Rapid-Review: A Digital Solution for Feedback Management

INNOVATIVE / MULTI-DISCIPLINARY PROJECT REPORT
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE
DEGREE OF **BACHELOR OF ENGINEERING** IN
COMPUTER SCIENCE AND ENGINEERING
OF THE ANNA UNIVERSITY

INNOVATIVE PROJECT
April/May 2025

PROJECT WORK

BATCH 2023 – 2027

Submitted by

SABARI M	**-**	**23CS132**
SANTHOSHKUMAR S	**-**	**23CS143**
SHANTHISH S R	**-**	**23CS152**
YOGENDRA N	**-**	**23CS195**

Under the Guidance of
Ms. M. ABINAYA ME., (Ph.D)
ASSISTANT PROFESSOR
COMPUTER SCIENCE AND ENGINEERING

Sri Eshwar College of Engineering
(An Autonomous Institution)
Kinathukadavu (Tk), Coimbatore - 641 202, Tamil Nadu
Approved by AICTE, New Delhi and Affiliated to Anna University, Chennai

BONAFIDE CERTIFICATE

Certified that this Report titled **Rapid-Review: A Digital Solution for Feedback Management"** is the bonafide work of

SABARI M	**23CS132**
SANTHOSHKUMAR S	**23CS143**
SHANTHISH S R	**23CS152**
YOGENDRA N	**23CS195**

who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. R. Subha, M.E, Ph.D.	Ms. M. Abinaya M.E,(Ph.D)
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Computer Science and Engineering,	Assistant Professor,
Sri Eshwar College of Engineering,	Computer Science and Engineering,
Coimbatore – 641 202.	Sri Eshwar College of Engineering,
	Coimbatore – 641 202.

Submitted for the **Autonomous Semester End Innovative / Multi-Disciplinary Project Viva-Voce** held on………………….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

Leadership & Excellence
Sri Eshwar College of Engineering
Coimbatore | Tamilnadu
An Autonomous Institution
Affiliated to Anna University, Chennai
NAAC NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL
Accredited by NAAC with 'A' Grade
NBA NATIONAL BOARD of ACCREDITATION
nirf National Institutional Ranking Framework
All India Council for Technical Education AICTE

QUALITY POLICY

To establish a system of Quality Enhancement, which would on a continuous basis evaluate and enhance the quality of teaching – learning, research and extension activities of the institution, leading to improvements in all processes, enabling the institution to attain excellence.

INSTITUTE VISION

To be recognized as a premier institution, grooming students into globally acknowledged engineering professionals.

INSTITUTE MISSION

- Providing outcome and value-based engineering education
- Nurturing research and entrepreneurial culture
- Enabling students to be industry ready and fulfill their career aspirations
- Grooming students through behavioral and leadership training programs
- Making students socially responsible

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT VISION

To groom students into globally competent software professionals and meet the ever changing requirements of the industry.

DEPARTMENT MISSION

- Creating a quality academic environment with relevant IT infrastructure and empowering faculty and students with emerging technologies.
- Motivating staff and students to actively involve in lifelong learning and fostering research.
- Inculcating leadership and entrepreneurship skills in students.
- Generating opportunities for students to evolve as competent software professionals with societal consciousness.

PROGRAM EDUCATIONAL OBJECTIVES

PEO1: To prepare graduates for a career in software engineering

PEO2: To prepare students for higher studies, research, entrepreneurial and leadership roles by imparting the quality of lifelong learning

PEO3: To enable students to apply innovative solutions for real-life problems in computer science domain.

PROGRAM OUTCOMES

PO1: Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)

PO8: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one’s own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)

PROGRAM SPECIFIC OUTCOMES

PSO1. Demonstrate knowledge in open source technologies

PSO2. Develop innovative solutions by adapting emerging technologies for industry oriented applications.

PSO3. Implement SDLC principles for project/product development.

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

We thank God for his grace and blessing that have been showered upon us to bring success for this project. The success we have accomplished in our project would not have been possible without the timely help and guidance rendered to us by many people. We take this opportunity to thank them all.

We wish to express our deep gratitude **Mr. R. Mohanram**, Chairman, Sri Eshwar College of Engineering, Kondampatty for being kindly permitted us to carry on the project. He has an abiding interest in our wellbeing.

We are grateful to **Dr. Sudha Mohanram M.E., Ph.D** Principal, Sri Eshwar College of Engineering, Kondampatty for having made all provision available to us.

We are thankful to **Mr. R. Rajaram**, Director, Sri Eshwar College of Engineering, Kondampatty for all his support to complete the project effectively.

We would like to express our heartfelt gratitude and humble thanks to Head of the Department **Dr. R. Subha M.E., Ph.D** for having facilitated us to complete the project successfully.

With the deep sense of gratitude, we extend our earnest and sincere thanks to our project guide **Ms. M. Abinaya M.E., (Ph.D)** Assistant Professor, Department of Computer Science and Engineering, for his kind suggestion and co-operation for doing the project. Finally, we would like to extend our thanks to our teaching and non- teaching staffs those who helped us in bringing out project successfully.

TABLE OF CONTENTS

ABSTRACT

RapidReview is a web-based platform developed to streamline student feedback collection during academic training sessions by enabling educational institutions to manage colleges, tutors, and feedback forms through a centralized system. The application allows administrators to log in, create feedback events, and share a single persistent link with students to gather input across multi-day sessions. It features a role-based dashboard for both students and administrators, ensuring transparency, accountability, and efficient feedback processing. A unique identity-reveal mechanism enhances feedback quality by keeping submissions anonymous until the final session day.

Built using the MERN stack (MongoDB, Express.js, React.js, and Node.js), RapidReview offers a responsive front-end, scalable back-end processing, and secure data storage. Each major function, such as feedback form creation, submission tracking, and analysis, is modularized for flexibility and ease of maintenance. The system is designed with a clean, intuitive interface to ensure a smooth user experience across devices. RapidReview lays a strong foundation for future expansion, including features like AI-based feedback insights, LMS integration, and mobile app support.

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 OBJECTIVES

The primary objective of the RapidReview project is to develop a web-based application that simplifies student feedback collection for academic training programs. It provides institutions with a structured platform to gather insights from students regarding sessions and tutors through a single shareable link. The system aims to ensure honest input by maintaining anonymity during initial sessions and associating identities only on the final day.

Additionally, the system enhances coordination between students and administrators using role-based dashboards and real-time feedback access. With centralized and secure data storage, it ensures reliable management of college profiles, tutor information, and feedback logs. This project ultimately aims to enhance the training experience by making feedback collection more efficient, transparent, and student-friendly.

1.2 SCOPE OF THE PROJECT

The RapidReview system is designed to streamline the feedback management process through an interactive and user-friendly web platform. Its scope includes enabling administrators to register colleges and tutors, create custom feedback forms, and manage feedback sessions over multiple days using a single persistent link. The system supports real-time tracking and identity control for each feedback submission.

It also features role-based access for administrators and students, allowing seamless feedback form access and submission management. All data, including student responses, session analytics, and tutor performance insights, is securely stored in a scalable database for future analysis. With built-in adaptability, the platform can be extended to include features like AI-based feedback analysis, LMS integration, and mobile app support. RapidReview aims to offer a reliable, insightful, and privacy-conscious solution for academic feedback collection.

CHAPTER 2

SYSTEM ANALYSIS AND SPECIFICATION

2.1 PROBLEM DESCRIPTION

Hostel Educational institutions often face difficulties in efficiently collecting and managing student feedback for academic training sessions. Traditional methods such as paper-based forms or scattered digital surveys result in delayed insights, low engagement, and a lack of actionable data. These outdated systems hinder continuous improvement and make it difficult for administrators to gauge the effectiveness of training programs or tutor performance.

Furthermore, students may be reluctant to provide honest feedback due to concerns over identity disclosure, especially in ongoing multi-day sessions. Without a unified platform, managing multiple training sessions, feedback entries, and participant details becomes cumbersome for administrators. There is also a lack of real-time feedback visibility and structured data analysis.

The **RapidReview** platform addresses these challenges by offering a smart, web-based solution designed specifically for training-based feedback collection. It allows administrators to manage colleges, tutors, and feedback sessions in one place, while students can easily submit feedback through a single persistent link. With anonymity control, real-time analytics, and role-based dashboards, the system improves transparency, accountability, and feedback quality—ultimately enhancing the overall training experience in academic environment.

2.2 FUNCTIONAL REQUIREMENT

2.2.1 USER AUTHENTICATION

- Implement secure login for administrators using credentials (email and password).
- Enable role-based access to manage functionalities for admin and student users.

2.2.2 FEEDBACK FORM CREATION AND LINK SHARING

- Allow admins to create training-specific feedback forms by selecting tutors, colleges, and date ranges.
- Generate a unique persistent link for each training session for easy sharing and reuse.

2.2.3 FEEDBACK SUBMISSION AND TRACKING

- Enable students to submit feedback anonymously for initial training days.
- Automatically reveal identities only on the final day for accountability.
- Display real-time feedback data to admins for daily tracking.

2.2.4 ADMIN DASHBOARD

- Centralized admin dashboard to manage colleges, tutors, and session.
- View, filter, and analyze feedback based on tutor, department, or date.

2.2.5 DATA STORAGE AND SECURITY

- Store feedback data, user profiles, and form configurations securely in MongoDB.
- Use JWT-based authentication and bcrypt for password hashing.

2.2.6 FEEDBACK ANALYSIS AND REPORTING

- Visualize feedback trends using graphs and statistics.
- Enable filtering by day, session, tutor, or department for targeted insights.

2.2.7 FUTURE ENHANCEMENTS AND SCALABILITY

- Plan for mobile application support and LMS (Learning Management System) integration.
- Integrate AI-based analytics for sentiment detection and performance tracking.

2.3 SOFTWARE AND HARDWARE REQUIREMENTS

2.3.1 HARDWARE REQUIREMENTS:

• Minimum: Intel i3 processor, 8 GB RAM
• Recommended: Intel i5 or higher, 16 GB RAM for faster performance

2.3.2 SOFTWARE REQUIREMENTS:

• Node.js and Express.js (for backend server)
• MongoDB (for database storage)
• React.js (for frontend development)
• Any design tool like Figma or Creately (for UI/UX and architecture planning)

2.4 NON-FUNCTIONAL REQUIREMENT

2.4.1 PERFORMANCE

The system should deliver fast responses for actions like feedback submission, dashboard rendering, and session analytics. It must maintain optimal performance with multiple concurrent users through efficient API handling and optimized database queries.

2.4.2 SCALABILITY

RapidReview should scale to handle an increasing number of users, feedback entries, and sessions. The architecture must support future enhancements such as AI integration or mobile app development with minimal codebase changes.

2.4.3 RELIABILITY & AVAILABILITY

The platform should ensure high availability and consistent uptime. Backup mechanisms, error handling, and monitoring tools must be in place to maintain data integrity and uninterrupted service delivery.

2.4.4 SECURITY

The application includes proper authentication measures to prevent unauthorized access by using secure login mechanisms. It employs techniques such as JWT and password hashing to safeguard user credentials.

CHAPTER 3

SYSTEM DESIGN

USECASE DIAGRAM

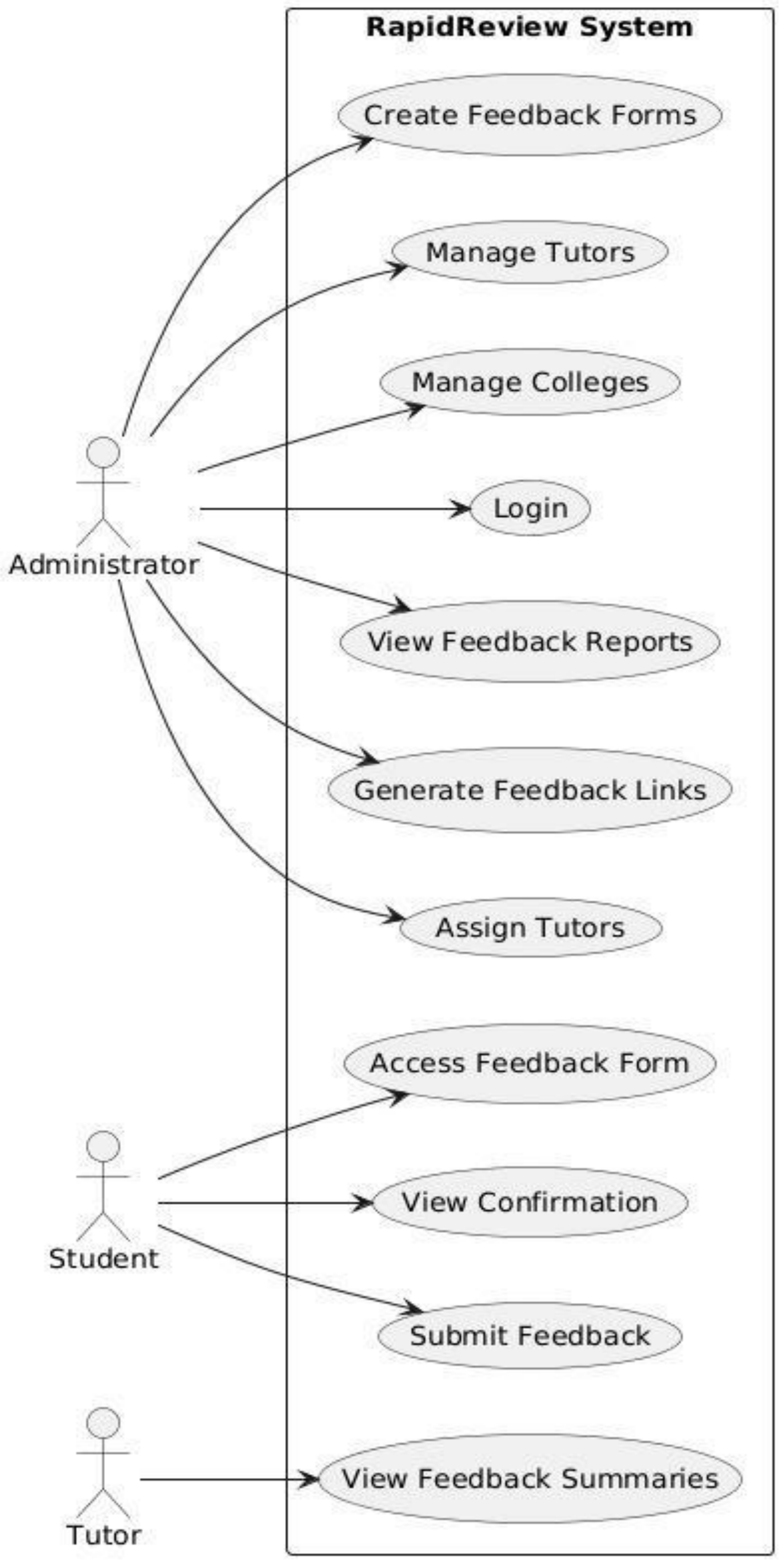


Fig:3.1 Usecase Diagram

FLOW DIAGRAM

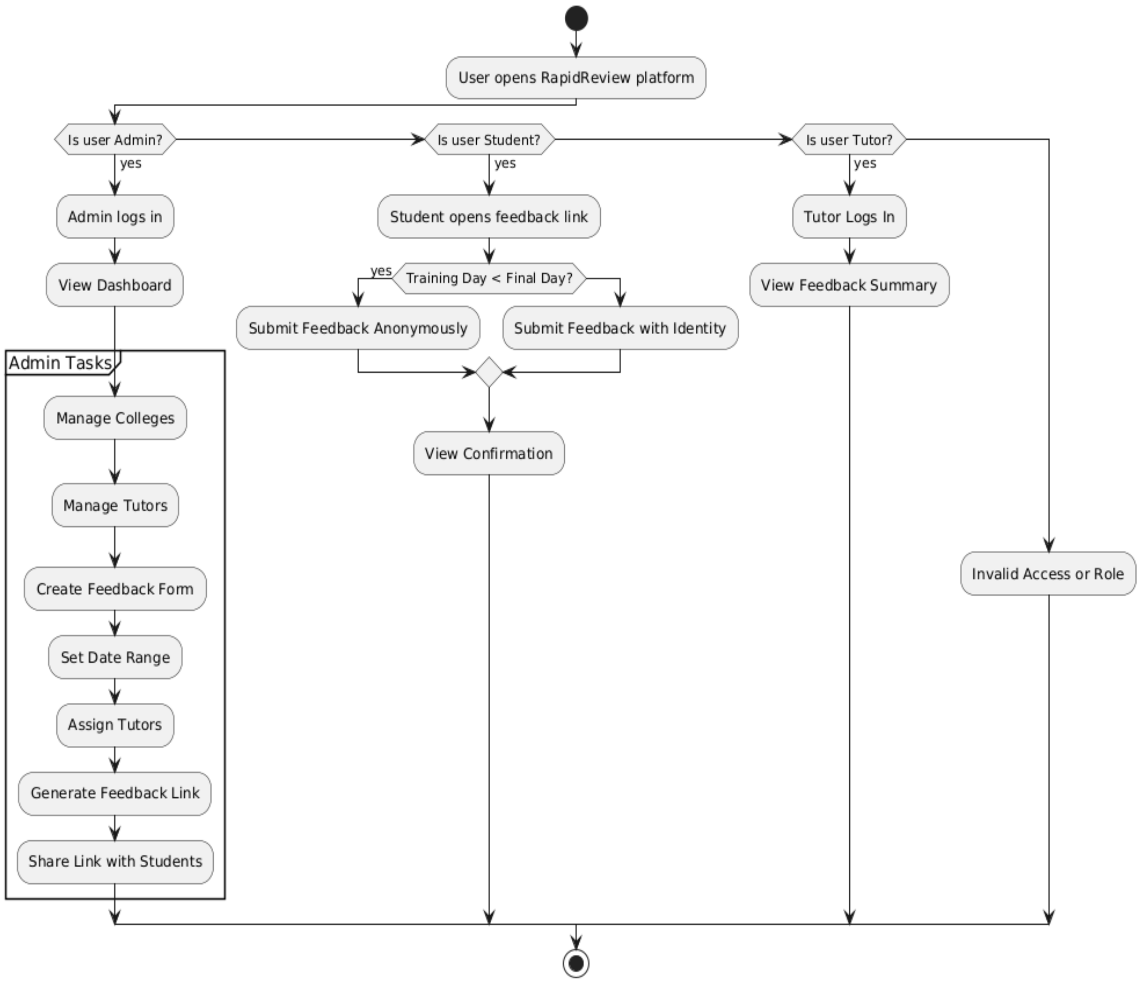


Fig:3.2 Flow Diagram

CHAPTER 4
PROJECT DESCRIPTION

4.1 MODULE DESCRIPTION

4.1.1 USER

The User module enables administrators and students to securely log in and access their respective dashboards. It features role-based access control, ensuring that each user interacts only with relevant functionalities. Administrators manage feedback sessions, while students submit feedback and view submission status within a streamlined interface.

4.1.2 FEEDBACK CREATION AND SUBMISSION MODULE

This module allows administrators to create feedback forms by selecting colleges, tutors, departments, and training dates. A unique feedback link is generated and shared with students, who can submit responses across multiple days. Anonymity is maintained until the final day, after which identities are revealed for accountability.

4.1.3 ADMIN DASHBOARD MODULE

The Admin Dashboard provides comprehensive tools for managing colleges, tutors, and feedback sessions. Administrators can filter feedback data by date, session, or tutor, and monitor trends in real-time. The dashboard also highlights pending feedback and provides insights for decision-making and quality improvement.

4.1.4 FEEDBACK ANALYTICS MODULE

This module aggregates submitted feedback and presents it in structured reports and visual charts. Administrators can evaluate tutor performance, identify areas needing improvement, and export data for review. Day-wise and department-wise analysis enhances transparency and supports informed action.

4.1.5 PROFILE MODULE

The Profile module provides user-specific details and a history of activity within the system. Students can view feedback they've submitted and their status, while admins can monitor session creation and tutor assignments. This organized structure promotes clarity and improves user engagement.

CHAPTER 5

IMPLEMENTATION

Home.jsx

```
import { ChakraProvider, Box } from '@chakra-ui/react';
import { BrowserRouter as Router, Routes, Route, useLocation, Navigate } from 'react-router-dom';
import Sidebar from './components/Sidebar';
import Colleges from './pages/Colleges';
import Tutors from './pages/Tutors';
import Feedback from './pages/Feedback';
import FeedbackDetails from './pages/FeedbackDetails';
import FeedbackForm from './pages/feedbackform';
import Login from './pages/Login';

function App() {
  return (
    <ChakraProvider>
      <Router>
        <MainLayout />
      </Router>
    </ChakraProvider>
  );
}

function MainLayout() {
  const location = useLocation();
  const isFeedbackFormPage = location.pathname.startsWith('/feedbackform/');
  const isHomePage = location.pathname === '/';
  const token = localStorage.getItem('token');
  if (!token && !isFeedbackFormPage && location.pathname !== '/') {
    return <Navigate to="/" replace />;
  }

  return (
    <Box display="flex">
      {!isFeedbackFormPage && !isHomePage && <Sidebar />}
      <Box flex="1" p={5}>
        <Routes>
          <Route path="/college" element={<Colleges />} />
          <Route path="/tutors" element={<Tutors />} />
          <Route path="/feedback" element={<Feedback />} />
          <Route path="/feedback/:id" element={<FeedbackDetails />} />
          <Route path="/feedbackform/:id" element={<FeedbackForm />} />
          <Route path="/" element={<Login />} />
        </Routes>
      </Box>
    </Box>
  );
}

export default App;
```

Feedback.jsx

```
import {
  Box,Heading,Button, Table,  Thead,Tbody,Tr,  Th, Td, useDisclosure, Modal, ModalOverlay
ModalContent,  ModalHeader,  ModalBody, ModalFooter,FormControl,FormLabel, Input,Select,
Badge, Link as ChakraLink, Tag, TagLabel, TagCloseButton, HStack, VStack, Text,  useToast,
Spinner
} from '@chakra-ui/react';

import { Link } from 'react-router-dom';
import { useState, useEffect } from 'react';
import axios from 'axios';
import { useNavigate } from 'react-router-dom';

function Feedback() {
  const { isOpen, onOpen, onClose } = useDisclosure();
  const [feedbacks, setFeedbacks] = useState([]);
  const [formData, setFormData] = useState({
    title: '',
    college: '',
    departments: '',
    selectedTutor: '',
    selectedTutors: [],
    fromDate: '',
    toDate: '',
  });
  const navigate = useNavigate();
  const [colleges, setColleges] = useState([]);
  const [tutors, setTutors] = useState([]);
  const [loading, setLoading] = useState(true);
  const [editMode, setEditMode] = useState(false);
  const [currentFeedbackId, setCurrentFeedbackId] = useState(null);
  const toast = useToast();

  const fetchFeedbacks = async () => {
    setLoading(true);
    try {
      const response = await axios.get(`${import.meta.env.VITE_API_URL}/api/feedback/feedbacks`);
      setFeedbacks(response.data.reverse());
    } catch (err) {
      console.error('Error fetching feedbacks:', err);
      toast({
        title: 'Error fetching feedbacks',
        status: 'error',
        duration: 3000,
      });
    } finally {
      setLoading(false);
    }
  };



  useEffect(() => {
    fetchFeedbacks();
  }, []);
  const handleAddTutor = () => {
    if (formData.selectedTutor && !formData.selectedTutors.some(t => t._id === formData.selectedTutor)) {
      const selectedTutor = tutors.find(t => t._id === formData.selectedTutor);
      if (selectedTutor) {
        setFormData({
          ...formData,
          selectedTutors: [...formData.selectedTutors, selectedTutor],
          selectedTutor: '',
        });
      }
    }
  };

  const handleRemoveTutor = (tutorToRemove) => {
    setFormData({
      ...formData,
      selectedTutors: formData.selectedTutors.filter((tutor) => tutor._id !== tutorToRemove._id),
    });
  };

  const handleEdit = (feedback, e) => {
    e.stopPropagation();
    setCurrentFeedbackId(feedback._id);
    setFormData({
      title: feedback.sessionname,
      college: feedback.college_id._id,
      departments: Array.isArray(feedback.departments) ? feedback.departments.join(', ') :
feedback.departments,
      selectedTutors: feedback.tutors,
      fromDate: new Date(feedback.startdate).toISOString().split('T')[0],
      toDate: new Date(feedback.enddate).toISOString().split('T')[0],
      selectedTutor: '',
    });
    setEditMode(true);
    onOpen();
  };

  const handleDelete = async (id, e) => {
    e.stopPropagation();
    try {
      await axios.post(`${import.meta.env.VITE_API_URL}/api/feedback/deletefeedback/${id}`);
      toast({
        title: 'Feedback deleted successfully',
        status: 'success',
        duration: 3000,
      });
      fetchFeedbacks();
    } catch (err) {
      console.error('Error deleting feedback:', err);
      toast({
        title: 'Error deleting feedback',
        status: 'error',
        duration: 3000,
      });
    }
  };
  const handleSubmit = async () => {
    if (!formData.title || !formData.college || !formData.fromDate || !formData.toDate ||
formData.selectedTutors.length === 0) {
      toast({
        title: 'Please fill all required fields',
        status: 'error',
        duration: 3000,
      });
      return;
    }
      fetchFeedbacks();
      onClose();
      setFormData({
      title: '',
        college: '',
        departments: '',
        selectedTutor: '',
        selectedTutors: [],
        fromDate: '',
        toDate: '',
      });
      setEditMode(false);
      setCurrentFeedbackId(null);
    } catch (err) {
      console.error('Error:', err);
      toast({
        title: `Error ${editMode ? 'updating' : 'creating'} feedback`,
        status: 'error',
        duration: 3000,
      });
    }
  };

  const handleRowClick = (feedback) => {
    navigate(`/feedback/${feedback._id}`);
  };
  if (loading) {
    return (
      <Box display="flex" justifyContent="center" alignItems="center" h="100vh">
        <Spinner size="xl" thickness="4px" speed="0.65s" color="blue.500" />
      </Box>
    );
  }

  return (
    <Box p={6}>
      <Box display="flex" justifyContent="space-between" alignItems="center" mb={6}>
        <VStack align="start" spacing={1}>
          <Heading size="lg">Feedback Forms</Heading>
          <Text color="gray.600">Manage and track feedback across departments</Text>
        </VStack>
        <Button colorScheme="blue" size="lg" onClick={() => {
          setEditMode(false);
          setCurrentFeedbackId(null);
          setFormData({
            title: '',
            college: '',
            departments: '',
            selectedTutor: '',
            selectedTutors: [],
            fromDate: '',
            toDate: '',
          });
          onOpen();
        }}>
          Create Feedback
        </Button>
      </Box>

      <Table variant="simple" bg="white" shadow="sm" rounded="lg">
        <Thead bg="gray.50">
          <Tr>
            <Th>Title</Th>
            <Th>College</Th>
            <Th>Departments</Th>
            <Th>Tutors</Th>
            <Th>Duration</Th>
            <Th>Status</Th>
            <Th>Actions</Th>
          </Tr>
        </Thead>
        <Tbody>
          {feedbacks.map((feedback) => (
            <Tr
              key={feedback._id}
              onClick={() => handleRowClick(feedback)}
              style={{ cursor: 'pointer' }}
              _hover={{ bg: 'gray.50' }}
            >
              <Td>{feedback.sessionname}</Td>
              <Td>{feedback.college_id.collegename}</Td>
              <Td>
                {Array.isArray(feedback.departments)
                  ? feedback.departments.join(', ')
                  : feedback.departments}
              </Td>
              <Td>
                <HStack spacing={2} wrap="wrap">
                  {feedback.tutors.map((tutor, index) => (
                    <Tag key={index} size="md" colorScheme="blue" borderRadius="full">
                      <TagLabel>{tutor.name}</TagLabel>
                    </Tag>
                  ))}
                </HStack>
              </Td>
              <Td>
                <VStack align="start" spacing={0}>
                  <Text fontSize="sm">
                    {new Date(feedback.startdate).toLocaleDateString()} to {new
Date(feedback.enddate).toLocaleDateString()}
                  </Text>
                  <Text fontSize="sm" color="gray.600">
                    ({feedback.days} days)
                  </Text
                  <Input
                    type="date"
                    value={formData.toDate}
                    onChange={(e) => setFormData({ ...formData, toDate: e.target.value })}
                  />
                </HStack>
              </FormControl>

              {formData.fromDate && formData.toDate && (
                <Box w="100%">
                  <Text color="gray.600">
                    Duration: {calculateDays(formData.fromDate, formData.toDate)} days
                  </Text>
                </Box>
              )}
            </VStack>
          </ModalBody>
            <Button onClick={onClose}>Cancel</Button>
          </ModalFooter>
        </ModalContent>
      </Modal>
</Box>
);
}

export default Feedback;
```

index.js

```
const express = require("express");
const mongoose = require("mongoose");
require("dotenv").config();
const db = require("./config/db");
const collegerouter = require("./routers/collegerouter");
const feedbackrouter = require("./routers/feedbackrouter");
const tutorrouter = require("./routers/tutor");
const authrouter=require("./routers/authenticationrouter")
const feedbackmodel = require("./models/feedbackschema");
const cron = require("node-cron");
const cors = require("cors");

const app = express();
app.use(cors());
app.use(express.json());

db();
cron.schedule("0 0 * * *", async () => {
  console.log("Running cron job to update feedback statuses...");
  try {
    const currentDate = new Date();
    const result = await feedbackmodel.updateMany(
      {
        status: "Active",
        enddate: { $lt: currentDate },
      },
      {
        $set: { status: "completed" },
      }
    );

    console.log(`Updated ${result.modifiedCount} feedback records to 'completed'.`);
  } catch (error) {
    console.error("Error running cron job:", error);
  }
});

app.use("/api/college", collegerouter);
app.use("/api/feedback", feedbackrouter);
app.use("/api/tutor", tutorrouter)
app.use("/api/auth",authrouter);

app.listen(8000,'0.0.0.0', () => {
  console.log("Server is running on port 8000");
});
```

Auth.js

```
const express = require('express');
const router = express.Router();
const Student = require('../models/Student');


router.post('/login', async (req, res) => {
  const { username, roomNo, password } = req.body;

  try {

    const student = await Student.findOne({ username, roomNo });
    if (!student) {
      return res.status(400).json({ error: "Student not found" });
    }
    if (student.password !== password) {
      return res.status(401).json({ error: "Incorrect password" });
    }
    res.json({
      _id: student._id,
 name: student.username,
      roomNo: student.roomNo
    });
  } catch (error) {
    console.error("Login error:", error);
    res.status(500).json({ error: "Server error during login" });
  }
});

module.exports = router;
```

Feedback.js

```
const feedbackmodel = require("../models/feedbackschema");
const collegemodel = require("../models/collegeschema");

const addfeedback = async (req, res) => {
  try {
    const { title, college_id, departments, days, tutors, sessionname, startdate, enddate } = req.body;
    console.log(req.body);

    if (!days) {
      return res.status(400).json({ error: 'Days field is missing' });
    }

    const feedback = new feedbackmodel({
      sessionname,
      college_id,
      departments,
      days,
      tutors,
      status: 'Active',
      startdate,
      enddate,
    });

    await feedback.save();

    const feedbackLink = `https://feedback-management-iota.vercel.app/feedbackform/${feedback._id}`;

    feedback.link = feedbackLink;

    await feedback.save();

    res.status(201).json({
      message: 'Feedback added successfully',
      feedbackId: feedback._id,
      link: feedbackLink,
    });
  } catch (err) {
    console.error('Error occurred:', err);
    res.status(500).json({ error: 'An error occurred while adding feedback' });
  }
};


const getFeedbacks = async (req, res) => {
  try {
    const feedbacks = await feedbackmodel
      .find()
      .populate('college_id',  'collegename')
      .populate('tutors', 'name');
    res.status(200).json(feedbacks);
  } catch (err) {
    console.error("Error occurred:", err);
    res.status(500).json({ message: "Server error", error: err });
  }
};
const getfeedbackbyid=async(req,res)=>{
  try {
    const { feedbackId } = req.params;
    console.log(feedbackId)
    const feedback = await feedbackmodel.findById(feedbackId)
      .populate('college_id')
      .populate('staffs')
      .populate('tutors');
    if (!feedback) return res.status(404).json({ message: 'Feedback not found' });

    const currentDate = new Date();
    feedback.feedbackcontent.forEach((content) => {
      if (content.feedbacktype === 'anonymous' && currentDate < feedback.enddate) {
        delete content.records.name;
        delete content.records.email;
      }
    });

    res.status(200).json(feedback);
  } catch (error) {
    res.status(500).json({ message: 'Error fetching feedback', error });
  }
}


const updatefeedback = async (req, res) => {
  try {
    const { feedbackId } = req.params;
    const updates = req.body;
    console.log(req.body)

    const feedback = await feedbackmodel.findById(feedbackId);
    if (!feedback) {
      return res.status(404).json({ message: "Feedback not found" });
    }

    Object.keys(updates).forEach((key) => {
      feedback[key] = updates[key];
    });

    await feedback.save();
    res.status(200).json({ message: "Feedback updated successfully", feedback });
  } catch (err) {
    console.error("Error occurred:", err);
    res.status(500).json({ message: "Server error", error: err });
  }
};
const deletefeedback = async (req, res) => {
  try {
    const { feedbackId } = req.params;

    const updatedFeedback = await feedbackmodel.findByIdAndUpdate(
      feedbackId,
      { status: "cancelled" },
      { new: true }
    );

    if (!updatedFeedback) {
      return res.status(404).json({ message: "Feedback not found" });
    }

    res.status(200).json({ message: "Feedback marked as inactive successfully", feedback:
updatedFeedback });
  } catch (err) {
    console.error("Error occurred:", err);
    res.status(500).json({ message: "Server error", error: err });
  }
};
const submitFeedback = async (req, res) => {
  const { feedbackId } = req.params;
  const { specificTopic, improvement, rating, feedbackType, name, email, department, tutor } = req.body;

  try {
    const feedbackSession = await feedbackmodel.findById(feedbackId);
    if (!feedbackSession) {
      return res.status(404).json({ error: 'Feedback session not found' });
    }

    const today = new Date().toISOString().split('T')[0];
    const sessionEndDate = new Date(feedbackSession.enddate).toISOString().split('T')[0];
    const isLastDay = today === sessionEndDate;
    const finalFeedbackType = isLastDay ? 'public' : 'private';

    const todayFeedbackIndex = feedbackSession.feedbackcontent.findIndex(
      (feedback) => new Date(feedback.date).toISOString().split('T')[0] === today
    );
console.log("sibsbob")
    const newFeedbackRecord = {
      rating,
      description: improvement,
      specificTopic,
      department,
      tutor,
      ...(finalFeedbackType === 'public' && { name, email })
    };
    console.log("sidsssssssssssssssbsbob")


    if (todayFeedbackIndex !== -1) {
      const existingFeedback = feedbackSession.feedbackcontent[todayFeedbackIndex];
      existingFeedback.records.push(newFeedbackRecord);
      existingFeedback.totalResponses += 1;
    } else {
      feedbackSession.feedbackcontent.push({
      date: new Date(),
        feedbacktype: finalFeedbackType,
        totalResponses: 1,
        records: [newFeedbackRecord],
      });
    }

    await feedbackSession.save();

    res.status(200).json({ message: 'Feedback submitted successfully' });
  } catch (error) {
    console.error('Error submitting feedback:', error);
    res.status(500).json({ error: 'Server error' });
  }
};


module.exports = { addfeedback, updatefeedback, deletefeedback, getFeedbacks, getfeedbackbyid,
submitFeedback }
```

CHAPTER 6
RESULTS AND DISCUSSIONS

LOGIN: Login page the authorize the admin to use application

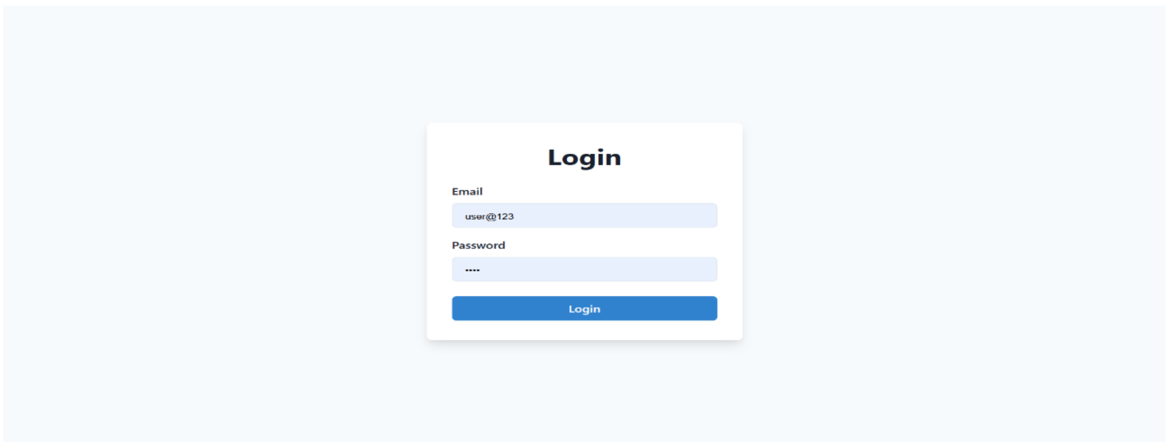
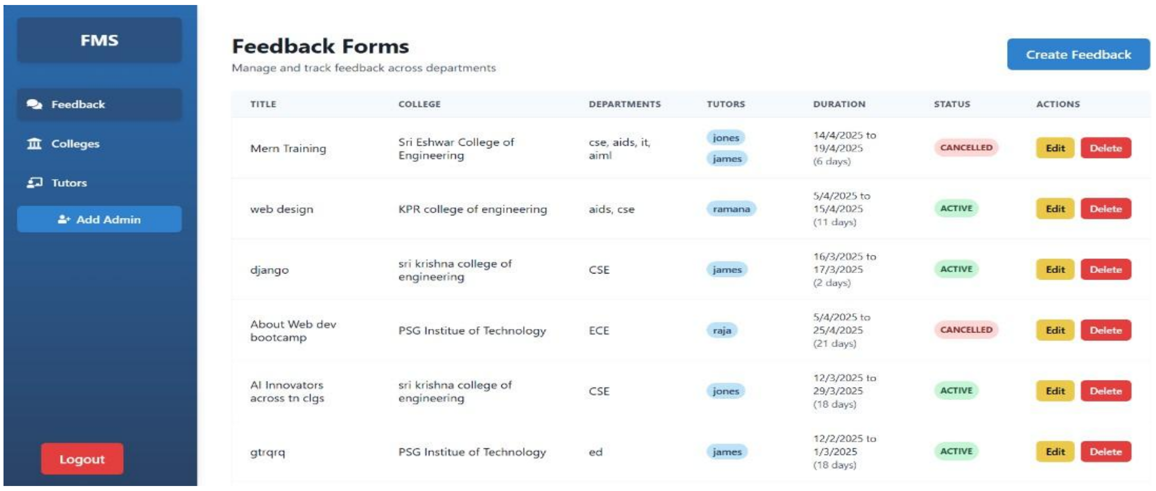


Fig:6.1 Login Module

FEEDBACK MODULE: Creating Feedbacks and can navigate to other features.



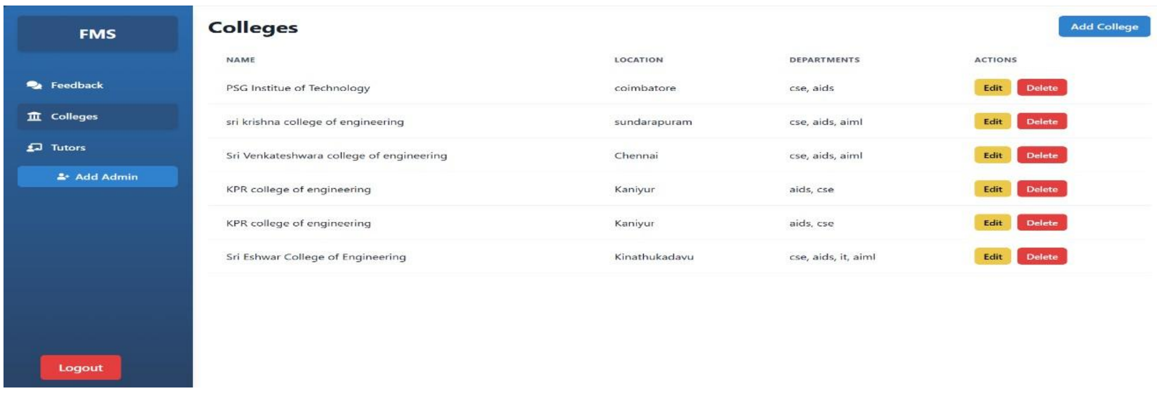
Feedback Forms

Manage and track feedback across departments

TITLE	COLLEGE	DEPARTMENTS	TUTORS	DURATION	STATUS	ACTIONS
Mern Training	Sri Eshwar College of Engineering	cse, aids, it, aiml	jones james	14/4/2025 to 19/4/2025 (6 days)	CANCELLED	Edit Delete
web design	KPR college of engineering	aids, cse	ramana	5/4/2025 to 15/4/2025 (11 days)	ACTIVE	Edit Delete
django	sri krishna college of engineering	CSE	james	16/3/2025 to 17/3/2025 (2 days)	ACTIVE	Edit Delete
About Web dev bootcamp	PSG Institue of Technology	ECE	raja	5/4/2025 to 25/4/2025 (21 days)	CANCELLED	Edit Delete
AI Innovators across tn clgs	sri krishna college of engineering	CSE	jones	12/3/2025 to 29/3/2025 (18 days)	ACTIVE	Edit Delete
gtrqrq	PSG Institue of Technology	ed	james	12/2/2025 to 1/3/2025 (18 days)	ACTIVE	Edit Delete

Fig:6.2 Feedback Module

COLLEGE MODULE: It shows all the registered colleges.

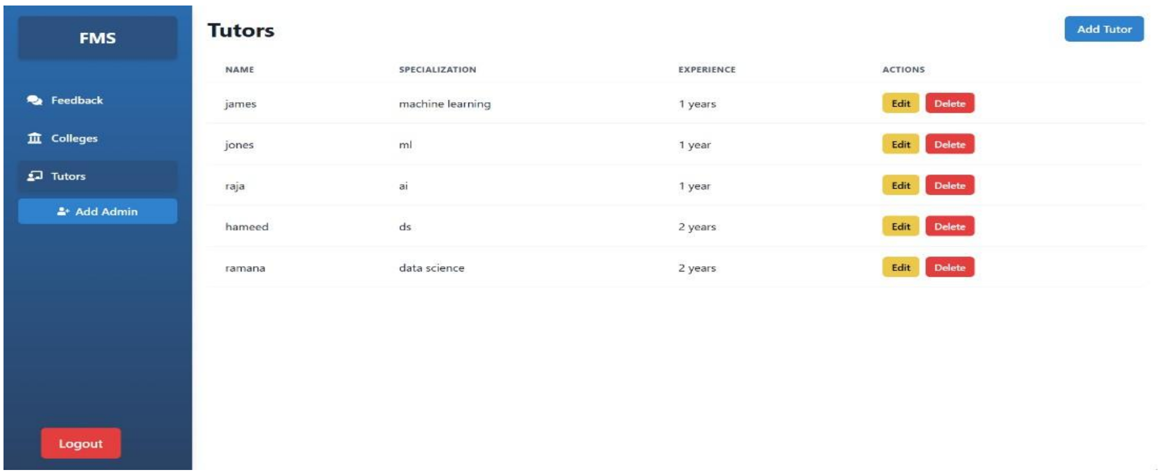


Colleges

NAME	LOCATION	DEPARTMENTS	ACTIONS
PSG Institue of Technology	coimbatore	cse, aids	Edit Delete
sri krishna college of engineering	sundarapuram	cse, aids, aiml	Edit Delete
Sri Venkateshwara college of engineering	Chennai	cse, aids, aiml	Edit Delete
KPR college of engineering	Kaniyur	aids, cse	Edit Delete
KPR college of engineering	Kaniyur	aids, cse	Edit Delete
Sri Eshwar College of Engineering	Kinathukadavu	cse, aids, it, aiml	Edit Delete

Fig:6.3 College Module

TUTORS PAGE: Allows us to display the tutors available.

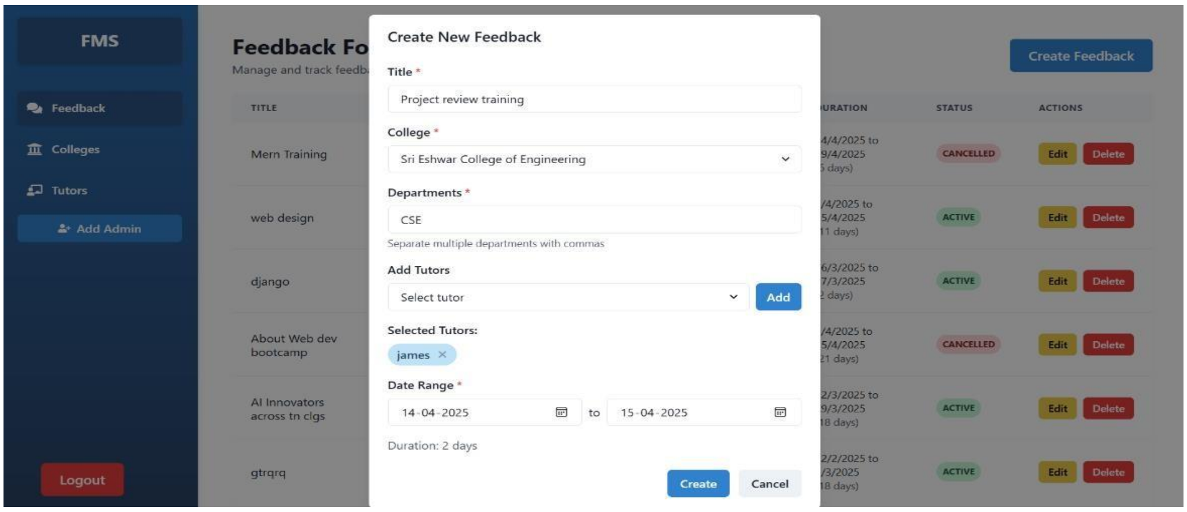


Tutors

NAME	SPECIALIZATION	EXPERIENCE	ACTIONS
james	machine learning	1 years	Edit Delete
jones	ml	1 year	Edit Delete
raja	ai	1 year	Edit Delete
hameed	ds	2 years	Edit Delete
ramana	data science	2 years	Edit Delete

Fig:6.4 Tutors Page

CREATE FEEDBACK: This module deals with creating a new feedback.



Feedback Fo

Manage and track feedb

TITLE	URATION	STATUS	ACTIONS
Mern Training	4/4/2025 to 9/4/2025 5 days)	CANCELLED	Edit Delete
web design	/4/2025 to 5/4/2025 11 days)	ACTIVE	Edit Delete
django	6/3/2025 to 7/3/2025 2 days)	ACTIVE	Edit Delete
About Web dev bootcamp	/4/2025 to 5/4/2025 21 days)	CANCELLED	Edit Delete
AI Innovators across tn clgs	2/3/2025 to 9/3/2025 18 days)	ACTIVE	Edit Delete
gtrqrq	2/2/2025 to /3/2025 18 days)	ACTIVE	Edit Delete

Create New Feedback

Title *
Project review training

College *
Sri Eshwar College of Engineering

Departments *
CSE
Separate multiple departments with commas

Add Tutors
Select tutor | Add

Selected Tutors:
james ×

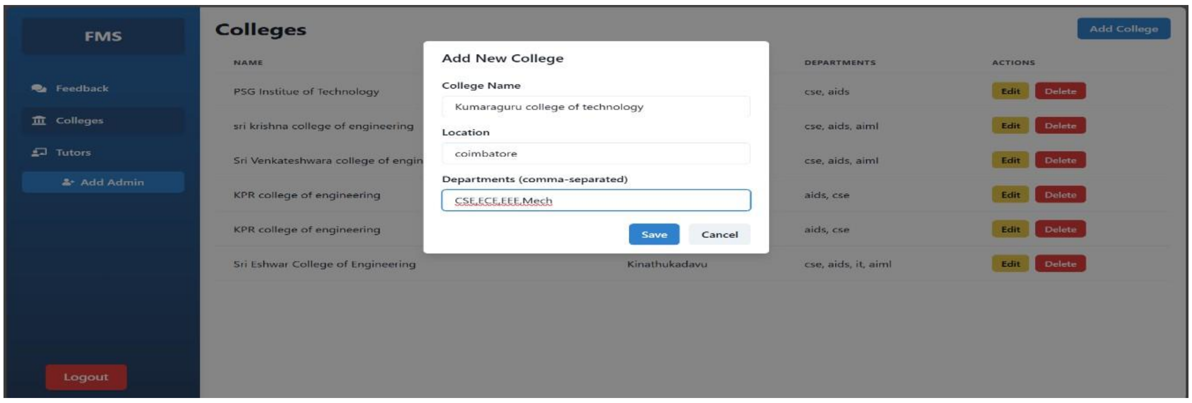
Date Range *
14-04-2025 to 15-04-2025

Duration: 2 days

Create | Cancel

Fig:6.5 Create Feedback Module

ADD NEW COLLEGE: This module has the features of adding, editing ,deleting of the colleges.



Colleges

NAME		DEPARTMENTS	ACTIONS
PSG Institue of Technology		cse, aids	Edit Delete
sri krishna college of engineering		cse, aids, aiml	Edit Delete
Sri Venkateshwara college of engin		cse, aids, aiml	Edit Delete
KPR college of engineering		aids, cse	Edit Delete
KPR college of engineering		aids, cse	Edit Delete
Sri Eshwar College of Engineering	Kinathukadavu	cse, aids, it, aiml	Edit Delete

Add New College

College Name
Kumaraguru college of technology

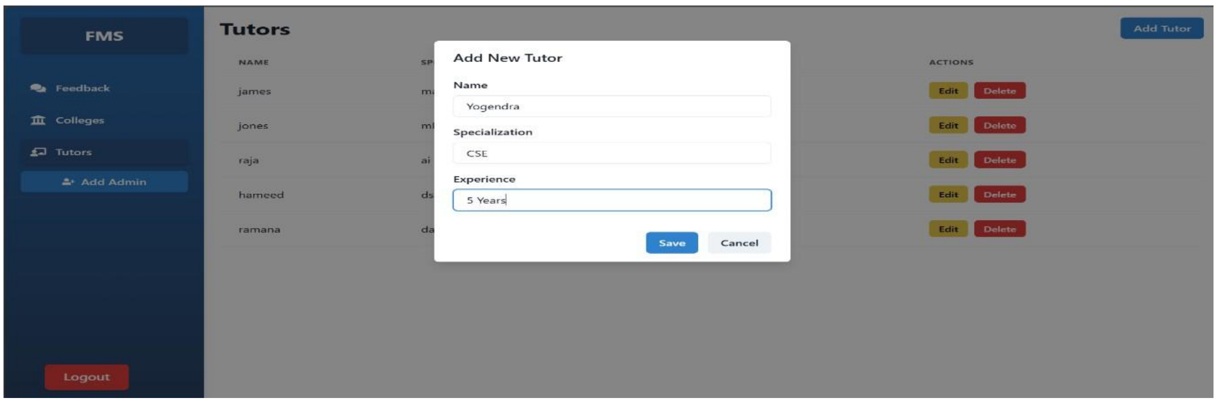
Location
coimbatore

Departments (comma-separated)
CSE,ECE,EEE,Mech

Save | Cancel

Fig:6.6 Add New College Page

ADD NEW TUTOR: This module has the features of adding, editing ,deleting of the tutors.



Tutors

Add New Tutor

Name: Yogendra

Specialization: CSE

Experience: 5 Years

NAME	SP	ACTIONS
james	m	Edit Delete
jones	ml	Edit Delete
raja	ai	Edit Delete
hameed	ds	Edit Delete
ramana	da	Edit Delete

Fig:6.7 Add New Tutor Page

CHAPTER 7
CONCLUSION AND FUTURE ENHANCEMENT

RapidReview is a comprehensive web-based platform designed to streamline and enhance the process of collecting and managing student feedback for training sessions in educational institutions. It enables students to easily submit feedback on sessions conducted by tutors, while administrators can create feedback forms, manage colleges and tutors, and analyze feedback data through an intuitive interface. The system allows secure login, real-time feedback submission, and role-based access for administrators and students, ensuring efficiency, transparency, and accountability throughout the feedback lifecycle. All data, including user profiles, session records, and feedback entries, is securely stored in a structured database for easy access and analysis.

Looking ahead, **RapidReview** can be enhanced with features such as automated notifications for upcoming or pending feedback sessions, AI-based sentiment analysis for evaluating student responses, and integration with mobile applications to increase accessibility and user engagement. Additionally, advanced analytics and customizable reports can offer administrators deeper insights into training effectiveness and student satisfaction. These enhancements will further elevate the platform's scalability, responsiveness, and utility, establishing **RapidReview** as an essential tool for modern, efficient, and data-driven academic feedback management.

CHAPTER 8
REFERENCES

[1] S. Kumar, A. Sharma, and P. Verma, "A Web-Based Feedback Management System for Educational Institutions," *International Journal of Computer Applications*, vol. 182, no. 47, pp. 25–30, 2021. doi:10.5120/ijca2021921613.

[2] A. A. Shaikh and B. D. Pindoriya, "Student Feedback System using Web Technology," *2022 IEEE International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp. 303–308, 2022. doi:10.1109/ICCCIS56565.2022.9934926.

[3] P. Patel, M. Mehta, and R. Bhavsar, "Design and Implementation of Feedback System using MERN Stack," *International Journal of Engineering Research & Technology (IJERT)*, vol. 10, no. 4, pp. 112–117, Apr. 2021.

[4] R. Joshi and S. Saxena, "Enhancing Educational Feedback Systems Using Data Analytics," *2023 4th International Conference on Education and Technology (ICET)*, pp. 89–95, 2023. doi:10.1109/ICET57405.2023.10140790.

[5] V. Yadav and N. Agarwal, "Survey of Online Student Feedback Systems in Higher Education," *Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 7, no. 6, pp. 987–992, June 2020.

[6] https://developer.mozilla.org/en-US/docs/Learn/Serverside/Express_Nodejs

[7] https://www.mongodb.com/docs/manual/introduction/

[8] https://reactjs.org/docs/getting-started.html

[9] https://expressjs.com/

[10] https://nodejs.org/en/docs

Annexure

Sri Eshwar College of Engineering
Coimbatore | Tamilnadu
An Autonomous Institution
Affiliated to Anna University, Chennai

PROJECT TITLE	**RapidReview: A Digital Solution for Feedback Management**	
PROGRAM	**B.E. COMPUTER SCIENCE AND ENGINEERING**	
PROJECT BATCH NUMBER	**48**	
BATCH MEMBERS	722823104132	**SABARI M**
	722823104143	**SANTHOSH KUMAR S**
	722823104152	**SHANTHISH S R**
	722823104195	**YOGENDRA N**
NAME OF THE SUPERVISOR	**Dr. M. ABINAYA, M.E.,(Ph.D)**	
NAME OF THE SDG GOALS MAPPED	**Industry, Innovation, and Infrastructure**	
MENTION THE SDG GOALS NUMBER	**SDG 9**	
NAME OF THE TRL LEVEL	**Technology Formulation**	
MENTION THE TRL LEVEL	**TRL 2**	

POs & PSOs Mapping (Put a tick mark in the mapped PO's & PSO's):

Program Outcomes											**Program Specific Outcomes**	
PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**
3	**3**	**3**	**1**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**

Signature of the Supervisor

(Name of the Supervisor)

Annexure

Sri Eshwar College of Engineering
Coimbatore | Tamilnadu
An Autonomous Institution
Affiliated to Anna University, Chennai

VENUE AND EXPENDITURE STATEMENT FOR THE PROJECT WORK

Laboratory details where the project is carried out	**Project Lab, CSE**
Software / Hardware details	**NA**

Details of the Component and Expenditure

S. No	Name of the Component	Qty	Price / Unit in (Rs.)	Amount (Rs.)
	NA	NA	NA	NA

(*Include any other charges which includes fabrication cost and others)

Signature of the student — **Signature of the Supervisor**

(Names of the Student) — **(Name of the Supervisor)**