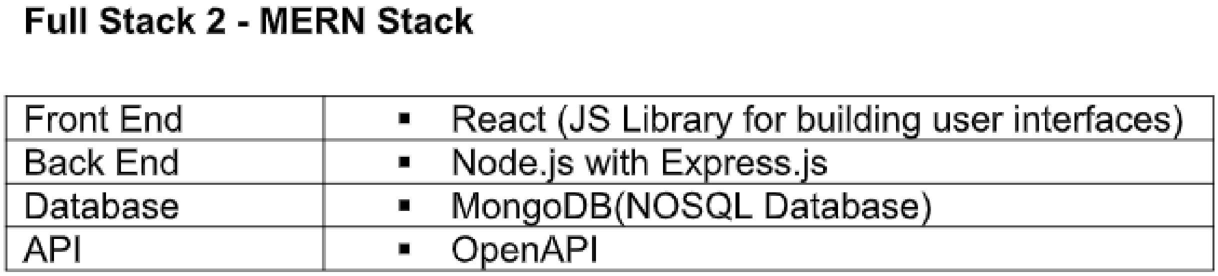
4.3.Reliability: Ensure 99.9% uptime to minimize disruptions in service availability.Implement redundancy and failover mechanisms to maintain functionality during hardware or software failures.

4.3.Security: Implement strong authentication and authorization mechanisms to protect sensitive data.Encrypt data at rest and in transit using industry-standard encryption methods.Regularly update and patch the system to protect against vulnerabilities.

4.4.Maintainability: The system should be designed with modularity to allow for easy updates and maintenance.Maintain a well-documented codebase to facilitate troubleshooting and future enhancements.

4.5.Usability: The interface should be intuitive and user-friendly, requiring minimal training for faculty and administrators.Provide comprehensive help documentation and support resources.

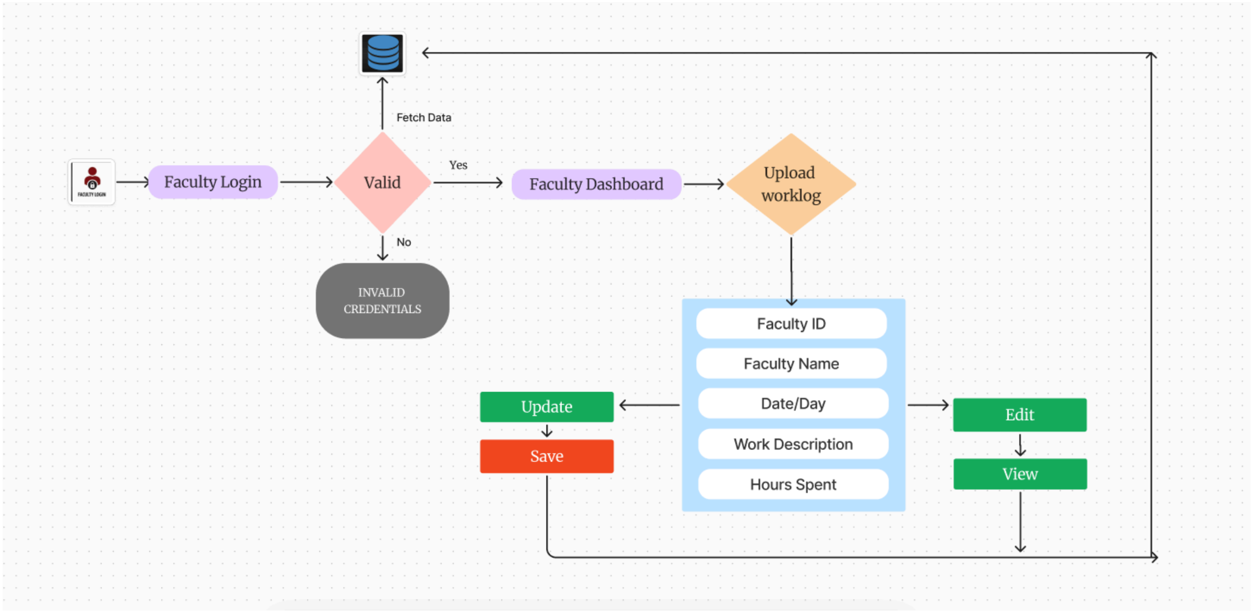
STACK:



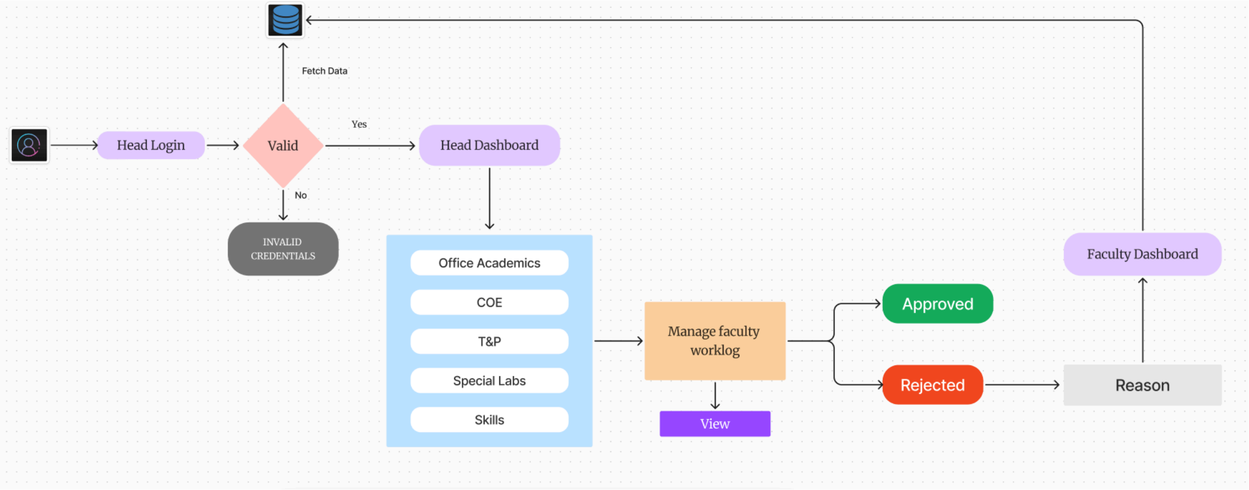
Full Stack 2 - MERN Stack

Front End	▪ React (JS Library for building user interfaces)
Back End	▪ Node.js with Express.js
Database	▪ MongoDB(NOSQL Database)
API	▪ OpenAPI

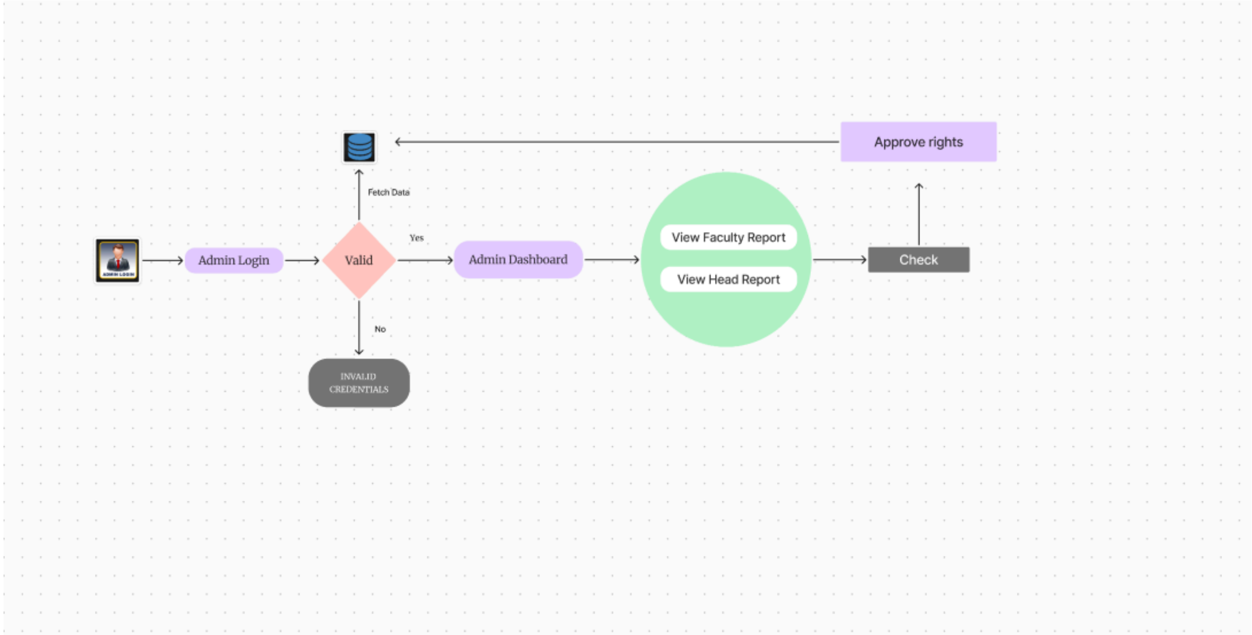
FACULTY INTERFACE:



HEAD INTERFACE:



ADMIN INTERFACE:


Approve rights
Fetch Data
Yes
Admin Login
Valid
Admin Dashboard
View Faculty Report
View Head Report
Check
No
INVALID CREDENTIALS