

Staff Appraisal System

MCA(Master in Computer Application)

**LOVELY PROFESSIONAL UNIVERSITY
PHAGWARA, PUNJAB**



Submitted By:

Sahil Rana – 12216166

Roshan Nayak – 12216172

Abhishek Rana – 12216226

Submitted To :

Dr. Tarandeep Singh Walia (25153)

To whom so ever it may concern

We **Sahil Rana,12216166, Roshan Nayak 12216172, Abhishek Rana 12216226** hereby declares that the work doneby us on “**Staff Appraisal System**”. Lovely professional University, Phagwara, Punjab, is a record of original work for the partial fulfillment of the requirements for the award of the degree,**MCA.**

Sahil Rana - 12216166

Roshan Nayak – 12216172

Abhishek Rana - 12216226

ACKNOWLEDGEMENT

We would like to express our profound gratitude to Dr. Tarandeep Singh Walia of School of Computer Application Department, for their contributions to the completion of our project titled Staff Appraisal System.

We would like to express our special thanks to our mentor **Dr. Tarandeep Singh Walia** for his time and efforts he provided throughout the year. Your useful advice and suggestions were really helpful to me during the project's completion. In this aspect, I am eternally grateful to you.

I would like to acknowledge that this project was completed entirely by us.

Sahil Rana - 12216166

Roshan Nayak – 12216172

Abhishek Rana - 12216226

Table of Contents

Sr. No.	Title	Page No.
1. Executive Summary		
1.1	Project Overview	5
1.2	Purpose and Scope of this Specification	6
2. Service Description		
2.1	User Characteristics	8
2.2	Constraints	10
3. Requirements		
3.1	Functional Requirements	12
3.2	User Interface Requirements	11
3.3	Usability	16
3.4	Performance	17
3.5	Maintainability	18
4. Data Flow Diagram		
4.1	DFD	20
4.2	References	21

1. Executive Summary

1.1 Project Overview :

An organization's ability to effectively manage its human resources depends in large part on the implementation of a comprehensive Staff Appraisal System. It's a methodical and organised method for checking up on workers' progress and accomplishments. A staff appraisal system is a framework for providing feedback, acknowledging achievements, and encouraging people to do better in their roles within the organisation.

1.2 Scope:

Employee performance and growth are only two of the many aspects of an organisation that fall within the purview of a staff assessment system. Organisational size, kind, and goals all have a role in determining the extent. Key components of an evaluation scheme for employees are as follows:

1. Performance Evaluation
2. Goal Setting and Expectation Management
3. Data Collection and Analysis

2. Project Description:

The explanation will dive deep into the structure and value of a staff evaluation programme, stressing its importance in today's business administration.

Purpose and Objectives:

- The staff appraisal system is designed to achieve several key objectives, including:
 - Evaluating individual employee performance.
 - Identifying strengths and weaknesses.
 - Setting performance goals and expectations.
 - Facilitating career development and growth.
 - Recognizing and rewarding outstanding contributions.
 - Enhancing communication between employees and managers.
 - Ensuring alignment with organizational goals and values.

2. Benefits:

- Implementing an effective staff appraisal system offers numerous advantages to both employees and the organization:
 - Improved employee performance and productivity.
 - Increased job satisfaction and motivation.
 - Enhanced communication and transparency.
 - Better alignment of individual and organizational goals.
 - Identification of training and development needs.
 - Fair and consistent recognition and reward systems.
 - Data for talent management and succession planning.

3. Key Components:

- A typical staff appraisal system consists of the following essential components:
 - Performance Standards and Criteria: Clear and measurable expectations for employee performance.
 - Performance Appraisal Forms: Documents or software used to record evaluations and feedback.
 - Self-Assessment: Employees' self-evaluation of their performance.
 - Managerial Assessment: Evaluations provided by supervisors or managers.
 - Goal Setting: Setting performance goals and development plans.
 - Feedback and Communication: Constructive feedback and discussion between employees and managers.
 - Performance Ratings: Assigning scores or ratings to various aspects of performance.
 - Development Plans: Identifying areas for improvement and plans for growth.

4. Process:

- The staff appraisal process typically involves several steps:
 - Goal Setting: Establishing performance objectives and expectations.
 - Monitoring: Regularly tracking and documenting employee performance.
 - Evaluation: Assessing performance against predetermined criteria.
 - Feedback: Providing constructive feedback to employees.
 - Development Planning: Identifying training and development needs.
 - Performance Review: Conducting formal appraisal meetings.
 - Recognition and Reward: Acknowledging outstanding contributions.
 - Follow-up: Monitoring progress and adjusting goals as necessary.

5. Challenges and Considerations:

- Implementing a staff appraisal system may face challenges such as bias, subjectivity, resistance from employees, and time constraints. Therefore, it's essential to design a fair and transparent system, provide training to managers, and ensure that the process aligns with the organization's culture and values.

2. User Characteristics:

User characteristics for a Staff Appraisal System in Python can vary depending on the specific organization and its requirements. However, we can define some common user characteristics that are typically applicable to such a system:

User Roles:

Employees: These are the individuals who are being appraised. They may need to access the system to review their own performance reports.

Managers/Supervisors: Managers or supervisors are responsible for conducting employee appraisals and providing ratings. They may also need access to review the performance reports of their subordinates.

HR Administrators: HR personnel are responsible for managing employee data, overseeing the appraisal process, and generating organization-wide reports.

System Administrators: System administrators maintain and manage the appraisal system, including user accounts, security, and system configuration.

Access Levels:

Read-Only Users: Some users may have read-only access to view performance reports, but they cannot conduct appraisals or add new employees.

Appraisers: Appraisers have the ability to conduct appraisals for employees but may not have access to all system settings.

Administrators: Administrators have full access to the system, including the ability to add employees, conduct appraisals, and configure system settings.

Technical Proficiency:

Users may have varying levels of technical proficiency. Some may be highly skilled in using computer applications, while others may be less comfortable with technology.

Security Requirements:

Different users may have different security requirements. For example, HR administrators and system administrators may need heightened security access to protect sensitive employee data.

Frequency of Use:

Some users may interact with the system on a daily basis (e.g., managers conducting ongoing appraisals), while others may use it infrequently (e.g., employees reviewing annual performance reports).

Training Needs:

Users with different roles may require different levels of training to effectively use the system. For instance, employees may need training on how to access and review their reports, while HR administrators may require training on data management and system configuration.

Feedback and Reporting Needs:

Employees may want to provide feedback on the appraisal process, and the system should have a mechanism for capturing this feedback. Managers and HR administrators may need custom reporting features to analyze appraisal data.

Preferred Communication Channels:

Users may have preferred communication channels for system notifications or alerts. For example, some users may prefer email notifications, while others may prefer in-app messages.

Device Preferences:

Users may access the system on various devices, such as desktop computers, mobile devices, or tablets. The system should be designed to be responsive and accessible across different platforms.

Compliance Requirements:

Depending on the organization's industry and location, users may need to adhere to specific legal and regulatory compliance requirements related to performance appraisals and data protection.

2.2 Constraints

When designing and implementing a Staff Appraisal System in Python, several constraints and considerations need to be taken into account to ensure the system's functionality, security, and usability. Here are some common constraints:

Technology Stack: The choice of technology stack can be a constraint. If the organization has specific technology preferences or existing systems, the new system may need to be compatible or integrate with those technologies.

Compliance and Regulations: Many organizations must adhere to legal and regulatory requirements related to employee appraisals, data protection, and privacy (e.g., GDPR, HIPAA, etc.). The system must comply with these regulations.

Data Security: Protecting sensitive employee data is crucial. The system needs robust security measures to prevent unauthorized access and data breaches. This includes user authentication, data encryption, and access controls.

Scalability: The system should be designed to handle a growing number of employees and data over time. Scalability constraints may arise if the system cannot handle increased workloads.

Performance: The system should perform well even with large datasets. Slow or inefficient performance can be a significant constraint.

User Training and Adoption: Constraints related to user training and adoption are essential. If users find the system complex or hard to use, it can limit its effectiveness.

Integration: The system may need to integrate with other existing HR or organizational systems. Constraints related to data sharing and compatibility can be challenging.

Budget: Budget constraints can impact the development and maintenance of the system. Organizations may have limited resources available for system development and support.

Time Constraints: The project may have specific deadlines, and the development team must work within those constraints to deliver the system on time.

Hardware and Infrastructure: The availability and quality of hardware and infrastructure can be a constraint. The system may require specific hardware or infrastructure for optimal performance.

User Accessibility: The system should be accessible to all users, including those with disabilities. Accessibility constraints may include compliance with accessibility standards (e.g., WCAG).

Feedback and Iteration: Gathering user feedback and making iterative improvements is crucial. However, constraints on resources and time may limit the ability to quickly implement changes based on feedback.

System Dependencies: If the system relies on external services or APIs, constraints related to those dependencies can impact the system's functionality.

Testing and Quality Assurance: The system should undergo thorough testing and quality assurance, which can be constrained by time, resources, or the availability of suitable testing environments.

Documentation and Training Materials: Constraints on the availability of documentation and training materials can hinder users' ability to understand and effectively use the system.

Data Migration: If transitioning from an existing system, data migration can be complex and constrained by data formats, quality, and historical data.

Legacy Systems: Organizations with legacy systems may have constraints in terms of compatibility and migration to a new system.

3. Requirements:

3.1 Functional Requirements:

These are some important functional requirements of Student Appraisal System:

1. **Employee Information:** The system should have a place to keep all the important details about each employee, like their name, job, and contact information.
2. **Data Handling:-** There a module for cleaning or deleting the records of employee.

```
def delete_employee(self, emp_id):
    employee = next((emp for emp in self.employees if emp.emp_id == emp_id), None)
    if employee:
        self.employees.remove(employee)
        print(f'Employee with ID {emp_id} deleted.')
    else:
        print("Employee not found!")

def import_data_from_csv(self, csv_filename):
    categories = ['goals', 'objectives', 'responsibilities', 'work_quality', 'productivity', 'attitude']
    new_employees = []

    try:
        with open(csv_filename, 'r') as csv_file:
            csv_reader = csv.DictReader(csv_file)
            for row in csv_reader:
                emp_id = row['Employee ID']
                name = row.get('Name', 'Name Missing')
                salary = float(row.get('Salary', 0)) # Provide a default salary if missing

                existing_employee = next((emp for emp in self.employees if emp.emp_id == emp_id), None)

                if existing_employee:
                    print(f"Employee with ID {emp_id} already exists. Updating information.")
                    for category in categories:
                        if category in row:
                            score = float(row[category])
                            existing_employee.add_score(category, score)
                        else:
                            print(f"Missing data for category {category} for Employee ID {emp_id}. Please re-e")
                            return
                else:
                    employee = Employee(emp_id, name, salary)
                    for category in categories:
                        if category in row:
                            score = float(row[category])
                            employee.add_score(category, score)
                        else:
```

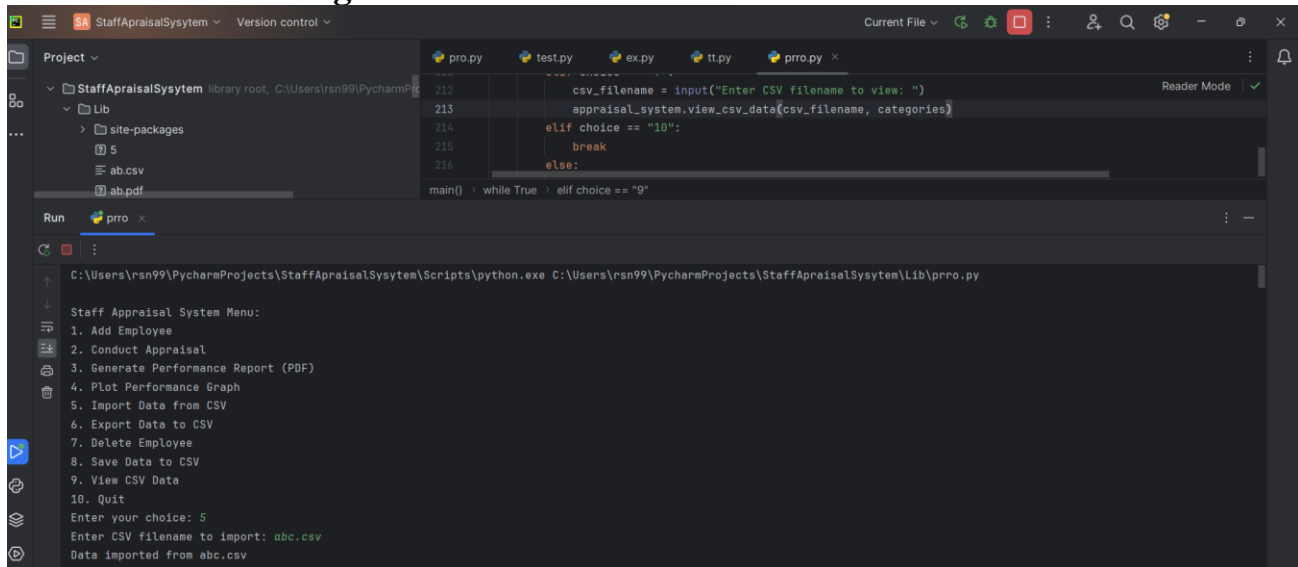
3. **Performance Goals:** It should let managers set clear goals for employees, like what they should achieve in their job.
4. **Feedback:** The system should allow managers and employees to give feedback on how well the employee is doing their job.
5. **Performance Ratings:** It needs to have a way to give scores or ratings to show how well an employee is performing.
6. **Development Plans:** It should help make plans for how employees can get better at their jobs and grow in their careers.
7. **Reminders:** The system can send reminders to managers and employees when it's time to do appraisals, set goals, or check progress.
8. **Reports:** It should be able to generate reports and summaries of employee performance for managers and HR to look at.
9. **Security:** It must keep all employee information safe and private, only accessible to those who need it.
10. **User-Friendly:** It should be easy for both managers and employees to use without needing a lot of training.
11. **Integration:** It should work well with other HR systems and tools that the company uses.
12. **Scalability:** It should be able to handle a growing number of employees as the company gets bigger.

Data Set Used: -

1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
2	Employee	Name	goals	objectives	responsibil	work_qual	productivii	attitude	scores										
3	1000	Yash	4	6	8	1	7	3	48.33333										
4	1001	Prit	0	4	7	5	8	10	56.66667										
5	1002	Meet	9	9	0	8	5	0	51.66667										
6	1001	Drashti	9	1	3	1	5	9	46.66667										
7	1002	Saloni	6	0	10	8	1	4	48.33333										
8	1003	Hinal	0	6	3	0	7	0	26.66667										
9	1002	Jay	9	10	8	4	9	9	81.66667										
10	1003	Darshana	7	3	8	9	1	8	60										
11	1004	Hardik	4	9	5	8	8	7	68.33333										
12	1003	Janvi	9	2	5	0	3	1	33.33333										
13	1004	Ronak	0	3	5	10	3	9	50										
14	1005	Naman	10	5	2	7	6	1	51.66667										
15	1004	Khyati	4	4	9	9	0	4	50										
16	1005	Sikha	4	6	3	0	3	8	40										
17	1006	Minal	7	0	8	6	10	4	58.33333										
18	1005	Milan	5	9	10	9	4	5	70										
19	1006	Kaushik	2	1	3	10	6	6	46.66667										
20	1007	Smit	5	6	5	8	7	0	51.66667										
21	1006	Ravina	4	4	4	2	3	0	28.33333										
22	1007	Priti	8	3	4	5	0	5	41.66667										
23	1008	Hetal	6	3	7	4	1	10	51.66667										
24	1007	Pooja	0	1	10	4	6	10	51.66667										
25	1008	Kishan	0	0	9	6	8	1	40										
26	1009	Akshar	5	9	2	1	5	3	41.66667										
27	1008	Akshay	9	7	0	8	9	5	63.33333										
28	1009	Radhika	9	3	7	3	10	8	66.66667										
29	1010	Riya	1	2	10	0	4	4	35										
30	1009	Komal	6	4	0	3	8	3	40										
31	1010	Mihir	0	1	4	2	1	4	20										
32	1011	kunjai	3	7	6	0	7	2	41.66667										

32	1010 Moxil	9	6	0	8	4	9	60											
33	1011 Mona	1	1	5	5	4	1	28.33333											
34	1012 Bhavya	7	8	6	3	1	3	46.66667											
35	1011 Bhavika	10	8	9	2	3	10	70											
36	1012 Rina	1	10	10	6	3	2	53.33333											
37	1013 Hetavi	7	5	10	0	0	4	43.33333											
38	1012 Manoj	7	0	7	1	3	1	31.66667											
39	1013 Raghav	5	0	1	6	10	4	43.33333											
40	1014 Gopal	2	3	9	3	3	4	40											
41	1013 Jasmin	6	1	3	7	10	7	56.66667											
42	1014 Akshay	3	7	8	2	1	0	35											
43	1015 Sweta	10	8	0	6	2	0	43.33333											
44	1014 Vidhi	7	5	0	7	8	8	58.33333											
45	1015 Nayan	5	7	6	10	10	4	70											
46	1016 Prashant	6	0	4	7	1	7	41.66667											
47	1015 Sonal	10	3	3	5	8	1	50											
48	1016 Bhavika	2	7	4	7	10	5	58.33333											
49	1017 Bhavik	8	2	6	9	8	5	63.33333											
50	1016 Manthan	2	9	10	5	7	0	55											
51	1017 Preet	8	5	7	4	8	4	60											
52	1018 Darpit	8	0	5	8	7	8	60											
53	1017 Harsh	0	3	0	8	2	7	33.33333											
54	1018 Kashyap	3	3	6	9	4	5	50											
55	1019 Bhautik	8	6	9	5	10	9	78.33333											
56	1018 Dhruv	9	6	0	10	0	1	43.33333											
57	1019 Ishva	5	9	1	10	2	7	56.66667											
58	1020 Mitesh	5	10	1	4	9	1	50											
59	1019 Denish	0	1	1	2	8	9	35											
60	1020 Abhi	8	7	5	4	8	1	55											
61	1021 Jenish	3	6	1	4	7	7	46.66667											
62	1020 Jenny	9	0	8	5	0	5	45											

Data Set used in Program



```
pro.py
212         csv_filename = input("Enter CSV filename to view: ")
213         appraisal_system.view_csv_data(csv_filename, categories)
214     elif choice == "10":
215         break
216     else:
217         continue
218 main()
while True:
    elif choice == "9":
```

Run pro

C:\Users\rns99\PycharmProjects\StaffAppraisalSystem\Scripts\python.exe C:\Users\rns99\PycharmProjects\StaffAppraisalSystem\Lib\pro.py

Staff Appraisal System Menu:

1. Add Employee
2. Conduct Appraisal
3. Generate Performance Report (PDF)
4. Plot Performance Graph
5. Import Data from CSV
6. Export Data to CSV
7. Delete Employee
8. Save Data to CSV
9. View CSV Data
10. Quit

Enter your choice: 5

Enter CSV filename to import: abc.csv

Data imported from abc.csv

```
Staff Appraisal System Menu:
1. Add Employee
2. Conduct Appraisal
3. Generate Performance Report (PDF)
4. Plot Performance Graph
5. Import Data from CSV
6. Export Data to CSV
7. Delete Employee
8. Save Data to CSV
9. View CSV Data
10. Quit
Enter your choice: 9
Enter CSV filename to view: abc.csv
Employee ID: 1000, Name: Yash, goals: 4, objectives: 6, responsibilities: 8, work_quality: 1, productivity: 7, attitude: 3

Employee ID: 1001, Name: Prit, goals: 0, objectives: 4, responsibilities: 7, work_quality: 5, productivity: 8, attitude: 10

Employee ID: 1002, Name: Meet, goals: 9, objectives: 9, responsibilities: 0, work_quality: 8, productivity: 5, attitude: 0

Employee ID: 1001, Name: Drashti, goals: 9, objectives: 1, responsibilities: 3, work_quality: 1, productivity: 5, attitude: 9
```

3.2 User Interface Requirements:

User interface (UI) requirements for a staff appraisal system are crucial to ensure that the system is easy to use and provides a positive experience for both managers and employees. Here are some key UI requirements for a staff appraisal system:

1. **Intuitive Design:** The user interface should be easy to understand and navigate, even for users with limited technical expertise. It should use common design conventions and terminology.
2. **User-Friendly Forms:** Forms for inputting performance data, feedback, and goals should be well-organized, with clear labels and instructions. Users should be able to complete them without confusion.
3. **Responsive Design:** The system should work seamlessly on various devices, including desktop computers, tablets, and smartphones. The interface should adapt to different screen sizes and orientations.
4. **Easy Editing:** Users should be able to easily edit and update information, such as goals or feedback, as needed.

3.3 Usability:

The usability of a staff appraisal system is crucial for its effectiveness and acceptance within an organization. When a system is user-friendly and easy to use, it can lead to several benefits:

1. **Improved User Adoption:** When a staff appraisal system is easy to navigate and understand, employees and managers are more likely to embrace it. This leads to higher participation rates and more accurate data in the appraisal process.
2. **Efficient Processes:** Usability helps streamline the appraisal process. Users can quickly input and access information, reducing the time and effort required for performance evaluations.
3. **Reduced Errors:** A user-friendly interface minimizes the likelihood of data entry errors and ensures that evaluations and feedback are recorded accurately.
4. **Enhanced Communication:** Usability fosters effective communication between managers and employees. Users can easily access feedback, set goals, and track progress, promoting better understanding and alignment.
5. **Increased Satisfaction:** Users are more satisfied when they find the system easy to use. This can positively impact morale and engagement within the organization.
6. **Training Efficiency:** A user-friendly system requires less training for users to become proficient, saving time and resources.
7. **Data Quality:** Usability encourages users to regularly update and maintain their performance records, leading to more reliable and up-to-date data.

3.4 Performance of Staff Appraisal System :

The performance of a staff appraisal system can have a significant impact on an organization's overall success and employee satisfaction. A well-designed and effectively implemented system can yield several positive outcomes, while a poorly executed one can lead to various challenges. Here's an overview of how to assess the performance of a staff appraisal system:

Positive Performance Indicators:

1. **Improved Employee Performance:** A successful staff appraisal system should lead to improved individual and team performance. Look for evidence of employees meeting or exceeding their performance goals and expectations.
2. **Increased Employee Engagement:** Engaged employees are more likely to perform well. Gauge whether the appraisal system has contributed to higher levels of employee engagement, job satisfaction, and motivation.
3. **Effective Goal Achievement:** Assess whether employees are successfully achieving the goals set during the appraisal process. Evaluate the alignment of individual goals with the organization's strategic objectives.
4. **Enhanced Communication:** A well-performing appraisal system should facilitate open and constructive communication between employees and managers. Measure the quality and frequency of feedback and discussions.
5. **Talent Development:** Determine whether the system identifies training and development needs accurately. Evaluate the effectiveness of development plans in helping employees grow their skills and competencies.

3.5 Maintainability:

Managing and maintaining a staff appraisal system is essential to ensure its ongoing functionality, relevance, and effectiveness within an organization. Here are key considerations for manageability and maintenance:

1. Regular Updates and Upgrades:

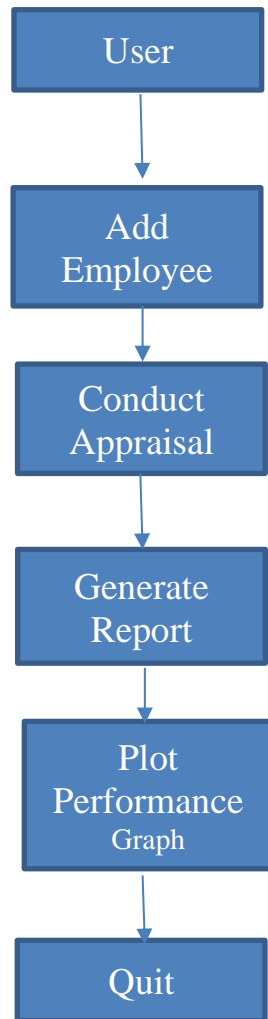
- Keep the system up to date with the latest software patches, security updates, and feature enhancements provided by the vendor or development team.

2. Performance Monitoring:

- Keep an eye on system performance to ensure it can handle increased usage as the organization grows. Address any performance issues promptly.

3. User Feedback and Improvement: - Encourage users to provide feedback on the system's functionality and usability. Use this feedback to make continuous improvements and enhancements.

4.1 DFD



4.2 References

- <https://www.geeksforgeeks.org/>
- <https://www.python.org/doc/>
- <https://www.w3schools.com/python>