# PERFORMANCE BASED FACULTY APPRAISAL SYSTEM

**A Project report**

Submitted in partial fulfilment of the

Requirements for the award of the Degree of

**BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)**

**By**

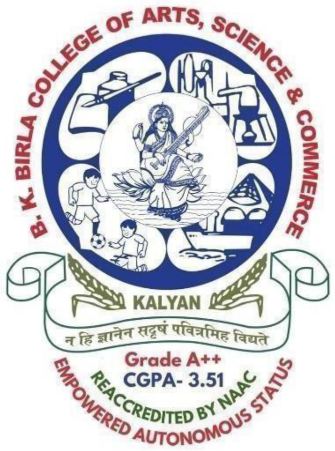
Mr. Ankit Gupta - 3837108

Mr. Saad Kureshi - 3837081

**Under the esteemed guidance of**

**Mrs. Esmita Gupta**

(Head, Dept of I.T)



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**B. K. BIRLA COLLEGE, KALYAN**

**(EMPOWERED AUTONOMOUS STATUS)**

***(Affiliated to University of Mumbai)***

**KALYAN, 421304 MAHARASHTRA 2024-2025**

# B. K. BIRLA COLLEGE, KALYAN

(EMPOWERED AUTONOMOUS STATUS)

*(Affiliated to University of Mumbai)*

**KALYAN, 421304 MAHARASHTRA**

## DEPARTMENT OF INFORMATION TECHNOLOGY

## CERTIFICATE

This is to certify that the project entitled**, "PERFORMANCE BASED FACULTY APPRAISAL SYSTEM"**, is bonafied work of **Mr. Ankit Gupta and Mr. Saad Kureshi** bearing Seat. No. **3837108, 3837081** submitted in partial fulfilment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from the University of Mumbai.

**Internal Guide** **Coordinator**

**External Examiner**

**Date: 17/04/2025** **College seal**

## PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

**PRN No :** 2022016401830893, 2022016401872371 **Roll no :** 3837108, 3837081

1. **Name of the Student(s)**

   Ankit Gupta, Saad Kureshi

2. **Title of the Project**

   Performance Based Faculty Appraisal System

3. **Name of the Guide**

   Mrs. Esmita Gupta

4. **Is this your First submission?** Yes ☐ No ☐

**Signature of the Student(s)** **Signature of the Guide**

**Signature of the Coordinator**

**Date: 17/04/2025**

## Abstract

The "Performance-Based Faculty Appraisal System" introduces a modernized method for assessing and enhancing employee performance within organizations. Unlike traditional evaluation methods, this system utilizes real-time data and clear, objective metrics. By employing advanced analytics, it creates a thorough evaluation framework that aligns individual objectives with those of the organization. This system ensures that employees receive prompt feedback, fostering a culture of continuous improvement and personal growth.

A key feature of this system is its reliance on technology to monitor and analyze performance data. It integrates various performance indicators, such as productivity, work quality, and teamwork, into a unified assessment model. This data-driven approach reduces biases and subjectivity, providing a transparent and equitable evaluation process. Employees gain a better understanding of their strengths and areas needing improvement, which helps in crafting targeted training and development programs. Additionally, the system promotes accountability and motivation, as employees can clearly see how their efforts impact their performance reviews.

The advantages of a Performance-Based Appraisal System extend to the entire organization. By aligning individual performance with strategic goals, it enhances efficiency, innovation, and competitiveness. This system also plays a crucial role in talent management by identifying top performers and future leaders, ensuring appropriate recognition and rewards. Furthermore, it aids in making informed decisions regarding promotions, compensations, and career development plans. Ultimately, a Performance-Based Appraisal System serves as a strategic tool that drives organizational growth and boosts employee satisfaction.

# ACKNOWLEDGEMENT

In the journey towards success, dedication, hard work, patience, and expert guidance serves as the fundamental pillars of our efforts. Our college, B.K. Birla College (Autonomous), has provided a nurturing environment for learning and growth.

A special tribute is reserved for **Mrs. Esmita Gupta Mam**, whose meticulous guidance and attention to detail have been invaluable. We extend our gratitude to everyone who has supported us, collectively transforming aspirations into tangible achievements, and illuminating the path forward.

# DECLARATION

I hereby declare that the project entitled, **“PERFORMANCE BASED FACULTY APPRAISAL SYSTEM”** done at **Kalyan**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

**Ankit Gupta** **Saad Kureshi**

# TABLE OF CONTENTS

# TABLE OF FIGURES

# Chapter 1

# Introduction

In today's educational landscape, ensuring high-quality teaching is essential for student success and institutional excellence. A performance-based appraisal system for student teaching faculty addresses this need by shifting the focus from traditional, often subjective evaluation methods to a more objective and comprehensive approach. This system evaluates educators based on their actual teaching performance and the impact they have on student learning outcomes. By incorporating diverse feedback sources, including student evaluations, peer reviews, and self-assessments, the system aims to provide a holistic view of teaching effectiveness.

Implementing such an appraisal system has the potential to enhance teaching practices and support faculty development by identifying strengths and areas for improvement. It encourages a culture of continuous professional growth and accountability, which benefits both educators and students. However, this approach also presents challenges, such as ensuring fairness and consistency in evaluations and managing the increased administrative workload. This project will explore these aspects in detail, offering insights into how a performance based appraisal system can be effectively integrated into educational institutions to foster a more dynamic and supportive learning environment.

## 1.1 Background

The traditional methods of evaluating teaching faculty often rely on outdated metrics such as tenure status, seniority, and infrequent classroom observations, which can fail to capture the true effectiveness of educators. As educational institutions strive to improve student outcomes and adapt to evolving pedagogical standards, there is a growing demand for more nuanced and evidence-based appraisal systems. Performance-based appraisal systems have emerged as a solution, offering a more comprehensive approach by integrating direct assessments of teaching performance with feedback from various stakeholders, including students and peers. This model not only emphasizes measurable student learning outcomes but also considers the quality of interactions and engagement within the classroom. Despite the promise of these systems, their implementation can be complex, requiring careful consideration of factors such as fairness, consistency, and the potential impact on faculty

## 1.2 Objectives

The objectives of a performance-based appraisal system for student teaching faculty are to accurately assess teaching effectiveness, provide actionable feedback for professional growth, and enhance student learning outcomes. Here are some common objectives:

1. **Enhance Evaluation Accuracy:** Provide a more precise assessment of teaching effectiveness beyond traditional metrics.

2. **Promote Professional Development:** Offer actionable feedback to guide faculty in their professional growth and skill enhancement.

3. **Improve Student Learning Outcomes:** Focus on how teaching practices impact student achievement and engagement.

4. **Ensure Fairness:** Implement consistent evaluation criteria to ensure equitable assessments across all faculty members.

5. **Incorporate Multiple Feedback Sources:** Use input from students, peers, and self-assessments for a comprehensive view of teaching performance.

6. **Support Continuous Improvement:** Facilitate ongoing refinement of teaching methods based on regular performance reviews.

7. **Encourage Accountability:** Establish clear expectations for teaching quality and hold faculty accountable for meeting them.

8. **Foster a Positive Teaching Environment:** Create a supportive atmosphere that values and nurtures faculty development and excellence.

## 1.3 Purpose, Scope, and Applicability

### 1.3.1 Purpose

The purpose of the performance-based appraisal system for student teaching faculty is to enhance the effectiveness and quality of teaching by implementing a more objective and comprehensive evaluation framework. This system aims to provide accurate assessments of teaching performance through multiple feedback sources, including student evaluations, peer reviews, and self-assessments. By focusing on actual teaching impact and student learning outcomes, the system seeks to support continuous professional development, foster a culture of accountability, and improve overall educational experiences. Ultimately, the goal is to align faculty performance with institutional standards and student needs, ensuring a more dynamic and effective teaching environment.

### 1.3.2 Scope

The scope of the performance-based appraisal system for student teaching faculty encompasses the development and implementation of a structured evaluation framework that integrates multiple feedback sources, including student surveys, peer reviews, and self-assessments. It involves setting clear performance metrics, training evaluators, and ensuring consistent application across different departments. This system will be piloted within selected departments before broader institution-wide adoption. The scope also includes addressing challenges related to fairness, consistency, and faculty buy-in to ensure the system's effectiveness and sustainability.

### 1.3.3 Applicability

The applicability of the performance-based appraisal system for student teaching faculty extends to enhancing the quality of education across various academic institutions. This system can be adapted to different educational settings by tailoring evaluation criteria and feedback mechanisms to align with institutional goals and specific teaching contexts. It is applicable for assessing diverse teaching methodologies and providing targeted professional development opportunities. By integrating this system, institutions student learning outcomes, making it a versatile tool for continuous improvement in higher education.can better support faculty growth, address individual teaching challenges, and ultimately improve

# Chapter 2

# System Analysis

## 2.1 Existing System

The Existing System section details the traditional methods currently used for faculty evaluations, often characterized by manual processes or outdated software. These methods suffer from inefficiencies, lack of transparency, and potential biases, resulting in inconsistent and delayed appraisals. This inadequacy underscores the necessity for a more efficient, accurate, and equitable appraisal system.

The Earlier Appraisal Systems used to elaborate on the traditional methods currently employed for faculty performance evaluations, which are often characterized by manual processes and the use of basic software tools such as Microsoft Excel. In these conventional systems, faculty members are typically required to fill out extensive self-evaluation forms and submit them for review. Administrators then manually compile and analyze these submissions, often leading to significant delays and inefficiencies. This process is not only time-consuming but also prone to human error, resulting in inaccuracies and inconsistencies in the appraisal data. Moreover, these traditional methods lack standardization and transparency, which can lead to subjective evaluations and potential biases. The absence of an integrated platform means that peer reviews and student feedback, if collected, are often managed separately and not systematically incorporated into the overall appraisal. Furthermore, the manual nature of these processes makes it difficult to provide timely and actionable feedback to faculty members, hindering their professional development. The reliance on basic tools like Excel also means that data management is cumbersome, with limited capabilities for data analysis and reporting. These limitations underscore the need for a more efficient, accurate, and holistic approach to faculty performance appraisal, paving the way for the development of an advanced, technology-driven system.

## 2.2 Proposed System

The "Proposed System" section outlines the innovative solution provided by the "Performance-Based Faculty Appraisal System," developed using the MERN stack and deployed on Amazon AWS. This system introduces a streamlined, automated process for faculty evaluations, incorporating features such as self-appraisal, peer evaluation, and feedback from students and parents. By leveraging modern web technologies, the proposed system ensures data accuracy, enhances transparency, and reduces biases. The platform provides secure, role- based access, allowing administrators, faculty, and students to interact with the system seamlessly. Ultimately, this proposed system aims to improve the overall quality of education by offering a reliable and efficient tool for faculty performance assessment.

Building on this foundation, the proposed system includes several keys designed to enhance the evaluation process. The faculty login interface provides options for submitting self-appraisals, viewing peer evaluation results, and accessing comprehensive self-appraisal reports. For administrators, the system offers functionalities to add and manage faculty and student profiles, create and distribute appraisal forms, and generate detailed performance reports. The student interface allows for the submission of faculty feedback forms, which are integral to providing a well-rounded appraisal. Additionally, the system incorporates analytics to track performance trends over time and identify areas for improvement. Automated notifications and reminders ensure that all participants complete their evaluation on time, further streamlining the process. By incorporating and digitizing the modern appraisal system, this solution not only improves efficiency but also fosters a culture of continuos and regular continual improvement and also helps gain a huge accountability within educational institutions

## 2.3 Requirement analysis

### 1. Functional Requirements:

a. User Management

b. Performance Metrics

c. Feedback Process.

d. Appraisal Collection

e. Reporting

f. Notification

g. Integration

## 2. Non functional Requirements:

a. Performance

b. Scalability

c. Security

d. Usability

e. Reliability

f. Maintainability

g. Compatibility

## 3. Technical Requirements:

a. Technology Stack

b. Development tools

c. Testing

## 4. Deployment and Maintainence:

a. Deployment Plan

b. Maintainence Plan

## 2.4 Hardware Requirements

1. **Laptops/Desktops:**
   - **Processor:** Intel i5 or AMD equivalent, or higher
   - **RAM:** Minimum 8GB(16GB recommended)
   - **Storage:** SSD with at least 256GB of available space
2. **Router/Switch:**
   - **Gigabit Ethernet Router or switch**
   - **Wi-Fi Access:** For wireless connectivity
3. **Backup and Storage:**
   - **External Hard Devices:**

- **Capacity:** Minimum 1 TB for regular backups
- **Cloud Storage Solution:** AWS S3, Google Cloud Storage, or Azure Blob Storage for offsite backups

## 2.5 Software Requirements

1. **Operating System:**

- **Development:** Windows 10/11, macOS Mojave or later , or a Linux distribution(Ubuntu 20.04 LTS recommended)

- **Deployment:** AWS Linux AMI or Amazon Linux 2 for the EC2 instance

**2. Development tools:**

- **Code Editor:** Visual Studio Code(VS code) with necessary extensions(e.g. ESLint, Prettier, MongoDB)

- **Version Control:** Git and a Git hosting service like Github, Gitlab or Bigbucket

**3. Web Technologies:**

- **Frontend:** React.js (including necessary libraries and tools like React Router, Axios)

- **Backend:** Node.js (with Express.js framework)

- **Database:** MongoDB (and tools like MongoDB compass for the database management)

**4. APIs and Libraries:**

- **Authentication:** JWT (JSON Web Tokens) or OAuth for user authentication

- **Deployment:** AWS CLI or SDK for deployment and management

**5. Testing:**

- **Unit Testing:** Jest or Mocha/Chai for backend, React Testing Library for frontend

- **Integration Testing:** Cypress or Selenium for end-to-end testing

## 2.6 Justification of Selection of Technology

### 2.6.1 MERN Stack

The MERN stack is a popular set of technologies used for building modern web applications. It consists of four main components:

1. **Mongo DB**: A NoSQL database that stores data in flexible, JSON- like documents. It's known for its scalability and ease of use.
2. **Express.js**: A web application framework for Node.js. It provides a robust set of features to develop web and mobile applications and is used to build the backend of the application.
3. **React**: A JavaScript library for building user interfaces, particularly single-page applications. React allows developers to create large web applications that can update and render efficiently in response to data changes**.**
4. **Node.js**: A JavaScript runtime built on Chrome's V8 JavaScript engine. It allows developers to run JavaScript on the server side, enabling the building of scalable network applications.

### 2.6.2 Render

Render is a modern cloud platform that enables developers to host full-stack web applications, static sites, APIs, background workers, and databases with ease. It supports a wide range of programming languages and frameworks such as Node.js, Python, Ruby, Go, and even custom environments using Docker. Developers can deploy directly from Git repositories like GitHub or GitLab through automatic CI/CD pipelines. Render provides free SSL (HTTPS) for every deployed service, ensuring secure communication by default. It supports both static and dynamic web apps, and even offers managed PostgreSQL databases with daily backups and scaling options. The platform includes features like custom domains, cron jobs, secret management, pull request previews, and real-time logs for monitoring application behavior. Developers can also define infrastructure configurations using a `render.yaml` blueprint file. With support for automatic scaling and a user-friendly dashboard, Render combines powerful cloud hosting features with the simplicity and automation modern developers need.

### 2.6.3 Literature Review

It's crucial to explore key areas of research and practice relevant to performance evaluation and appraisal systems. Begin by examining existingmodels of performance appraisal in educational settings, focusing on their effectiveness and limitations. Highlight studies on performance metrics, feedback mechanisms, and their impact on teaching quality and faculty development. Discuss methodologies used for evaluating faculty performance, including quantitative and qualitative approaches. Review best practices in aligning appraisal systems with educational goals and institutional objectives. Address challenges and gaps in current systems, such as biases, inconsistencies, or lack of actionable feedback. Finally, consider innovations in performance appraisal technology, such as automated systems and data analytics, and how they can enhance the appraisal process for teaching faculty. This review will provide a solid foundation for developing a robust, evidence-based appraisal system tailored to improving teaching effectiveness and faculty performance.

The literature on performance appraisal systems in education reveals both the strengths and limitations of traditional evaluation methods and the potential benefits of performance-based approaches. Traditional systems often rely on student evaluations, peer reviews, and administrative observations. However, research indicates that these methods can be limited by biases and Inconsistencies, student evaluations may reflect satisfaction rather than actual teaching effectiveness (Marsh & Roche, 1997), while peer reviews can vary widely in quality and objectivity (Smith, 2008). Performance-based appraisal systems seek to address these shortcomings by integrating diverse evaluation metrics, such as student outcomes, peer assessments, and self-reflections, which can provide a more comprehensive and accurate measure of teaching performance (Kane, Shaw, & Crampton, 1995). Studies suggest that these systems enhance the reliability of evaluations and support professional development by offering detailed feedback and setting clear performance goals (Locke & Latham, 2002). However, challenges such as ensuring fairness, managing faculty resistance, and allocating resources must be carefully managed to maximize effectiveness (Landy & Farr, 1980).
Overall, adopting performance-based appraisal systems holds promise for improving teaching quality and fostering a more effective educational environment.

The literature review will also explore the integration of technology appraisal systems. It will highlight how web-based platforms and automated tools can address many of the issues found in traditional methods such as improving efficiency, accuracy, and transparency. Studies will be reviewed to illustrate the successful implementation of similar technological solutions in educational settings demonstrating the potential benefits of adopting such systems.

Furthermore, specific attention will be given to the MERN stack (MongoDB, Express.js, React, Node.js) and its application in building saclable and responsive Web applications. This review will cover the technical adavantages of using the MERN stack for developing robust educational tools. Additionally, the deployment on Amazon AWS will be discussed emphasizing its reliability, scalability, and suitability for hosting complex web based systems.

By examining case studies and existing applications, the literature will demonstrate the proven benefits of using the MERN Stack and AWS in educational settings. This analysis will provide a solid foundation for understanding the technological choices and their impact on enhancing the appraisal process.

Finally, the literature review will identify gaps in the current research that the proposed "Performance Based Faculty Appraisal System" aims to address. Thsese gaps may include the need for more integrated and real-time feedback mechanisms, enhanced user interfaces, and more efficient administrative workflows. By bridging these gaps, the proposed system aims to provide a more effective, user friendly, and comprehensive tool for faculty performance evaluation, thereby building to the overall improvement of educational quality and faculty development.

### 2.6.4 Why MERN Stack?

The MERN stack (MongoDB, Express.js, React, Node.js) offers several compelling advantages for building modern websites compared to other technologies:

1. **Unified Language:** With the MERN stack, developers use JavaScript throughout the entire stack (client-side and server-side), which simplifies development and reduces the need to switch between different languages.

2. **Scalability:** Node.js provides a non-blocking, event-driven architecture that supports scalable network applications. This makes it well-suited for handling large volumes of concurrent requests efficiently.

3. **React for Frontend:** React.js offers a powerful and flexible way to build dynamic, single-page applications with a virtual DOM that enhances performance by minimizing direct manipulation of the actual DOM.

4. **Flexibility and Performance:** MongoDB's NoSQL database model allows for flexible data storage and quick access to data, making it suitable for handling varied and evolving data structures.

5. **Full-Stack Solution:** The MERN stack provides a complete solution from front-end to back-end, reducing integration issues and improving overall efficiency. Express.js simplifies server-side development and API creation, while MongoDB and Node.js handle data storage and application logic.

6. **Community and Ecosystem:** All components of the MERN stack have strong community support and extensive libraries and tools, which can accelerate development and problem-solving.

7. **Real-Time Data:** Node.js and MongoDB are well-suited for real-time applications like chat applications or live updates, providing a smooth user experience.

Overall, the MERN stack's unified use of JavaScript, combined with its scalability, performance, and robust ecosystem, makes it a powerful choice for developing modern, dynamic websites and applications.

### 2.6.5 Why Render ?

Render is a good choice for website deployment due to its extensive range of features and benefits:

1. **Free Hosting Tier:** Render offers a free hosting plan with decent resources, perfect for small projects, prototypes, and academic websites. This helped us avoid hosting costs during development and testing phases.

2. **Easy Deployment Process**: The platform simplifies deployment by allowing direct integration with GitHub repositories. This means every time we update our code on GitHub, Render automatically deploys the changes without extra steps.

3. **Supports Full Stack Applications:** Render is compatible with frontend frameworks like React and backend technologies like Node.js, Express, and MongoDB. This made it ideal for our MERN stack-based Faculty Appraisal System.

4. **Automatic SSL(HTTPS):** SSL is automatically provided by Render, securing our website with HTTPS without needing to manually install certificates. This improves trust and security, especially for login-based systems.

5. **Custom Domain Support:** Render allows us to connect a custom domain name to the deployed app easily. This gave our project a more professional identity and branding.

6. **Automatic Build and Deployment from Github:** Once linked to our GitHub repository, Render automatically builds and deploys the app after each push. This ensures the live version of our site is always in sync with the latest code.

7. **Scalable Infrastructure:** As traffic increases or app requirements grow, Render allows seamless scaling of resources. This future-proofing made it a better choice than some other limited free hosting platforms.

8. **User friendly Dashboards and real time logs:** The Render dashboard is simple to navigate and provides real-time logs for every deployment. This makes it easier to track errors and monitor app performance during and after deployment.

Render provides a seamless and developer-friendly platform for hosting full-stack applications with easy deployment, scalability, and built-in security features. Its integration with GitHub, free hosting tier, and automatic updates make it an excellent choice for efficient and hassle-free website deployment.

# Chapter 3

# System Design

## 3.1 System Architecture

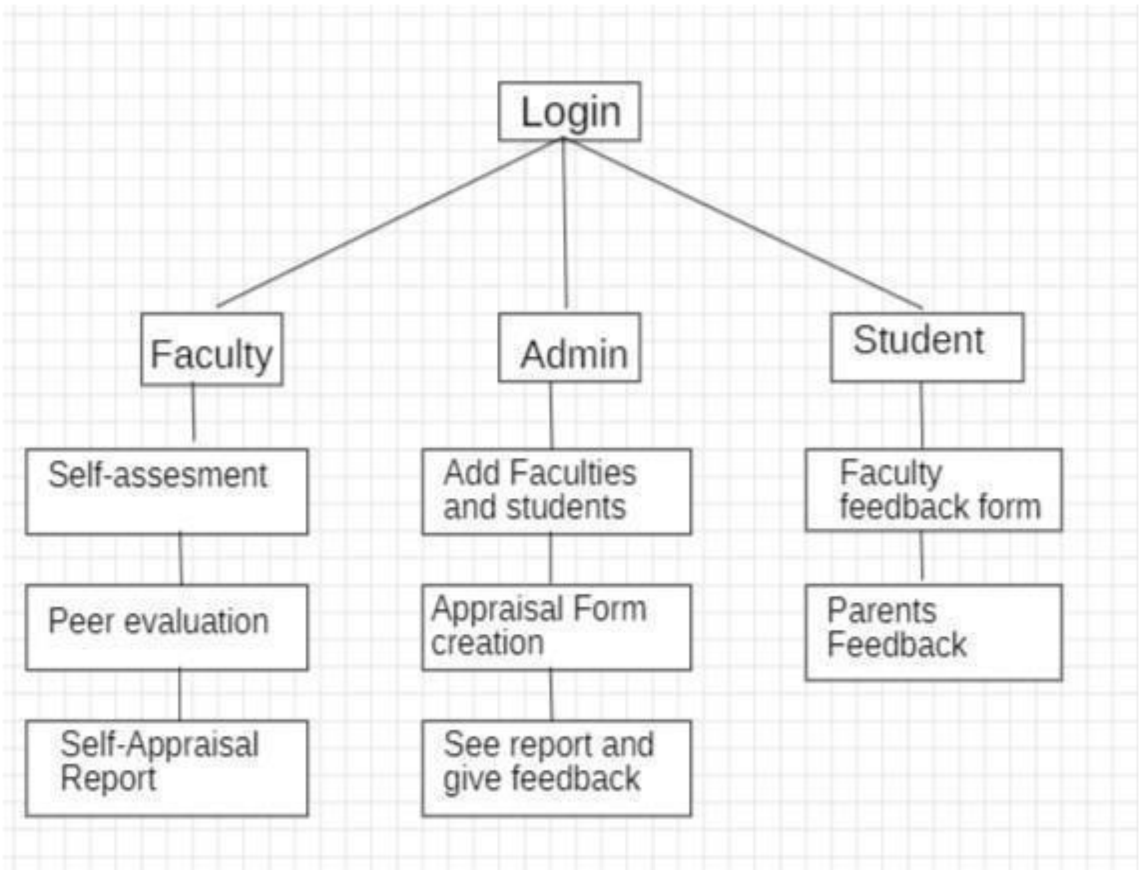


Fig 1: System Architecture

The data design for the project is structured to support three distinct user roles: faculty, admin, and student. Each role has access to specific functionalities that facilitate the appraisal process and ensure a seamless flow of data. This design ensures that the right data is accessible to the appropriate users.

## Faculty Login

Faculty members access the system through a dedicated login page that provides them with options for self-appraisal, peer evaluation, and viewing self-appraisal reports. The data design for the faculty module includes the following components:

- **Self-assessment**: Stores self-evaluation information submitted by the faculty, including reflections on teaching effectiveness, research activities, and contributions to the institution.
- **Peer Evaluation**: Captures feedback from peers regarding a faculty member's performance, focusing on aspects such as collaboration, mentoring, and professional conduct.
- **Self-Appraisal Reports**: Aggregates self-appraisal and peer evaluation data to generate comprehensive reports that faculty members can review to understand their performance metrics and areas for improvement.

## Admin Login

The admin login page is designed to provide administrative users with the tools needed to manage the appraisal system. The functionalities include adding faculties and students, creating appraisal forms, viewing reports, and providing feedback. The data design for the admin module includes:

- **Add faculties and students**: Maintains records of all users in the system, including faculty and students, with details such as roles, permissions, and account status.
- **Appraisal Form Creation**: Stores the templates and criteria used for appraisal forms, which can be customized by the admin based on the evaluation needs of the institution.
- **See reports and give feedback**: Contains data related to the performance reports of faculty members, as well as feedback provided by the admin. This ensures that performance evaluations are documented and actionable feedback is available to faculty members.

## Student Login

Students access the system through a dedicated login page that allows them to provide feedback on faculty performance and submit feedback from their parents. The data design for the student module includes:

- **Faculty Feedback Form:** Captures student evaluations of faculty performance, focusing on teaching effectiveness, engagement and support. This feedback is crucial for the appraisal process as it provides direct insights from the student's perspective

- **Parent's Feedback**: Stores feedback submitted by parents regarding faculty performance, offering an additional dimension to the appraisal process by incorporating the views of the broader community.

# Chapter 4

# Implementation and Testing

## 4.1 Code

### Faculty Self Assesment Schema.js

```
const mongoose = require('mongoose');
const FacultyAppraisalSchema=new mongoose.Schema({
    id: { type: String, required: true, unique: true},
    basicInformation: {
        fullName: { type: String, required: true },
        facultyId: { type: String, required: true, },
        email: { type: String, required: true },
        dateOfBirth: { type: Date, required: true },
        dateOfJoining: { type: Date, required: true },
        qualification: { type: String, required: true },
        designation: { type: String, required: true },
        currentSalary: { type: Number, required: true },
        additionalQualification: { type: String },
        teachingMethodology:[type:String}],
        teachingExperience: [
            {
                duration: { type: String, required:true },
                organization: { type: String,required:true},
                designation: { type: String, required: true},
                jobResponsibilities: { type: String, required: true }}],

    workloadAndSubjectsTaught: {
      underGraduate: {
        lectures: { type: Number },
        practicals: { type: Number },
        subjects: { type: String }
      },
      postGraduate: {
        lectures: { type: Number },
        practicals: { type: Number },
                subjects: { type: String }
      },
      certificateCourse: {
        lectures: { type: Number },
                practicals: { type: Number },
        subjects: { type: String }
      }
    }
  },
    academicAndProfessionalGrowth: {
        researchProject: { type: String },
        publications: { type: String },
        seminarWorkshop: { type: String },
        orientation: { type: String },
        activities: { type: String },
        extraActivities: { type: String },
        keyAchievements: [{ type: String }]
    },

     participationInActivities: {
        coCurricular: { type: String },
        communityService: { type: String },
        collegeAdmin: { type: String },
        improvementAreas: [{ type: String }]
    },


     selfAppraisal: {
        attendance: { type: Number, min: 1, max: 5 },
        leadershipAbility: { type: Number, min: 1, max: 5 },
        abilityToMeetDeadlines: { type: Number, min: 1, max: 5 },
        qualityOfWork: { type: Number, min: 1, max: 5 },
        teamworkAbility: { type: Number, min: 1, max: 5 },
        studentsFeedback: { type: Number, min: 1, max: 5 },
        studentsPerformanceIndex: { type: Number, min: 1, max: 5 }
    },


       performanceAppraisalReport: {
        Objectives: { type: String },
        Acheivements: { type: String },
        AcademicActivities: { type: String },
        Researchactivities: { type: String },
        ExtraCorricular: { type: String },
        futurePlans: { type: String },
        AnyOtherComment: { type: String },
        place: { type: String, required: true },
        date: { type: String, required: true }
    },


       uploads: {
        facultyImage: { type: String },
        signature: { type: String }
    },
    hasVoted: { type: Boolean, default: false },
});
```

```
module.exports = mongoose.model('FacultyAppraisal', FacultyAppraisalSchema);
```

## Server.js

```
require('dotenv').config();
const express = require('express');
const bodyParser = require('body-parser');
const db = require('./db');
const path = require('path');
const cookieParser = require('cookie-parser');
const cors = require('cors');
const updatePasswordRoute= require('./routes/
update_password');
const { verifyToken, checkRole } = require('./jwt');
const userSchema = require('./models/user')
const bcrypt = require('bcrypt');



const PORT = process.env.PORT || 5000;
const app = express();




app.use(cookieParser());


// Middleware setup
app.set('view engine', 'ejs');
app.set('views', path.join(_dirname, 'views')); // Ensure the views directory is set correctly
app.use(bodyParser.json({ limit: '10mb' })); // Adjust the size as per your requirement
app.use(bodyParser.urlencoded({ limit: '10mb', extended: true }));
app.use(cors()); // Allow all origins (for development purpose)
app.use(express.static('public'));
app.use(updatePasswordRoute);
app.get('/', (req, res) => { res.redirect('/login'); });

app.post('/logout', (req, res) => {
   res.clearCookie('token'); // Clear the token cookie
   res.status(200).send({ message: 'Logged out successfully' });
});


// Import the Routes
const loginRoutes = require('./routes/loginRoutes');
const adminRoutes = require('./routes/adminRoutes');
const studentRoutes = require('./routes/studentRoutes');
const facultyRoutes = require('./routes/facultyRoutes');
const useFacultyRoutes = require('./routes/useFaculty
Routes');
const useStudentRoutes = require('./routes/
useStudentRoutes');


// Example routes with role-based access control
app.use('/', loginRoutes);
app.use('/admin', verifyToken, checkRole(['admin']), adminRoutes);
app.use('/admin', verifyToken, checkRole(['admin']), studentRoutes);
app.use('/admin', verifyToken, checkRole(['admin']), facultyRoutes);


app.use('/faculty', verifyToken, checkRole(['faculty']), useFacultyRoutes);
app.use('/student', verifyToken, checkRole(['student']), useStudentRoutes);

// Start server
app.listen(PORT, () => console.log(`Server running on http://localhost:${PORT}`));
```

## Database.js

```
const mongoose = require('mongoose');
require('dotenv').config();


const mongoURL = process.env.LOCAL_MONGODB_URL;

mongoose.connect(mongoURL);


const db = mongoose.connection;

db.on('connected', () => {
   console.log("Connected to MongoDB server");
});

db.on('error', (err) => {
   console.log("MongoDB connection error : ", err);
});


db.on('disconnected', () => {
   console.log("Disconnected from MongoDB server");
});



module.exports = db;
```

## 4.2 Testing Approach

### 4.2.1 Unit Testing

In this project, the Unit Testing section is crucial to ensure the functionality, accuracy, and reliability of individual components of the system. Unit testing refers to the process of testing small, isolated parts of the application (usually functions or methods) to verify that they work as expected before integrating them into the larger system.

Unit testing is an essential phase in the development of the Performance Based Faculty Appraisal System, as it helps in identifying and fixing errors in individual components early in the development cycle. The goal is to ensure that each module functions as intended under various conditions.

The system includes various modules like **faculty profile management**, **performance evaluation**, **feedback collection**, and **report generation**. Each of these modules has specific functions that need to be tested in isolation. For instance, the faculty profile management module includes functions for adding, updating, and deleting faculty details. These functions should be tested with valid, invalid, and edge-case inputs to ensure they handle all scenarios correctly.

For the **performance evaluation** module, unit tests would focus on verifying the accuracy of calculations (e.g., average performance scores) and correct classification based on evaluation criteria. Testing should check if the system handles performance metrics correctly for various types of data input (e.g., high, low, or missing scores).

Similarly, the **feedback collection** module should be tested for different feedback scenarios, ensuring that feedback is properly stored, retrieved, and linked to the appropriate faculty member.

Unit tests can be automated using tools like **JUnit** for Java, **pytest** for Python, or **Mocha** for JavaScript, depending on the technology stack used. Each test case should cover specific functionality, checking expected outcomes for both valid and invalid inputs, ensuring that exceptions or errors are handled appropriately.

Unit testing ensures the overall reliability of the system, contributes to smoother integration, and reduces the cost of bug fixing later in the development process.

### 4.2.2 Integration Testing

In this project, the **Integration Testing** phase ensures that the various components and modules of the system work together as intended when integrated. While unit testing focuses on individual functions, integration testing validates the interaction between these modules, ensuring seamless communication and data flow. Below is an outline of what to include in the Integration Testing part of your project.

Integration testing plays a crucial role in verifying that the different modules of the Performance Based Faculty Appraisal System work cohesively when integrated. The system consists of several interconnected modules such as **faculty profile management**, **performance evaluation**, **feedback collection**, and **report generation**. Each of these modules operates independently, but once combined, their interactions need to be thoroughly tested.

During integration testing, the primary objective is to validate the data flow between these modules and ensure that the overall system performs as expected when the modules are combined. For example, when a faculty member's profile is updated in the **faculty profile management** module, this update should automatically reflect in the **performance evaluation** module, which requires accurate faculty data for generating evaluations. Integration testing will check if data is correctly passed between these two modules without errors.

Another important aspect is ensuring that feedback collected through the **feedback collection** module is seamlessly integrated with the performance evaluation data to generate accurate reports in the **report generation** module. The integration tests should confirm that data is correctly synchronized between these modules, ensuring that reports reflect up-to-date and accurate information.

Test cases for integration testing will simulate real-world use cases, such as updating faculty profiles, submitting performance evaluations, collecting feedback, and generating reports. The tests should verify if the system can handle multiple, simultaneous requests, and check for potential data integrity issues or performance bottlenecks.

Automated tools like **Postman** for API testing or **Selenium** for testing web-based interfaces can be used for integration testing. The goal is to ensure that all integrated components function as expected and no data is lost or corrupted during the exchange between modules.

# Chapter 5

# Results and Discussions

## Login Page

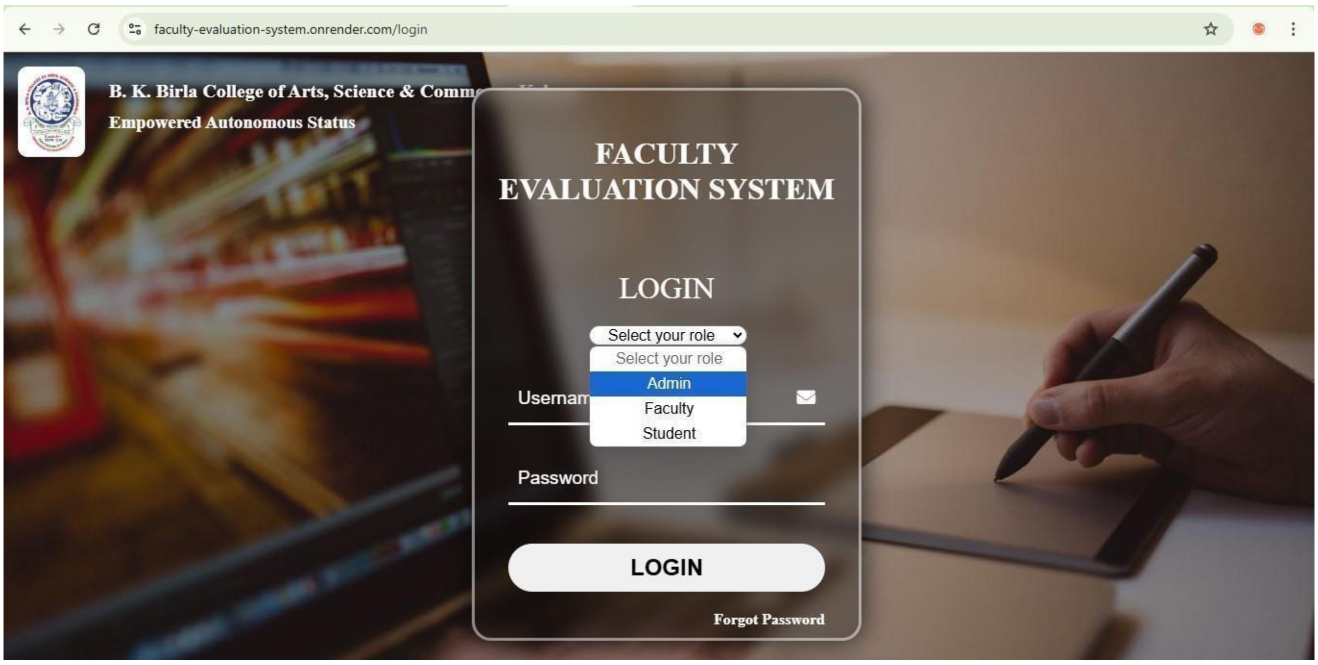

faculty-evaluation-system.onrender.com/login
B. K. Birla College of Arts, Science & Comm
Empowered Autonomous Status
FACULTY EVALUATION SYSTEM
LOGIN
Select your role
Select your role
Admin
Faculty
Student
Usernam
Password
LOGIN
Forgot Password

Fig 2: Login Page

### Functionality of the Login Page

The login page serves as the gateway to the **Performance Based Faculty Appraisal System** and provides three distinct login options: **Admin**, **Faculty**, and **Student**. Each option has specific access levels and functionality tied to it.

- **Admin Login**: The admin has full control over the system, including the ability to manage faculty profiles, oversee performance evaluations, generate reports, and monitor the system's overall usage. After successful login, admins are directed to the administrative dashboard where they can perform various administrative tasks.
- **Faculty Login**: Faculty members can log in to view their own performance evaluations, submit self-assessments, and access feedback provided by students and peers. They also have the option to update their personal details. The system ensures that faculty users can only view their own data, maintaining privacy and security.
- **Student Login**: Students can log in to submit feedback on faculty members and view the status of their evaluations. They have restricted access to ensure they cannot access or alter any data beyond their feedback submissions.

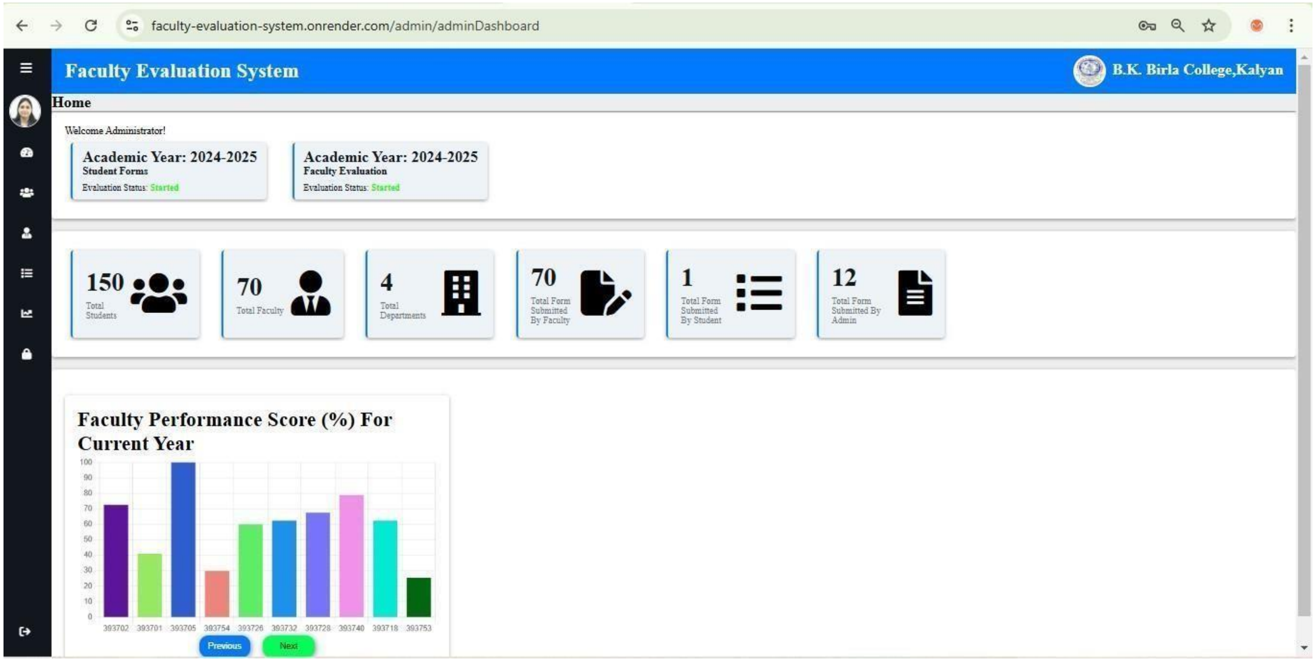
The login page utilizes role-based authentication to ensure that users are redirected to their respective dashboards after authentication. The system validates credentials through a secure process, utilizing **hashing** and **encryption** for password protection. Additionally, the system includes **error handling** to notify users of invalid login attempts or forgotten credentials.

**Discussion on Login Functionality:**

The inclusion of role-based login functionality enhances security and ensures that users only have access to the features relevant to their role. The clear separation between the roles (Admin, Faculty, and Student) ensures smooth user experience and data integrity. Any issues with the login process, such as failed authentication attempts, are promptly addressed through feedback messages, preventing users from getting stuck in the process.

This login system is an essential part of maintaining the security and integrity of the system, as it ensures that only authorized users can access sensitive information. The system's robust login functionality guarantees appropriate access control, which is crucial for both privacy and data protection in an academic environment.

## Admin Dashboard



faculty-evaluation-system.onrender.com/admin/adminDashboard

Faculty Evaluation System

B.K. Birla College,Kalyan

Home

Welcome Administrator!

Academic Year: 2024-2025
Student Forms
Evaluation Status: Started

Academic Year: 2024-2025
Faculty Evaluation
Evaluation Status: Started

150 Total Students

70 Total Faculty

4 Total Departments

70 Total Form Submitted By Faculty

1 Total Form Submitted By Student

12 Total Form Submitted By Admin

Fig 3: Admin Dashboard

## Admin Login Functionality in "Performance Based Faculty Appraisal System"

The **Admin Login** functionality is a crucial component of the **Performance Based Faculty Appraisal System**, providing administrators with full control and access to manage and monitor various aspects of the system. The admin login page is designed to offer a secure, authenticated entry point for users with administrative privileges, ensuring that only authorized personnel can access sensitive data and perform critical operations. Upon successful login, the admin is redirected to a dedicated **Admin Dashboard**, which serves as a centralized control panel for overseeing the system's various features and functionalities.

### Key Features Accessible by Admin:

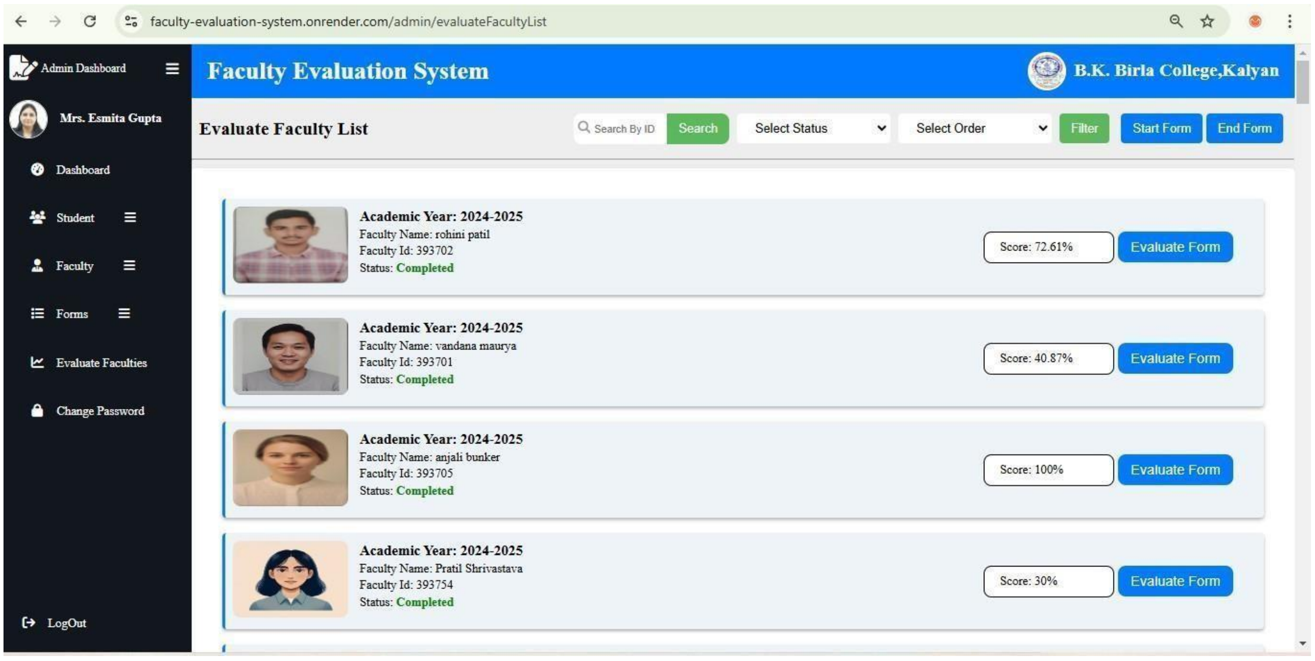
1. **View Total Students**: The admin can easily view the total number of registered students in the system. This information helps the admin track the number of active students who have the ability to submit feedback on faculty members. It also allows for the monitoring of student participation in the appraisal process.
2. **View Total Faculties**: The admin has access to a comprehensive list of all faculty members, with the ability to see key details such as faculty names, departments, and performance evaluations. This feature helps the admin manage and oversee faculty performance and ensures that the evaluation process is conducted effectively across all faculty members.
3. **View Submitted Forms**: Admins can monitor the number of forms (performance evaluation forms, feedback forms) submitted by students and faculty. This feature helps track the progress of the appraisal process, ensuring that all forms are submitted on time. The admin can also monitor if any forms are pending or incomplete, prompting reminders where necessary.
4. **View Departments**: The admin can view and manage faculty groups by department. This allows for a structured approach in analyzing faculty performance within each department and enables admins to generate department-specific reports. By viewing department data, the admin can gain insights into departmental performance trends, resource allocation, and identify areas for improvement.
5. **Performance Graph of Faculties (Current Year)**: One of the most powerful features available to the admin is the ability to visualize faculty performance through **performance graphs**. These graphs display the overall performance of each faculty member for the current year based on the evaluation criteria (e.g., teaching quality, research output, feedback from students). The graphs allow the admin to quickly assess trends and identify high-performing and underperforming faculty members. These visualizations assist in data-driven decision-making and ensure transparency in the faculty appraisal process.

**Process Flow:**

- **Authentication**: Admin enters their credentials (username and password) on the login page. Upon validation, the system grants access to the admin dashboard.
- **Data Access**: After logging in, the admin is presented with an intuitive dashboard displaying key metrics like the total number of students, faculty, submitted forms, departments, and performance graphs.
- **Data Monitoring**: The admin can view and filter the data based on specific criteria such as department or performance score.
- **Report Generation**: Admin can also generate detailed reports based on the data available in the system, offering insights into faculty performance, department-level evaluations, and the overall success of the appraisal process.

The **admin login functionality** is secured using modern authentication mechanisms, including **password encryption** and **session management**. To further enhance security, two-factor authentication (2FA) may be implemented to ensure that only authorized admins can access the dashboard. The user interface is designed to be intuitive and easy to navigate, ensuring that admins can quickly find and interpret the data they need without unnecessary complexity.

The **Admin Login** functionality serves as the backbone of the system, allowing administrators to maintain control over the performance appraisal process. By having access to critical data, such as the number of students, faculty, submitted forms, and department-specific insights, the admin can ensure that the system is operating efficiently and that all aspects of the faculty evaluation process are running smoothly. The **performance graphs** provide a powerful tool for analyzing trends, identifying areas of improvement, and making informed decisions to improve faculty development. This centralized, data-driven approach enhances the overall effectiveness of the Performance Based Faculty Appraisal System.



Evaluate Faculty List

**Academic Year: 2024-2025**
Faculty Name: rohini patil
Faculty Id: 393702
Status: Completed
Score: 72.61%

**Academic Year: 2024-2025**
Faculty Name: vandana maurya
Faculty Id: 393701
Status: Completed
Score: 40.87%

**Academic Year: 2024-2025**
Faculty Name: anjali bunker
Faculty Id: 393705
Status: Completed
Score: 100%

**Academic Year: 2024-2025**
Faculty Name: Pratil Shrivastava
Faculty Id: 393754
Status: Completed
Score: 30%

Fig 4: Faculty Evaluation List

- **Evaluate Submitted Forms**: Admins can review and assess performance appraisal forms submitted by faculty members. They can verify the completeness and accuracy of the submitted data to ensure consistency and reliability.

- **Check Evaluation Status**: The admin can easily view the status of each evaluation form, marked as **Complete** or **Incomplete**. This helps the admin track the progress of the appraisal process and send reminders if necessary.

- **Sort by Evaluation Scores**: Admins can sort the faculty evaluation forms based on performance scores in both **ascending** and **descending** order. This functionality aids in identifying high-performing faculty members and those who may need additional support.

- **Start/End Evaluation for the Current Academic Year**: The admin has the ability to start or end the evaluation process for the academic year, marking the period during which the faculty appraisals are active.

## Faculty Dashboard

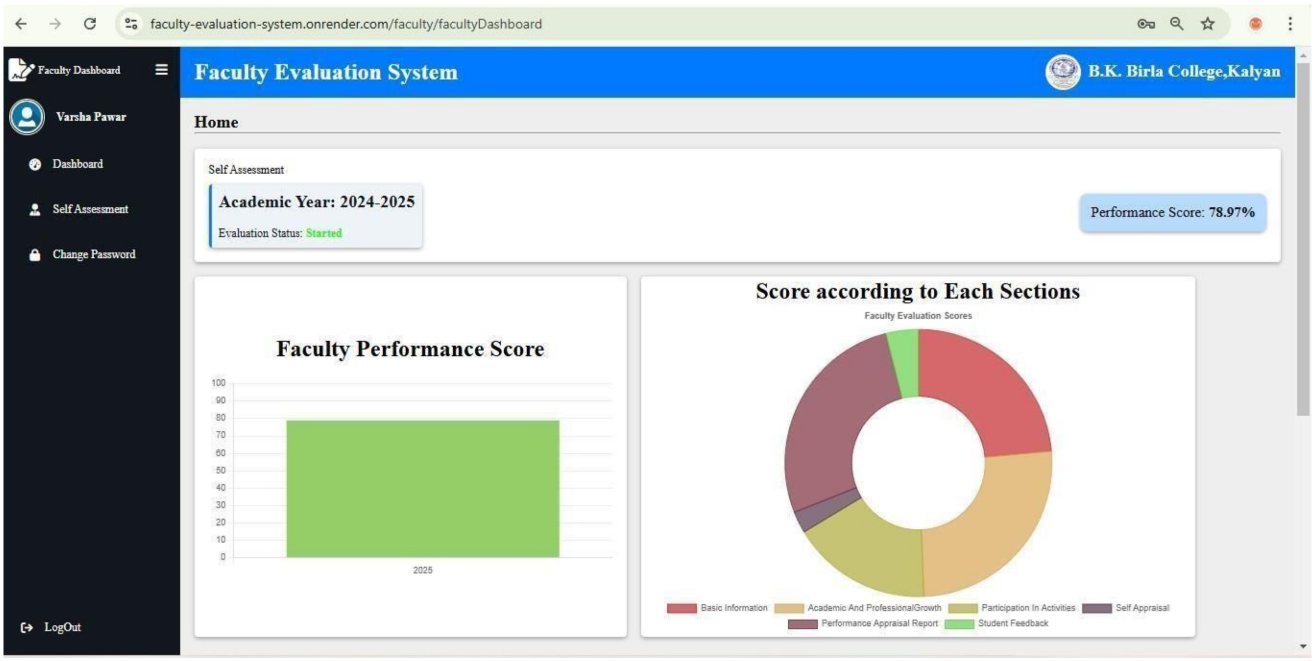

faculty-evaluation-system.onrender.com/faculty/facultyDashboard
Faculty Dashboard
Varsha Pawar
Dashboard
Self Assessment
Change Password
LogOut
Faculty Evaluation System
B.K. Birla College,Kalyan
Home
Self Assessment
Academic Year: 2024-2025
Evaluation Status: Started
Performance Score: 78.97%
Faculty Performance Score
100
90
80
70
60
50
40
30
20
10
0
2025
Score according to Each Sections
Faculty Evaluation Scores
Basic Information
Academic And ProfessionalGrowth
Participation In Activities
Self Appraisal
Performance Appraisal Report
Student Feedback

Fig 5: Faculty Dashboard

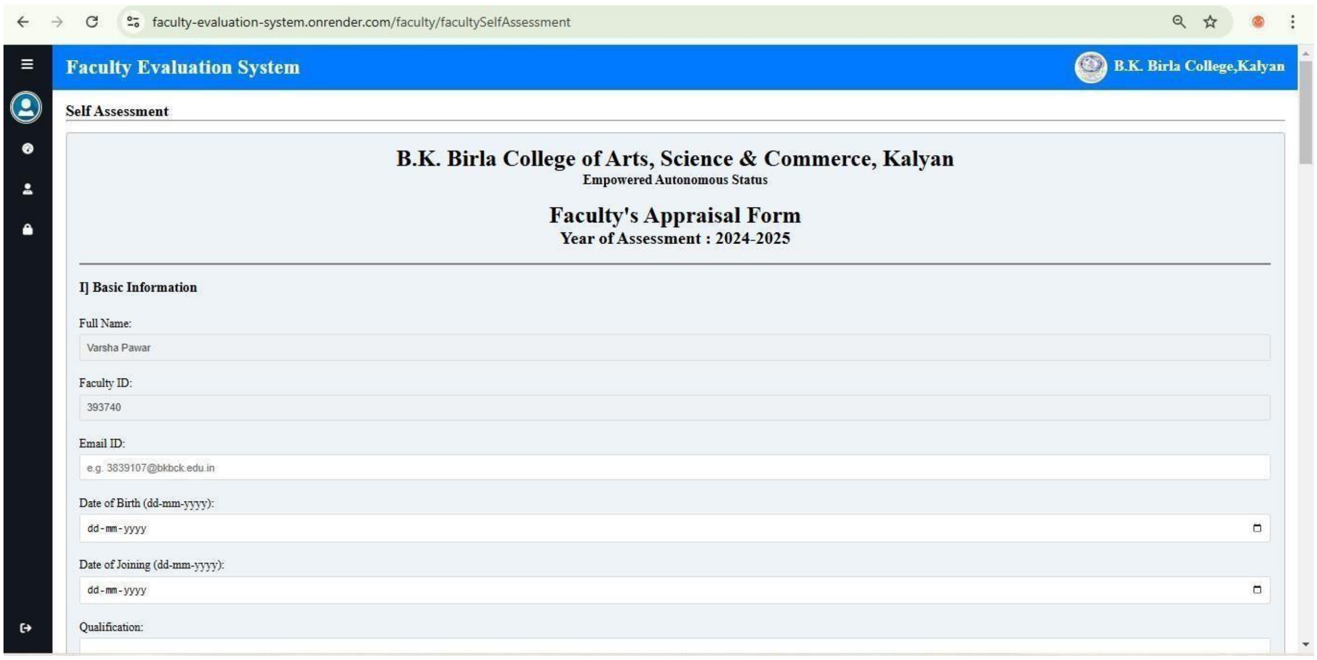
### Faculty Login Functionality in "Performance Based Faculty Appraisal System"

The **Faculty Login** functionality in the **Performance Based Faculty Appraisal System** provides faculty members with a personalized, secure interface to view and assess their performance. Once logged in, faculty members can access essential features related to their own performance evaluations.

**Key Features:**

1. **Self-Assessment Graph**: Faculty members can view a **self-assessment graph**, which visually represents their own evaluations across different criteria. This graph allows faculty to track their performance trends over time, providing insights into their strengths and areas for improvement.
2. **Individual Component Score (Pie Chart)**: A **pie chart** displays the breakdown of the faculty member's scores across various performance components, such as teaching quality, research, student feedback, and other relevant metrics. The chart offers a clear, visual representation of how each component contributes to their overall performance, enabling faculty to better understand their evaluation results.

These features provide faculty with the tools to reflect on their performance, encouraging self-improvement and fostering transparency in the appraisal process. The **faculty login** ensures that each individual has secure access to their personal data and evaluation results, facilitating informed decisions and continuous professional development.


faculty-evaluation-system.onrender.com/faculty/facultySelfAssessment

Faculty Evaluation System

B.K. Birla College,Kalyan

Self Assessment

B.K. Birla College of Arts, Science & Commerce, Kalyan
Empowered Autonomous Status

Faculty's Appraisal Form
Year of Assessment : 2024-2025

I] Basic Information

Full Name: Varsha Pawar

Faculty ID: 393740

Email ID: e.g. 3839107@bkbck.edu.in

Date of Birth (dd-mm-yyyy): dd-mm-yyyy

Date of Joining (dd-mm-yyyy): dd-mm-yyyy

Qualification:

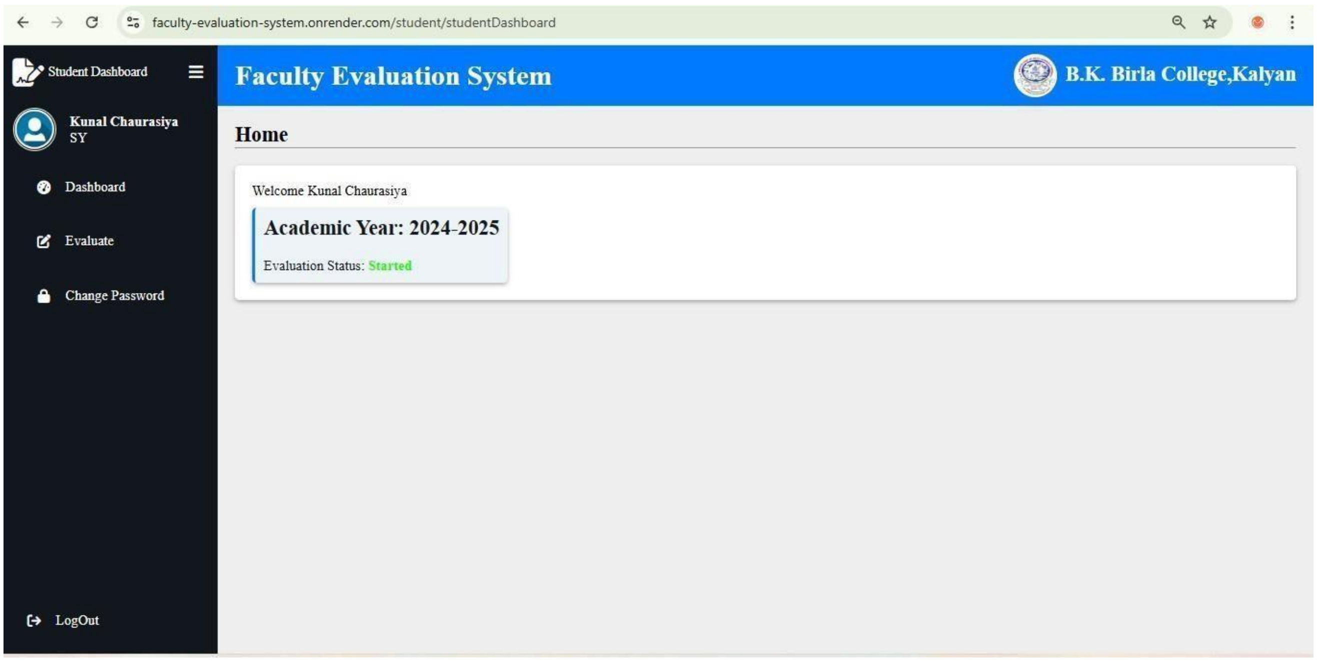
Fig 6: Faculty’s self-assessment form

The **Faculty Login** functionality allows faculty members to securely access their personalized dashboard. Once logged in, faculty can fill out their **self-assessment form**, providing an evaluation of their own performance across various criteria. The self-assessment form includes sections on teaching effectiveness, research contributions, and other relevant metrics. After completing the form, faculty members can **submit it to the admin** for review and inclusion in the overall appraisal process. This feature ensures that faculty have a chance to reflect on and contribute to their own performance evaluation in a structured manner.

After logging in, faculty members can view and complete their **self-assessment form**, which is designed to capture their reflections on various aspects of their performance, such as teaching effectiveness, research activities, student feedback, and other evaluation criteria.

The self-assessment form allows faculty to rate their own performance, providing a comprehensive overview of their achievements and areas for growth. Once the form is completed, faculty can **submit** it directly to the **admin** for review. This submission becomes part of the faculty's overall appraisal process, allowing the admin to incorporate the faculty's self-evaluations alongside feedback from other sources.

## Student Dashboard


faculty-evaluation-system.onrender.com/student/studentDashboard
Student Dashboard
Kunal Chaurasiya
SY
Dashboard
Evaluate
Change Password
LogOut
Faculty Evaluation System
B.K. Birla College,Kalyan
Home

Welcome Kunal Chaurasiya

Academic Year: 2024-2025

Evaluation Status: Started

Fig 7: Student Dashboard

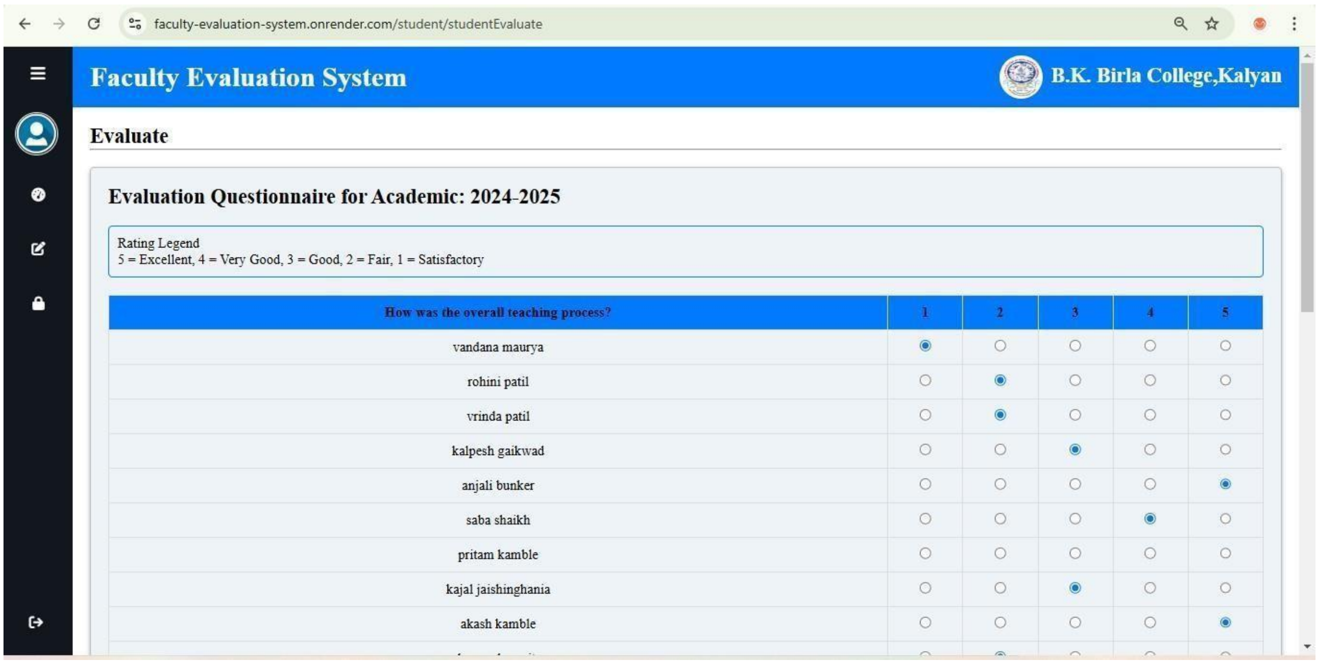
### Student Login Functionality in "Performance Based Faculty Appraisal System"

The **Student Login** functionality in the **Performance Based Faculty Appraisal System** provides students with a secure, personalized entry point to participate in the faculty evaluation process. After logging in with their credentials, students are granted access to a dedicated interface where they can provide feedback on faculty performance.

**Key Features:**

1. **Submit Feedback**: Students can evaluate faculty members based on specific criteria such as teaching effectiveness, communication skills, course content delivery, and overall engagement. The feedback form is designed to capture detailed insights into the faculty's performance during the academic term.
2. **View Feedback Status**: Students can track whether they have completed their feedback submissions. The system indicates if their feedback has been submitted, ensuring that students are aware of their participation status in the appraisal process.
3. **Confidentiality and Anonymity**: To promote honest and constructive feedback, the system ensures that all student submissions are anonymous. This confidentiality encourages students to provide genuine evaluations without fear of repercussions.

The **Student Login** functionality plays a critical role in the system by facilitating active student involvement in the appraisal process. It empowers students to contribute to the improvement of teaching quality while maintaining privacy and ease of use throughout the evaluation period.


faculty-evaluation-system.onrender.com/student/studentEvaluate
Faculty Evaluation System
B.K. Birla College,Kalyan

Evaluate

Evaluation Questionnaire for Academic: 2024-2025

Rating Legend
5 = Excellent, 4 = Very Good, 3 = Good, 2 = Fair, 1 = Satisfactory

| How was the overall teaching process? | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| vandana maurya | ● | ○ | ○ | ○ | ○ |
| rohini patil | ○ | ● | ○ | ○ | ○ |
| vrinda patil | ○ | ● | ○ | ○ | ○ |
| kalpesh gaikwad | ○ | ○ | ● | ○ | ○ |
| anjali bunker | ○ | ○ | ○ | ○ | ● |
| saba shaikh | ○ | ○ | ○ | ● | ○ |
| pritam kamble | ○ | ○ | ○ | ○ | ○ |
| kajal jaishinghania | ○ | ○ | ● | ○ | ○ |
| akash kamble | ○ | ○ | ○ | ○ | ● |

Fig 8: Student’s Feedback

**Key Features:**

1. **Submit Feedback**: Students can rate faculty on various aspects, such as teaching effectiveness, clarity of communication, engagement, and course content. The feedback is designed to capture the students' overall experience with the faculty.
2. **Anonymous Feedback**: To ensure candid and constructive evaluations, student feedback is submitted anonymously, promoting transparency and honesty without fear of identification or retaliation.
3. **Track Submission Status**: Students can check if they have submitted feedback for all their faculty members, ensuring complete participation in the evaluation process.

This functionality empowers students to contribute to the quality of education while ensuring their feedback remains confidential and impactful in improving faculty performance.

# Chapter 6
# Conclusions and Future Work

## 6.1 Conclusion

The "Performance-Based Faculty Appraisal System" successfully integrates key performance indicators (KPIs) into an objective and transparent framework for evaluating faculty members. By leveraging data from multiple sources such as teaching quality, research contributions, and student feedback, the system offers a comprehensive assessment that not only helps identify areas of improvement for faculty members but also supports the decision-making process for academic leadership. The system enhances fairness and consistency in the evaluation process by eliminating biases that may be inherent in traditional appraisal methods.

Through this project, it has been demonstrated that automating faculty appraisal through data-driven insights can streamline administrative tasks, improve faculty engagement, and promote a culture of continuous improvement. The system’s flexibility allows it to be adapted for different educational institutions, making it a versatile tool for diverse academic environments.

## 6.2 Future Work

While the current version of the system provides a solid foundation, there are several areas for future enhancement. One potential direction is to incorporate real-time data integration, where faculty performance metrics are updated continuously to provide more accurate and up-to-date feedback. Additionally, incorporating peer evaluations and collaborative contributions into the appraisal system can further enhance the comprehensiveness of faculty assessments.

Another future improvement could be the development of predictive analytics to foresee trends in faculty performance, allowing institutions to identify areas that require intervention or support. Enhancing the user interface and experience for both faculty members and administrators will also be critical to ensure broader adoption and usability.

Furthermore, conducting pilot studies across different institutions can help refine the system's effectiveness and gather feedback to ensure that it meets the unique needs of each academic environment. These efforts will ensure that the system evolves to become an essential tool for fostering faculty development and institutional growth.

# Chapter 7

## References

[1] Node.js Documentation: https://youtu.be/ZLxOUw2ougo?si=3gZ_kGMq2z0_y9_c

[2] ReactJS Documentation: https://youtu.be/4z9bvgTlxKw?si=9YvSfYj57bGYqBYH

[3] MongoDB Documentation – https://www.mongodb.com/docs/