**CE431: DBMS**

**Project Proposal**

**1** **ABSTRACT**

This document explains our project proposal for the Database Management Systems (DBMS) course. The proposal is about developing a Campus management system commonly referred to as CMS to be used by a university or college to handle details about students, faculty members, academics and exams. The developed CMS will be a web application having backend developed in Node JS and frontend in React JS. The application will use MySQL relational database management system. All technologies to be used and the problem statement is explained in detail below.

**2 INTRODUCTION**

**Campus Management System (CMS)** deals with all kind of student details, academic related reports, institute details, course details, curriculum, batch details and other resource related details too. It tracks all the details of a student from the day one to the end of his course which can be used for all reporting purpose, tracking of attendance, progress in the course, completed semesters years, coming semester year curriculum details, exam details, project or any other assignment details, final exam result; and all these will be available for future references too. Our application will have the databases of Courses offered by the institute under all levels of graduation or main streams, teacher or faculty's details, batch execution details, students' details in all aspects. This application can facilitate us explore all the activities happening in the institute, even we can get to know which teacher / faculty is assigned to which batch, the current status of a batch, attendance percentage of a batch and upcoming requirements of a batch. Different reports and Queries can be generated based of vast options related to students, batch, course, teacher / faculty, exams, semesters, certification and even for the entire institute.

**3** **PROBLEM STATEMENT**

This CMS will replace old manual management systems used that are prone to errors, have no backup, are time consuming and no analytics can be generated from them. This system will require less manpower and maintain accuracy. The system can be accessed by every students/faculties/employees of the institution through internet connected computers or internet enabled mobile devices with the aid of their user name and password. Every user will have a customized home page with his/her profile management facilities. Users can access different options of the application through links displayed on the home page.

**4** **OUTCOME REQUIREMENTS**

We have a very versatile range of technologies that can be used to address the problem. It can be concluded from the formal problem statement; we need technologies capable of performing the tasks below:

1. We need a platform that is easily accessible
2. We need a database server which can hold the records of Students, Teachers and Courses
3. An interface that can communicate with the server and present the information in a user-friendly way.

**5** **PROPOSED TECHNOLOGIES**

We are developing a web-based application and for that we have selected the following technologies:

1. Node.js + Express.js for Backend
2. React.js for frontend
3. MySQL
4. A Database Server
5. Sequelize ORM
6. GitHub

**6** **Feasibility and Defence of Proposed Technologies:**

**6.1 Node.js + Express.js**

Node.js will be our application's back-end platform that will run our application on the Cloud Server. Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. The logical reasons for selecting it are as follows:

* **Robustness:** Node.js is an incredibly robust and production-ready platform, with millions of developers trusting it across thousands of reputable technology companies.
* **Speed:** Node.js is very efficient and feasible. Node uses a single-threaded, non-blocking server module that can quickly scale up to increasing levels of use. Node is very powerful and feasible compared to other HTTP or REST servers, such as Apache or Flask.

**6.2 React.js**

React is an open-source, front end, JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used for building fast and interactive user interfaces for web and mobile applications. The logical reasons for selecting it are as follows:

* **Performance:** React uses Virtual DOM, thereby creating web applications faster. Virtual

DOM compares the components’ previous states and updates only the items in the Real

DOM that were changed, instead of updating all the components again, as conventional web applications do.

* **Reusable components:** Components are the building blocks of any React application, and a single app usually consists of multiple components. These components have their logic and controls, and they can be reused throughout the application, which in turn dramatically reduces the application’s development time.
* **Easy creation of dynamic applications:** React makes it easier to create dynamic web applications because it requires less coding and offers more functionality, as opposed to JavaScript, where coding often gets complex very quickly.

**6.3** **MySQL**

MySQL is a relational database management system (RDBMS) based on the SQL (Structured Query Language) queries. It is one of the most popular languages for accessing and managing the records in the database. The following are the most important features of MySQL:

* **Security:** MySQL consists of a solid data security layer that protects sensitive data from intruders. Also, passwords can be encrypted in MySQL.
* **Scalability:** MySQL supports multi-threading that makes it easily scalable. It can handle almost any amount of data, up to as much as 50 million rows or more. The default file size limit is about 4 GB. However, we can increase this number to a theoretical limit of 8 TB of data.
* **Speed:** MySQL is considered one of the very fast database languages, backed by a large number of the benchmark test.
* **Flexibility:** MySQL supports a large number of embedded applications, which makes MySQL very flexible.

**6.4 Sequelize**

Sequelize is a powerful library in JavaScript that makes it easy to manage a SQL database. Sequelize can layer over different protocols, but here we will use MySQL. At its core, Sequelize is an Object-Relational Mapper – meaning that it maps an object syntax onto our database schemas. Sequelize uses Node.JS and JavaScript’s object syntax to accomplish its mapping.

**7** **Tentative Activity Plan:**

The following table breaks down our activity plan for this project:

|  |  |  |
| --- | --- | --- |
| Activity | Time Allocated | |
|  |  |  |
| Learning Node, Express and React | 3 | Weeks |
|  |  | |
| Designing Minimal Front End of Web App | 1 Week | |
|  |  |  |
| Designing Back End with Database | 1 | Week |
|  |  |  |
| Debugging and Refactoring | 3 | Days |
|  |  |  |
| Making Interactive UI of Web App | 3 | Days |
|  |  | |
| Testing Web App | 1 Day | |
|  |  |  |
| Finalizing Web App | 1 | Day |
|  |  |  |