

Performance Measurement Solution for University

Purpose:

The objective of this application is to measure the quality of education delivered to students considering performance of students, rating of course, ranking of the university, performance of faculties and job offers acquired by the students. These factors will help the universities to figure out the factors influencing the performance of university and find areas where the universities can improve.

Functionalities:

- Admin has access to the complete application.
- Admins can login into the application by entering the username and password.
- New admins can be created for universities.
- Only admins of the university can access performance dashboard.
- Admins possess recover account option.

Approach:

The application provides feedback and performance measurement for students to track the performance of universities in terms of job trends, course rating and faculty rating. The feedback from alumni collected will help in identifying the rating of the university. The application also provides universities insights about the courses that has high potential and helps them update the content of it.

Alumni provides information regarding their favorite course, recommended course, job domain, graduation year, employment history, job course satisfaction, etc. Using the information provided by the alumni we then rank the courses, faculty, current job trends in the market. This information can then be utilized by the university admin to guide current graduate students, in the right direction of their interest.

Based on the current and historical job market data collected from the alumni we can also rank the job domains and the student's success in the domain. From the courses taken by the alumni, the respective faculty, and the job trend, the student will be able to decide how to adapt to a career path.

Classes and relevant methods:

Department:

- name: string
- courseDirectory: ArrayList<course>
- personDirectory: ArrayList<person>
- facultyDirectory: ArrayList<faculty>
- studentDirectory: ArrayList<student>
- alumniDirectory: ArrayList<alumni>
- employerDirectory: ArrayList<employer>
- addToDepartmentList – add new department to the list
- updateDepartmentDetails – update an existing department detail

Course:

- courseName: string
- courseCode: string
- addCourse – To add new course which gets description, hours, seats and faculty

Course Directory:

- courseList: ArrayList<Course>
- getAllCourses – Gets all the courses available
- updateCourseCatalog – Update the course list with new courses

Person:

- id: string
- status: string
- isMatch – check if the person is valid
- addPerson – add new person to the list

Person Directory:

- personList: ArrayList<person>
- addNewPerson – add new person to the list

Faculty:

- facultyId: string
- facultyName: string
- facultyCourses: ArrayList<facultyCourseList>

Faculty Directory:

- faculty: ArrayList<Faculty>
- addFacultyDetails – Gets name, id, courseList and creates faculty details
- updateFacultyDetails – updates faculty details

Student:

- id: string
- name: string
- studentCourseList: ArrayList<StudentCourse>
- addStudent – adds new student
- updateStudentDetails – updates student details
- getStudentTranscript – gets the transcript of specific student
- getStudentCourses – gets the list of specific student course

StudentCourse:

- studentCourseName: string
- studentCourseId: string
- studentCourseSemester: string
- studentCourseFaculty: string
- addStudentCourse: gets id, name, semester, faculty and adds the student course to the list
- updateStudentCourseDetails: updates the student course detail

StudentTranscript:

transcript: id

getStudentCgpa – gets the student cgpa

getAllTranscriptDetails – gets the student transcript

updateStudentTranscript – updates the student transcript

StudentDirectory:

studentList: ArrayList<student>

addStudentDetails – adds the new student details

updateStudentDetails – updates the student details

Alumni:

- alumniId: String
- alumniName: String
- alumniGraduationYear: int
- alumniCourses: ArrayList<alumniCourseList>
- alumniEmploymentHistory: ArrayList<alumniEmploymentHistory>
- getAllAlumni – gets the list of all the alumni
- addAlumni – adds new alumni to the list

AlumniCourseList:

- alumniCourseName: string
- alumniCourseSemester: string
- alumniCourseGrade: string
- alumniFacultyName: String
- alumniCourseAdd – adds new alumni course
- alumniCourseUpdate – updates course of alumni

AlumniEmploymentHistory:

- numberOfPromotions: int
- companyNames: ArrayList<string>
- jobSatisfaction: int
- alumniJobDomain: string
- addAlumniEmployment – gets the company name and adds the alumni employment
- updateAlumniEmployment – gets the company name, promotion, domain, satisfaction and updates the employment of the alumni
- findJobSatisfactionScore – gets the job satisfaction score

AlumniDirectory:

alumni: ArrayList<alumni>

getAlumniProfile: gets the profile of alumni

updateAlumniProfile: updates the profile of alumni

Feedback:

- alumniFavoriteCourse: ArrayList<string>
- alumniRecommendedCourse: ArrayList<string>
- alumniCourseRanking: ArrayList<string>
- addAlumniFeedback – adds the feedback of alumni with respect to courses
- updateAlumniFeedback – updates the feedback of alumni
- findAlumniFacultyRating – gets the rating of faculty from alumni
- courseToJobSatisfactionScore – gets the score of job satisfaction with respect to the course

UniversityAdmin:

- adminUserName: String
- adminPassword: String
- adminVerify – gets the username and password and verifies the user
- evaluator – performs evaluation of course, respective faculties, and relevance of the course in that particular job domain as per the Alumni feedback.

Evaluator:

- courseRating: int
- facultyRating: int
- courseIndustryRelevance: int
- evaluateCourse: evaluates the course with job
- evaluateFaculty: evaluates the faculty
- overallRating: computes the overall rating with course, faculty and job

Faculty:

- facultyName: string
- faculty: int
- facultyCourses: ArrayList<facultyCourseList>
- addFacultyCourse – gets id, name, course and adds new course under faculty
- updateFacultyCourse – updates the faculty course

FacultyDirectory:

- Faculty: ArrayList<faculty>
- addFacultyDetails: gets id, name, courselist and creates faculty details
- updateFacultyDetails: updates faculty details

Teams Members:

1. Anirudh Voruganti
2. Muralikrishna Shanmuganathan
3. Yashwanth Sridharan