



**COLLEGE CODE : 9606**

**COLLEGE NAME : DMI ENGINEERING COLLEGE**

**DEPARTMENT : COMPUTER SCIENCE  
ENGINEERING**

**STUDENT NM-ID :  
75D532AEA2B107197D94D40EC909FCC4**

**ROLL NO : 960623104070**

**DATE : 14.10.2025**

**Completed the project named as  
Phase -V**

**TECHNOLOGY PROJECT**

**NAME: IBM-FE-AngularJS with SQL Integration**

**SUBMITTED BY,**

**NAME:M. Sukanya  
MOBILE NO: 7200836423**

## Phase 5 – Project Demonstration & Documentation

### 1. Final Demo Walkthrough

The Student Management System is a simple web-based application built using AngularJS for the front end and SQL for the backend database.

It allows users to:

- \* Add new student records (Name, Age, Department, Email, Phone)
- \* View the list of students
- \* Edit or delete student details
- \* Data is stored and fetched from a local SQL database

#### Demo Flow:

1. Open the app in the browser.
2. Fill out the form with student details.
3. Click “Add Student” to save data.
4. View all stored students in a table format.

5. Update or delete records as needed.

## 2. Project Report

Title: Student Management System

Technology Used:

Frontend: HTML, CSS, AngularJS

Backend: SQL (local server)

language: JavaScript

Database: MySQL or SQLite

Objective:

To create a simple and efficient interface to manage student data using AngularJS as the front-end framework and SQL for persistent data storage.

Features:

Add, view, update, delete student records

Form validation using AngularJS, Dynamic table rendering

Local SQL database connection

Outcome:

A functional web app that allows seamless student record management using AngularJS and SQL.

### 3. Screenshots / API Documentation

Screenshots to include:

The image displays four screenshots of the REVA University Student Portal. The top-left screenshot shows the university's logo and name, 'REVA UNIVERSITY Bangalore, Karnataka', along with the tagline 'Technology-based Education | 15,000+ Students | 45 Acres Campus'. The top-right screenshot is the 'Welcome to Student Portal' login page, featuring fields for 'Student ID' and 'Password', a 'Remember me' checkbox, and buttons for 'Login', 'Forgot Password?', and 'New Student? Register Here'. The bottom-left screenshot is the 'Student Registration' page, which includes a 'Personal Information' section with fields for 'Full Name \*' (Sukanya), 'Email Address \*' (suganyam728@gmail.com), 'Phone Number \*' (9952886423), 'Date of Birth \*' (22/12/2005), and 'Gender'. The bottom-right screenshot is the 'Login Credentials' section, showing the 'Student ID' (R2025004) and 'Create Password \*' and 'Confirm Password \*' fields. It includes a password strength indicator (Fair), a 'Passwords do not match' error message, and a checkbox for 'I agree to the Terms and Conditions \*'. Navigation buttons for 'Back to Login' and 'Register' are also present.

### API (if applicable):

| Endpoint                         | Method | Description               |
|----------------------------------|--------|---------------------------|
| <code>/addStudent`</code>        | POST   | Adds a new student record |
| <code>/getStudents`</code>       | GET    | Fetches all students      |
| <code>/updateStudent/:id`</code> | PUT    | Updates student details   |
| <code>/deleteStudent/:id`</code> | DELETE | Removes a student         |

### 4. Challenges & Solutions

| Challenge                         | Solution  |
|-----------------------------------|---|
| Connecting AngularJS with SQL     | Used local Node.js/Express middleware for SQL queries                               |
| Data not refreshing automatically | Implemented AngularJS <code>`\$scope`</code> and <code>`\$apply()`</code> functions |
| Validation issues                 | Added AngularJS form validation and error messages                                  |
| Deployment server setup           | Used local server with XAMPP/Node   |

## 5. GitHub README & Setup Guide

### Setup Instructions:

1. Clone the repository

Git clone <https://github.com/username/student-management-system.git>

2. Install dependencies (if Node used):

Npm install

3. Setup database:

\* Create a SQL database named `studentdb`.

\* Run `student.sql` file to create table structure.

4. Start local server

Node server.js

4. Open `index.html` in your browser to view the app.

## 6. Final Submission (Repo + Deployed Link)

GitHub Repository: [https://github.com/m-sukanya/NM\\_phase5.git](https://github.com/m-sukanya/NM_phase5.git)

Deployed Link (if any): [https://m-sukanya.github.io/NM\\_phase5/](https://m-sukanya.github.io/NM_phase5/)