

# Web-based System for Assessment and Continuous Improvement

Md Shabbir Hassan  
Taif University

Arif M. Bhatti  
Taif University

## ABSTRACT

Academic programs are required to monitor, evaluate and improve in order to acquire or maintain program accreditation from standards bodies. Data collection, management, statistical analysis and result aggregation are important tasks to achieve this goal. A manual approach may result in inconsistent, redundant and poor quality data that can be avoided using web-based technologies. This paper presents design and implementation of a web-based system that collects and maintains course related information, computes metrics for assessment and continuous improvement, and generates reports that can be manipulated off-line.

## Keywords

Web-based system, accreditation, assessment, continuous improvement, Django Framework, python.

## 1. INTRODUCTION

Program assessment and continuous improvement are important components in accreditation of any academic program. Mission statement, program objectives, student outcomes and course outcomes are defined and evaluated to measure the performance of a program [5].

System presented in this paper is a web application designed to meet the assessment needs of ABET [1] and NCAAA [2] accreditation. This system facilitates the implementation of assessment and continuous improvement process described in [3] and its evaluation engine computes metrics defined in [4] for quantitative analysis.

Program evaluation requires data collection, quantitative analysis and performance reports for further actions based on the findings [3, 5, 6, 7]. Significant effort is needed for management of collected data to ensure that required data is available when it is needed. Web technologies can be used to manage all aspects of assessment to ensure that right data is available to the right person. Amount and type of data collected, tools and approaches used for quantitative analysis depends on the implemented assessment process [3, 5, 7].

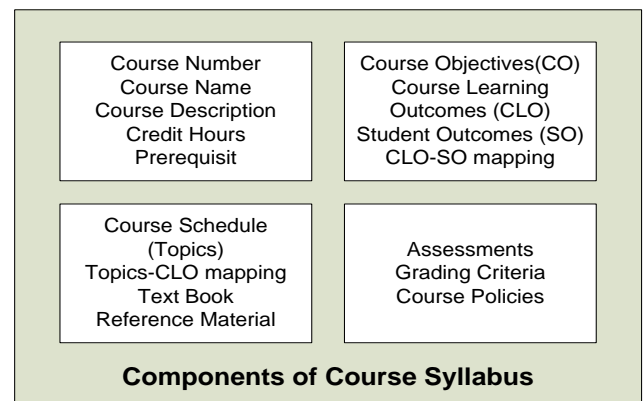
Study plan of an academic program enumerate core as well as elective courses that a student must complete to graduate from the program. Program objectives are required to evaluate the program performance. Student objectives (SO) can be defined by the program or adopted from some accreditation body such as ABET [1, 6].

Student outcomes are skills that a graduate must have by the time of graduation. These skills are related to the competence, self learning, understanding of social impacts and ability to apply knowledge gained in the area of studies to the real life problems.

Students acquire SO skills while studying required courses and each course focus on some skills that contributes to the achievement of student outcomes. Course learning outcomes (CLO) are micro level skills that students acquire from

studying a course. It is expected that if a student successfully completes a course then he should have the skills listed in the course CLOs. These skills contribute to the program level skills defined by SOs.

Course syllabus is a document that describes the course contents, course learning outcomes and related student outcomes. Figure 1 shows components of a course syllabus that includes catalog information, assessment outcomes, topics, policies, relationship between topics and CLOs, and relationship between CLOs and SOs. Course designer should be careful in defining these relationships as they are used in quantitative analysis for assessment.



**Fig 1: Course information that includes course description, contents, policies, objective and outcomes used for assessment and continuous improvement.**

A web based system used for ABET accreditation is presented in [5]. This system manages course objectives for courses used for accreditation. Course instructors enter their evaluation for each of the objectives using an interface. Students also submit their evaluation of objectives. Statistical analysis is performed and presented as a web page. Data collection is limited to the input from instructors and students, which is very subjective. Approach adopted in this paper collects question-wise data for all assessments conducted in addition to the instructors' and students' evaluation of the course.

Visualization of quantitative analysis performed on collected data helps in finding out meaningful information. In [6] dashboard approach is used for visualization of assessment results. It is not clear how these dashboards are created. In this paper, a similar approach is adopted and system generates graphs using Excel spreadsheet functionality. A system for ABET accreditation is presented in [7]. A portal approach is used to present objectives, assessment and course material in a meaningful way to facilitate assessment and continuous improvement process.

Our system is very comprehensive, it collects data for all assessments used in the evaluation of a course, performs quantitative analysis at course and program levels.

Visualization and easy access to all material required for assessment are important features of the system. Section 2 presents requirements and system architecture. Section 3 discusses the implementation details and structure of data repository. Section 4 briefly discusses the results and last section presents the conclusion.

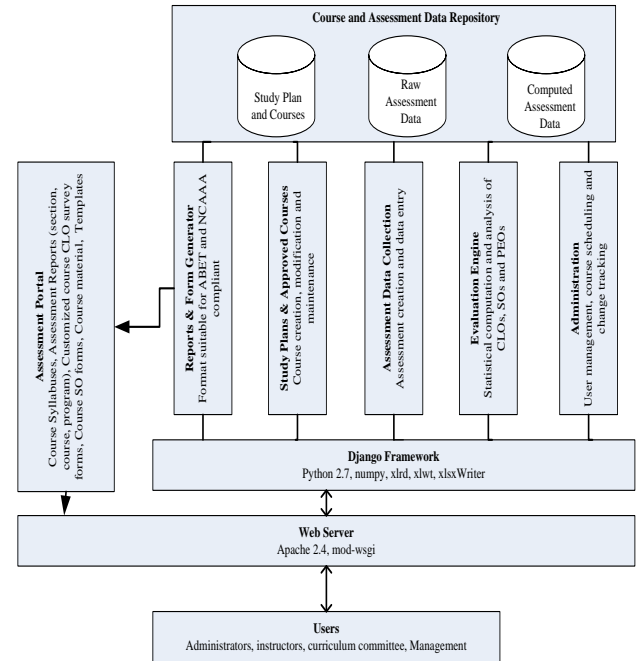
## 2. SYSTEM ARCHITECTURE

Advances in web and cloud computing technologies made it possible to have your data at a centralized place and access it anytime, anywhere and from any device. Excel spreadsheet is one of most commonly used tool by the academic community to manage course data, perform statistical analysis and visualize the data using graphs. To ensure that not everyone has to be skilled in spreadsheet features, templates can be designed and made available for consistency in presentation.

Prior to this system, Excel-based template approach was used but quality of data was poor, inconsistent and incomplete for different sections of the same course that does not meet the requirements of assessment process. Decentralized nature of the data also made it difficult to create course-level assessment for multi-section courses and program level assessment that require aggregation of data for all courses. Easy access to all relevant data for assessment was also a challenging task. To deal with these issues, a web-based system was envisioned that meet the following requirements.

1. Centralized course specifications to avoid redundancy, inconsistency and duplication to have control on quality of course contents.
2. Centralized management of outcomes both at course and program level. Management of relationships among CLOs, SOs, topics and assessment questions for each course. This information is crucial for assessment and quantitative analysis.
3. Data collection for all assessments conducted in a course and performs quantitative analysis for continuous improvement.
4. Ability to compare different sections of the same course and compare courses offered in different semesters using pre-defined metrics
5. System generated documents for course syllabus, and course survey for expected outcomes in Excel format.
6. System generated quantitative analysis reports with raw as well as processed data suitable for visual presentation using Excel features such as formulas and graphs.
7. Generate documents and reports suitable for course instructor, course coordinators, curriculum committee, and management to be used for continuous improvement at different levels in the program.

Component based approach is used in system design and architecture. Figure 2 shows components of the system where each component deals with one or more of the requirements listed above. Rest of the section explains the technologies, user interfaces and portal functionality of the system.



**Fig 2: System architecture showing functional components, data repositories and portal for information sharing. Technologies used to implement this system are also listed.**

### 2.1 Technologies

This system is developed using Django [9], an open source web framework written in python, which is suitable for rapid application development. Python is one of the most popular languages that comes with a huge number of libraries and allows writing concise and compact programs. Python libraries such as numpy and xlsxWriter [10] are used for statistical analysis and Excel formatted report generation. MySQL database is used to store raw as well as processed data.

### 2.2 User Interfaces

In traditional web systems, forms are used for accepting user input and system generated web pages are used for responses. In this system forms are used for accepting user input but all responses except trivial ones are generated as Excel formatted files. These files can be saved, edited for inserting additional information or can be manipulated to perform additional quantitative analysis. Rest of the section explains use and functionality of system interfaces.

#### 2.2.1 System management and scheduling

Major tasks performed through this interface are user management and course management. Each semester new courses are offered and each course usually has multiple sections. Setting up the system to collect data for the semester requires course scheduling information to be inserted. This activity may require creation of new users or new role assignment.

A role based scheme is used for authorization where a user is assigned one or more roles. Depending on the assigned role a user may see different menus. New user creation and role assignment is done through this interface.

### **2.2.2 Study plan management and course specifications**

Study plan is the set of courses offered by a program for students to successfully complete before graduating from the program. Figure 1 shows course information that must be made available in the system for each course in the study plan.

This interface allows creation and update of program study plans and courses in the plan. Only authorized users can access this interface and role-based user authorization is managed by previously defined interface. This interface is directly related to the requirement 1 and 5 listed at the start of this section.

### **2.2.3 Course-level assessment constraints management**

Program assessment approach adopted in this system is bottom-up. Skills acquired in course CLOs directly contribute in the SO skills acquired at program level. As shown in figure 1, for each course mapping between course topics and CLOs, and mapping between CLOs and SOs is required to perform quantitative analysis. This interface allows authorized users to insert and update required outcome mappings for each course. This interface is related to the requirement 2 as mentioned before.

### **2.2.4 Course assessment creation and data submission**

Course instructors conduct multiple assessments in the form of home works, quizzes, projects, midterm and final exams. At the start of each semester course instructors are assigned an instance of a course in the system for each course that instructor is teaching.

An instructor sees all assigned course instances that he is required to submit course assessment data and teaching material. Using this interface an instructor can create assessments and submit students' question-wise grades. After creating an assessment, instructor can use an interface similar to the one shown in figure 4 for data entry. Number of columns and rows in the interface depends on the number of questions in the assessment and number of students in the course. This information is provided at the assessment creation time.

There is no limit on the number of assessments that an instructor can create for one course. Minimum number and type of assessments required is a policy issue for the program.

## **2.3 Information Portal**

This is a key component of the system to provide seamless access to all information and documents needed by the course instructors and accreditation teams.

### **2.3.1 Course Syllabus**

These documents are generated from the system in Excel format and its contents are shown in figure 1. Instructors use these documents for their courses and accreditation reviewers are able to review course contents, CLOs and relevant SOs. A course instructor can edit or update a syllabus for his course but an official change can only be done through the interface by authorize user only.

### **2.3.2 Course Survey Forms**

Course surveys are conducted to include students' perception of what they have learned in the course. For each section of a course, a survey form is generated that asks user input on CLOs of the course. This form is made available on the portal.

### **2.3.3 System Generated Reports (CAF, Course, Program)**

Evaluation engine of the system performs quantitative analysis of collected data and generates reports for different audience. These reports are made available on the portal.

Course Assessment File (CAF) is meant for course instructors. This is a multi sheet Excel file that has course syllabus, data for each assessment as instructor has entered in the system, grade distribution and quantitative analysis of the course data.

Course File (CF) is designed for multi-section courses to compare the students' performance in different sections of the same course. It is useful for finding out issue in teaching style, grading policy and collaboration among course instructors.

### **2.3.4 Digitized course material**

This section of the portal is for new instructors, course coordinators and accreditation reviewers. At the end of each semester, instructors submit their course material, samples of student work and evidence of SOs covered in the course. All submitted material is digitized and made available at the portal.

## **3. IMPLEMENTATION**

There are five major components as shown in figure 2. Functionality of these components and data repositories used to implement the functionality are explained in this section.

### **3.1 Study plans and Approved Courses**

This system is built on the concept of approved courses where each course has gone through a process to ensure that all assessment related information such as CLOs, SOs, topic-CLO mapping, and CLO-SO mapping are appropriate and suitable for program evaluation. This information changes very infrequently and only an authorized user can change this.

#### **3.1.1 Course Specifications**

Data in this category includes course description in catalog, text book, weekly teaching schedule, course policies and course outcomes. Significant part of this data is static and a small part of it changes infrequently. Following data tables are used to capture course information in the database. Contents of each table can be guessed from the table name.

- CourseInfo
- CourseObjectives
- BookInfo
- CourseBook
- CourseCLOs
- CourseTopics
- CoursePolicy

#### **3.1.2 Course Outcome Specification**

Assessment of student outcomes (SO) at program level is derived from assessment of CLOs and SOs at course level. Data in this category is required for course assessment. This information is used by the evaluation engine to perform

quantitative analysis. Figure 3 shows how information in the following tables is used for computation of CLO metrics and derivation of SO metrics from the computed CLOs. This information is also very infrequently updated by authorized users.

- CourseCLOs: This table is used for storing course CLOs, which is used in automated generation of reports, course syllabuses and course survey forms.
- CourseTopics: Weekly schedule is stored in this table where each topic is also mapped to the relevant CLOs.
- SO\_CLO\_Mappings: Mappings between CLOs and SOs is stored in this table. Performance of CLOs of a course is directly computed from students' performance in the course assessments. Course SOs are derived from corresponding CLOs [4].

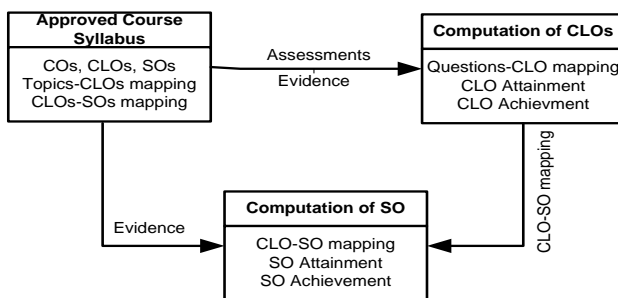


Fig 3: Computation of metrics (attainment and achievement) for CLOs and SOs of a course listed in the course syllabus.

### 3.2 Administration and Scheduling Information

Life of information in this category is one semester. New courses possibly with multiple sections are offered each semester. For each course section, a course instance is created and assigned to the course instructor. OfferedCourses table is used to capture course instructor, specification, semester, and section information.

- OfferedCourses
- Exams\_Assessments

For each course, instructor creates multiple assessments such as home work, midterm and final exams. Information on number of students in the course and number of questions in an assessment is required at assessment creation time. There is one entry for each assessment created in the Exams\_Assessments table that holds information about all assessments created by all instructors for all courses.

### 3.3 Assessment Data Collection

This module provides an interface and instructors are required to insert student's question-wise grades for each assessment conducted in their course. Information captured by this module is for the current semester only. As mentioned previously, instructors are required to create course assessments in the system by providing number of students and number of question information and this information is stored in Exams\_Assessments table.

Before insertion of grades for an assessment, instructor selects one assessment from the assessments that he has created previously. A table, as shown in figure 4, shows up for data entry. Dimension of the table depends on the information

provided at assessment creation time. Before inserting students' grade, instructor is required to enter maximum marks for each question and CLOs covered by the question. Topic-CLOs mapping in the syllabus should help instructor in selecting the right CLOs. All students' grades and mapping information is stored in EnterAssessmentMarks table. This table is not normalized to avoid excessive joins, as all this information is used by the evaluation engine once for quantitative analysis.

CLOs	2	2	4	1	3	1	2	4
Question	1	2	3	4	5	6	7	8
Question Marks	3	3	3	3	3	3	3	3
Student SNO.								
1								
2								
3	2	2	1	1	2	1	1	1
4	2.5	3	2	2	2	1.5	3	1
5	0	2	0.5	2	2.5	0	1	0
6	3	3	2	3	3	3	1	1
7	3	3	1	1.5	3	2	1	3
8	0.5	2	1	3	1	1	1	1
9	2.5	2	1	2.5	2.5	1	1	1
10	2.5	1	0.5	0.5	1.5	0	0	0

Fig 4: Data collection interface for a course assessment showing mapping between assessment questions, maximum marks and CLO for each question.

Final grades for students are not computed from the assessment data entered by the course instructor. This design choice is adopted to give instructors flexibility in using fixed grading scale, grade curving or grade scaling. Instructor is required to insert students' final grade to visualize grade distribution in the course. This information is stored in FinalGrade table.

Course surveys are done at the end of each course and students' are required to give their input on the scale of 1-5 on how well each of the course CLO is covered. Survey forms are generated by the system from course specification. Course instructors are required to use this system generated survey forms. Instructor inserts the survey responses that are stored in CourseCLOSurvey table.

### 3.4 Evaluation Engine

This module is a major component of the system as it automates the quantitative analysis of collected data that is used in system generated reports discussed in the next section. Details of the performance metric computed by this system are discussed in [4].

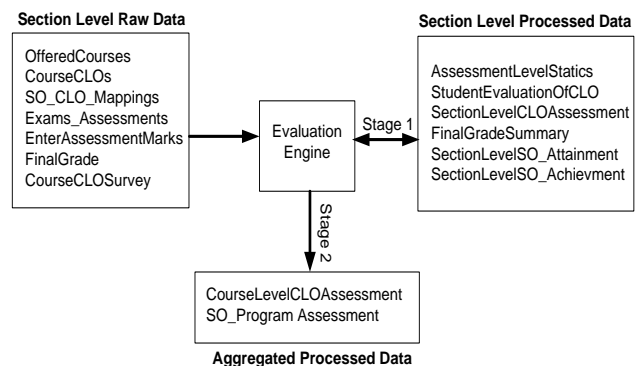


Fig 5: In first stage, evaluation engine perform quantitative analysis of raw data for each course section and store as section-wise processed data. Second stage aggregates the section-wise process data to generate course-level and program level data.

Two metrics, attainment and achievement, are frequently mentioned in this section. Attainment for an assessment is based on students' average grade in the assessment. Achievement in an assessment is the number of students who are above a threshold. Default threshold is set to minimum of 70% or average total grade. Both of these metrics are defined in terms of percentage instead of absolute numbers.

This system assumes that each course has multiple sections in each semester. Evaluation engine runs in two stages as shown in figure 5. In the first stage, assessment data of a course is processed and stored in new data tables that will be used to compare with previous and future offerings of the same course. In the second stage, processed data is aggregated for courses with multi-sections and for computing program level attainment and achievement of SOs.

### 3.4.1 Data processing at section level (stage 1)

Processing starts with calculation of average grade and number of students above threshold for each and every assessment, for which data is inserted in the system by course instructors. These computed values are stored in AssessmentLevelStatistics table. This table is retained across semesters and is used in further processing described below.

Assessment level statistics for all assessments conducted in a course section are aggregated to compute CLO attainment and achievement for the section and results are stored in SectionLevelCLOAssessment table.

CLO-SO mapping defined in the course specification is used to derive SO attainment and achievement values from assessment level statistics and section level CLO performance. These computed values are stored in SectionLevelSO\_Attainment and SectionLevelSO\_Achievement tables.

Student survey results are stored in StudentEvaluationOfCLO and grade distribution for course section is stored in FinalGradeSummary tables.

### 3.4.2 Course and Program level data processing (stage 2)

For multi-section courses it is required to compare results of performance metrics and grade distribution among different sections. This activity requires aggregation of section level results. CLO attainment and achievement of different sections of a course is aggregated and stored in CourseLevelCLOAssessment table. CLO-SO mapping in the course specification is used to derive SO attainment and achievement, which is stored in SO\_ProgramAssessment. After this step, all raw data submitted by the course instructors is processed and can be used in different types of system generated reports.

## 3.5 Report Generation

Web based interfaces are used for system management, traversing menu options, assessment creation and data insertion for created assessments. For data retrieval from the system, instead of web interfaces, Excel formatted documents are used for following reasons.

- It is easier to browse through a well organized document online as well as offline, instead of browsing a web page and clicking through links online only.
- Spreadsheet documents are frequently used by academic community to manage course related data.

Academic staff is familiar with Excel formulas, macros and graphs for data analysis and visualization.

Three major group of reports are generated by the system as shown in figure 6. In academic environment, each faculty members play one or more roles such as instructor, coordination, committee member and program chair. Generated reports are meant for different roles as shown by 'relevance' column in figure 6.

	Relevance	Course Specifications (Detail)	Course Summary	Outcomes (CLOs, SOs)	Raw Data (Assessments, Survey, Processed Data (Tables, Graphs))	Grade Distribution
Course Assessment File (CAF)	Section, Instructor	X	X	X	X	X
Course Report	Course, Instructors, Coordinator	X	X	X	Comparison	Comparison
Program Report	Program, Curriculum Committee and ...			Sos Only		X

**Fig 6: System generates multiple reports for different audience with different components. Program report presents aggregated course analysis of whole program semester wise and for the whole year.**

### 3.5.1 Course assessment file (CAF) is a course-section level

Document and it presents all information required to review a course-section. This document is a multi-sheet Excel formatted file that has sheets for complete course syllabus, data as it was inserted by the course instructor for all assessments conducted in the course, student survey results, final grades and grade distribution graph, and quantitative analysis for performance metrics in numbers as well as graphs for visual interpretation.

Course report (CR) is meant for course coordinators of multi-section courses. For single section courses this document is a summary of CAF document. This report serves following objectives.

- Presents course level performance aggregated from performance of different sections of the course.
- Presents comparison of different sections of a course in both tabular as well as in graphs for visual analysis and interpretation.
- Analyze grading policies, coordination, CLO coverage and students' perception in different sections of a course taught by different instructors.

Program report (PR) is meant for program chair and committees involved in program assessment and accreditation. This report provides a summary of SO attainment and SO achievement for each course during the current semester and during the current academic year. This report provides comprehensive analysis of the whole study plan of the program.



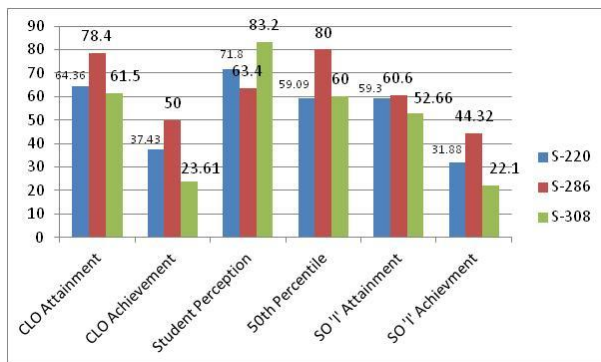
Excel Formula	CAF	Course Report	Program Level Report
SUM()	√		
IF()	√	√	√
ROUND()	√	√	√
ISERR()	√	√	√
AVERAGEIF()	√	√	√
Excel Single Bar Chart	√	√	√
Excel Multiple Bar Chart		√	
Excel formatting features	√	√	√

**Fig 7: Excel features used in reports generated by the system for simple statistics and graph generation from processed data.**

Evaluation engine uses python language scripts using numpy module to compute performance metrics. For presentation of this data, Excel features as shown in figure 7 are used for data pruning and visualization.

#### 4. RESULTS

This system has been in use for the last three years as a single source for all academic activities in the college of computer and information technology. Faculty members use system generated course syllabuses and any change in a course syllabus is done through a process [3]. This helps to avoid inconsistency among different sections of a course. Figure 8 shows a system generated graph that compare performance of three sections of a course for a single CLO.



**Fig 8: Comparison of three sections of the same course showing student performance for a single CLO. Metrics computed for each section of a course are shown.**

Samples of user interface, auto generated documents and quantitative analyses are available in [8].

#### 5. CONCLUSION

Traditional approaches to data collection, analysis and aggregation use Excel or Word templates to be used by the faculty members. In our experience this approach leads to incorrect, inconsistent and duplicate data. Data aggregation

for program level reports need human effort that may result in errors.

System presented in this paper was designed to deal with the above mentioned problems. In three years of use, system has shown that it can decrease the work needed to be done by faculty members. Automated analysis and report generation make the information readily available on demand basis. This system is a single point of information for course instructors, academic managers and accreditation reviewers.

#### 6. REFERENCES

- [1] ABET, "ABET: Accreditation Board for Engineering and Technology.," *ABET Vision and Mission*, 2014. [Online]. Available: [www.abet.org](http://www.abet.org). [Accessed: 23-Aug-2015].
- [2] "National Commission for Academic Accreditation & Assessment," *البوابة الإلكترونية لوزارة التعليم العالي*. [Online]. Available: [https://pbs.twimg.com/profile\\_images/882666205/mohe\\_new\\_logo\\_400x400.png](https://pbs.twimg.com/profile_images/882666205/mohe_new_logo_400x400.png). [Accessed: 23-Aug-2015].
- [3] A. Bhatti, and I. Ahmed, "Academic Diversity and Assessment Process for CS Program Accreditation," *Creative Education*, **6**, 773-784, 2015. doi: 10.4236/ce.2015.68080
- [4] I. Ahmed and A. Bhatti, "Design and Implementation of Performance Metrics for Evaluation of Assessments Data," <https://arxiv.org/submit/1333271>, 2015.
- [5] S. Poger and F. Bailie, "A Web-Based System for Course and Program Assessment: Prototype to Product", Association for the Advancement of Computing in Education (AACE) Proceeding of the Society for Information Technology and Teacher Education (SITE) Annual Conference, March 2006, Orlando, FL.
- [6] Courtney Lamar, Velma Latson, Quincy Brown, Lethia Jackson, Gail Fink, "Using a Dashboard as a Visualization Tool for Assessment Data ", *The 2013 International Conference on Frontiers in Education: Computer Science and Computer Engineering*
- [7] Raina Petrova, Abhilasha Tibrewal, Tarek M. Sobh, "An Electronic Web-based Assessment System", *Journal of STEM education*, volume 7, Issue 3 & 4, July-december 2006
- [8] A. Bhatti, "AIS Sample Results", 2015. Retrieved from <https://db.tt/MNfNLcrN>
- [9] Django Web Framework, 2016, Retrieved from <https://www.djangoproject.com/>
- [10] XlsxWriter: Creating Excel files with Python, 2016, Retrieved from <http://xlsxwriter.readthedocs.org/>