# A REPORT ON

**DBMS MINI PROJECT ON**

# " T-Shirt Lelo.com"

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR:

**DATABASE MANAGEMENT SYSTEMS LAB (DBMSL) BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)**

**SUBMITTED BY**

Shreyash Bachate (31303)

Gaurav Chawda (31307)

Prajwal Kakade (31318)

Yash Kalavadiya (31319)



**DEPARTMENT OF COMPUTER ENGINEERING**

**PUNE INSTITUTE OF COMPUTER TECHNOLOGY**

DHANKAWADI, PUNE – 43

**SAVITRIBAI PHULE PUNE UNIVERSITY 2022 -2023**

## Title:

DBMS Mini Project on “T-Shirt Lelo.com”

## Assignment No:

C-1 (13)

## Problem Definition:

Using the database concepts covered in Group A and Group B, develop an application with the following details:

1. Follow the same problem statement decided in Assignment-1 of Group A.
2. Follow the Software Development Life Cycle and other concepts learnt in Software Engineering Course throughout the implementation.
3. Develop application considering:
   * Front End: React.Js
   * Back end: MongoDB/Mongoose/Node.Js/Express.Js

## Project Objectives:

1. To design and model the database structure using E-R modeling concepts and provide an overview of the project scope for “PICT Student and Faculty Achievement Tracking System”.
2. To maintain a tracking system for students’ achievements relating to the following domains:
   * Internships
   * Social Work
   * Sporting Achievements
   * Project Competitions and Hackathons
   * Research Publications
3. To maintain a tracking system for faculty achievements relating to the following domains:
   * Research Publications
   * Conferences attended
   * Paper Presentations
   * Sponsored Projects/ Projects with Grant
   * Collaboration Projects
4. To develop a user-friendly and efficient module to generate filtered reports according to conditions specified in input forms.

## Requirements:

1. Hardware Requirements:
   * PC with a latest 64-bit Operating System configuration.
   * 8 G.B. RAM and 512 G.B. HDD/SDD
   * 15” Color Monitor, Keyboard, Mouse
2. Software Requirements:
   * VS Code- Support for React.Js and Node.Js
   * MongoDB No-SQL database management system

## Project Description:

1. Entity Design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Entity** | **Type** | **Primary Key** | **Attributes** |
| **1** | Student | Strong | ID | Name, Email\_ID, DOB, Age, Roll\_No, Class\_ID, Class, Department, Mobile, Password |
| **2** | Faculty | Strong | Teacher\_ID | Name, Mobile, Address, Department, DOB, Age, Email\_ID |
| **3** | Admin | Strong | Admin\_ID | Name, Email\_ID, Mobile, Password |
| **4** | Research\_Paper | Strong | Title | Domain, Guide, Organization, Date |
| **5** | Conference\_Report | Strong | UID | Activity, Title, Speaker, Participants, Date |
| **6** | Project\_Report | Strong | UID | Activity, State, Sponsor, Participants, Date, Co- Ordinator |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **7** | Collab\_Projects | Strong | UID | Project\_Name, Collaborators, Sponsors, Date |
| **8** | Internships | Weak | Company (discriminator) | Stipend, Duration, Address, Date\_Join, Description, Date\_End, Job\_Title, Type, Source, Approve |
| **9** | Social\_Work | Weak | Work\_Title (discriminator) | Organisation, ClassID, Date, Nature, Description, Venue |
| **10** | Sport\_Achievements | Weak | Achievement (discriminator) | Sport\_Name, Description, Venue, Date |
| **11** | Project\_Competition | Weak | Title (discriminator) | Achievement, Domain, Type, Description, Date, Venue |

1. Relationship Modelling
   1. Student: completes  Internship

Takes\_part  Social Work

Achieves  Sporting\_Achievements Wins Project\_Competition Publishes  Research Paper

* 1. Admin: manages Student

Manages  Faculty Generates  Project\_Report Generates  Collab\_Projects

Generates  Conference\_reports

* 1. Faculty: assigned  Designation

Publishes  Research Paper Attends  Conference

1. Implementation Details
   1. Front-End:

The interface for the project is developed using React Js. React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.

Declarative views make your code more predictable and easier to debug.

Build encapsulated components that manage their own state, then compose them to make complex UIs.

Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.

We don’t make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code.

React can also render on the server using Node and power mobile apps using React Native.

* 1. Back-End:

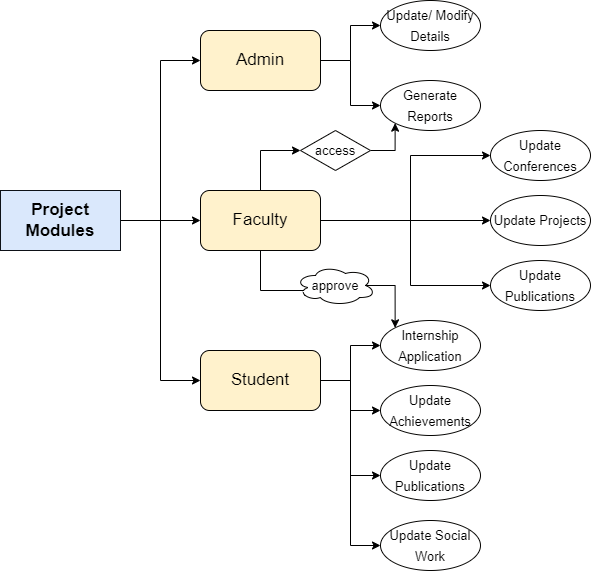
The system is designed to work with a MongoDB database back-end to store all the

Products, Orders, Users, Categories data free and open-source. MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License which is deemed non-free by several distributions.

For Handling Backend Workloads we are using Node js to manage routes and manipulate database based on routes invoked.

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on a JavaScript Engine and executes JavaScript code outside a web browser, which was designed to build scalable network applications.

PROJECT MODULE DESCRIPTION



## Screenshots of Implementation:

Future Scope:

This project is currently limited to T-Shirts Only. In future we can expand Products Range like other fashions including but not limited to shirts, denims.

For Implementing Future Scope we will be required to change Products Schema, And Also might need to create new schemas.

As project is fully secured in terms of authentication, there is no future scope for adding extra layers of security. But we can indeed add OAuth like Google, Apple, Facebook.

Right now admin can only perform CRUD Operations on all schemas, but when user base grows we can add google analytics to view and analyze how user interact with our page.

## Conclusion:

Thus, through this project, we have successfully:

1. Designed and modelled the database structure and provided a definite project scope for “T-Shirt Lelo.com”.
2. Developed user-friendly and efficient modules to generate filtered reports according to conditions specified in input forms.
3. Designed UI Dashboard for User as well as for Admin.