Report By:
Aditya Tripathi, Aman Sharma, Chandra Bhatt, Siva Reddy, Sri Sai Lekha

"Student Academic Portal": Project implementing internet security features:

Confidentiality | Integrity | Authentication

Abstract

A grading system in education is a system that is used to assess the academic progress of a student which is entirely based upon points scored. Grades are standardized measurements of varying levels of comprehension within a subject area. An online academic portal, is used by most of the educational and commercial training institutes these days. This online portal allows a centralised access to all the related entities to the education system i.e. instructors, students, etc. In this project we have created a similar web-portal which is a centralised access system for students and instructors in which they can view, add student details and related information.

Through this access system we have implemented three security features namely Confidentiality, Integrity and Authentication. Confidentiality has been exhibited by the security feature which only enables a specific student to view his own grades and courses enrolled. Unauthorized attempt to view the grades and information of a particular student results into no results. While, we also have implemented Authentication through a login and password sign in operation by which only genuine users with correct credentials could login to the system. Lastly, we have implemented Integrity through the feature which disables instructors to modify the grades awarded by other instructor for a unique student course combination. Grades are generally assigned in alphabets such as A-F which corresponds to the range of marks obtained.

The web portal includes several pages and views namely the Login home page, student-faculty selection, student login page showing student and courses enrolled, while the instructor page including details such as courses taught and students enrolled in a course. All these views of the portal have been implemented using Html and CSS web designing for the front end while Python as the main programming language in the back end framework which calls several html views as and when desired. We have used JetBrains PyCharm IDE for all the programming languages used. While, for incorporating the database management in the backend SQLite3 has been used to create and modify student database. Also to provide encryption we are using AES alogorithm, which prevents any data to entered only in encrypted format in the database which makes this portal even more secure.

Previous works in this field of web-portals include systems which have a robust security framework including end to end encryption which are deployed on commercial level. We have worked upon a fresh code to exhibit these three security features. Therefore, in the current stage our code base successfully exhibits the security feature but is yet not robust enough to block all the potential threats of malicious unauthorized access attempts. The grades are dynamically fetched from the database which only allows the specific student to view his own grades. Same concept applies to the instructor page in which grades of the course not taken by a particular instructor cannot be modified by another instructor or student.

Report By:
Aditya Tripathi, Aman Sharma, Chandra Bhatt, Siva Reddy, Sri Sai Lekha

Introduction

In this web application for student portal we have created a centralised access system for the faculty members and the students. The home page prompts the user to select whether he/she is a student or an instructor and accordingly redirects to the student or instructor page for further taking the credentials and signing in to respective profiles with the data fetched from the corresponding table of the database.

Such a dynamic web-application which fetches data from real time databases is very important in any education system since it enables its users, many easy to use features such as online access and modification of information anytime. This portal incorporates essential security features such as Confidentiality, Integrity, and Authenticity. These features are further implemented in this web-application as below:

Confidentiality: The basic definition of Confidentiality says that it is a set of rules or a promise that limits access or places restrictions on certain types of information. Measures undertaken to ensure confidentiality are designed to prevent personal information from reaching the wrong people, while making sure that the right people can in fact get it. In our project one student can only access his own grades and information while is denied access to other students' information. Similarly, any instructor can only update the grades for students and courses which he is offering.

Integrity: Integrity involves maintaining the consistency, accuracy, and trustworthiness of data over its entire life cycle. Data must not be changed in transit, and steps must be taken to ensure that data cannot be altered by unauthorized people(for example, in a breach of confidentiality). We are implementing integrity in our project by making sure that students are seeing only those grades which are entered by professor. For this feature, we have made sure that database is not accessed by any person other than instructor who is supposed to update the grades.

Authentication: Authentication is the process of identifying an individual, usually based on a username and password. If any of the username of password is incorrect he/she should not be allowed access inside the portal. Here in our project we have implemented the same. We query the secured SQLite database at the backend using Python- Flask to check whether the credentials entered by user exists in the database and are correct. If the details entered are correct, user can access Student or Faculty module depending on the accesses allowed to them. But if the details do not exist in the database then user is NOT allowed to log in and prompted with a message that the details entered are incorrect.

Problem Description and its importance:

Given the availability of sophisticated technology and seemingly limited investment required, potential attackers can very easily hack the applications or data that we might think secure.

This leads to very serious concern about privacy in general. This loss of data not only reveals information about any user, but can lead to hazardous effects to lives.

- Representative example:

For a moment say any attacker can get into a bank database and can transfer all the money from one account into some other account, or if any terrorist hacks military database and gets all the secret information leading to compromise of nation's security. Thus, it is very important to make system secure enough to avoid such attacked.

- Brief description of related works:

As already stated earlier there are many ongoing researches as well as many existing applications which involve CIA features of internet security. We have included one research paper in references. Along with it some other related applications are like Banking applications, Emailing applications like gmail, Chatting applications like Whats app etc. The usage of such applications provide convenience to their owners, but they also reveal information like individuals' habits, interests, activities, and relationships, their location, their personal or corporate secrets…which user do not want! Hence, they are bound to include CIA features.

- What is missing?

The applications which are developed for high end users like military, banks already have robust implementation. But the applications for university are mostly neglected. There are cases where all the university data has been compromised and either the exam papers, student grades, or health information is revealed to public. Thus, our main aim is to develop such an application for the university database which makes it more secure and avoid any such attacks.

- Your work addressing the research gap:

Most of the internet security applications that we come across are developed using PHP and HTML. Since there are many resources and tools available for these languages, posted to the internet and downloadable for anyone to use malicious intent, regardless of intentions. So we decided to implement proposed Student Academic Portal using Python with Flask framework which is more secure. Also there are very limited resources available for this framework and it is not easy to crack the application using those resources. Along with this for front end we have used HTML 5 which is the latest version of HTML. It has new parsing rules and is more oriented towards flexible parsing and compatibility.

Related Works and their Limitations

There are several applications which incorporate features of internet security in themselves. However, such applications fail to address the needs to low end users and small organizations. Also, these application development is more robust for sectors like banking, military etc. But according to us equal privacy is required for applications that work for

universities as well. But the applications that most of the universities are using either use broadcasting data or have very high end applications which are very costly or requires lot of manual efforts to achieve privacy. Hence, we focused on developing an application from scratch which cater to the needs of university as well as student and achieve very good privacy.

Description of project

- Technicalities:

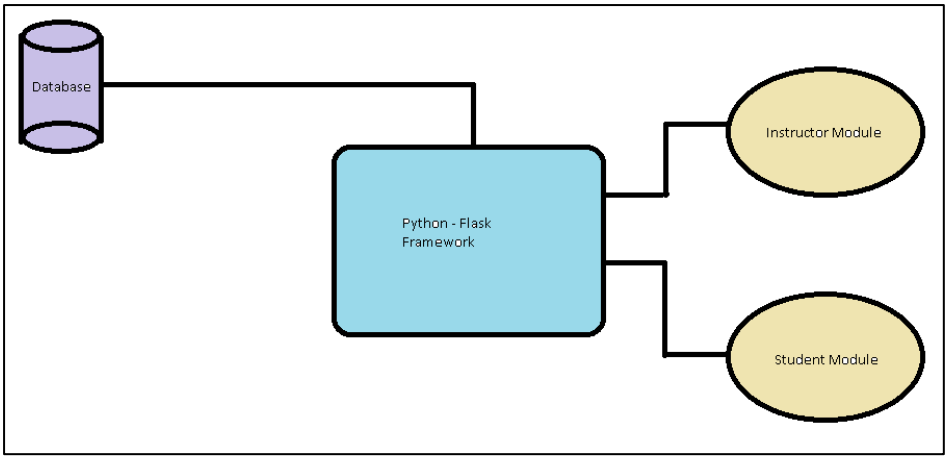
Technical Implementation: We have created a web application that is currently available on local host, port 5000. This application is developed using Python programming language with Flask Framework. We used Python since it is one of the most robust Object Oriented Languages and very simple to use. To create our Database we are using SQLite 3, which is an open source Database. SQLite 3 can be interfaced to Python programs by establishing a connection and creating a cursor for this connection. This cursor is further utilized to execute all SQL queries. SQLite is also present as an add on in Mozilla Firefox and this add on can be directly used to manipulate data directly. Static User Interface is developed using HTML5 and CSS is used for styling. We used JetBrains PyCharm IDE, since it allows us to create and modify all required different type of files at one place. Encryption has been included using AES algorithm which encrypts the sensitive information like username, password, and grades of user before saving them in the database. Also it avoids any transmission from DB to portal which has actual records. The data is decrypted by python program at the back end and then displayed on the portal.

Programming Language: Python
Framework: Flask
IDE: JetBrains Pycharm
Front End or Web Development: HTML5, CSS(styling)
Database: SQLite 3
Encryption Algorithm: AES (Advanced Encryption Standard).

We will divide the description of project into modules and Database description to make it easy to understand the design.

- Block Diagram:

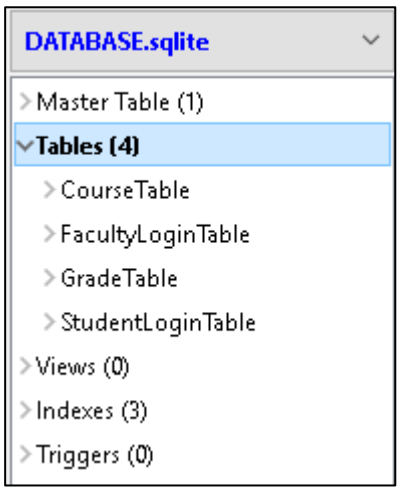
Report By:
Aditya Tripathi, Aman Sharma, Chandra Bhatt, Siva Reddy, Sri Sai Lekha



Database:

We are using SQLite 3 open source Database, which is accessible using an add on- "SQLite Manager" in Mozilla Firefox internet browser. We have created following four tables(with column names mentioned along with their importance and data types) which sums up our entire database. We have tried using as few as tables as possible because it is easier to prevent data loss when you have less data to prevent from attacks. Also, the Database is completely secured because if any attacker gets access to the database then also he will not be able to read the grades and password or username from the data because everything is encrypted using AES algorithm.

1. StudentLoginTable (StudentName text,SID real PRIMARY KEY,UserName text,Password text, Email text)
2. FacultyLoginTable(FacultyName text,FID real PRIMARY KEY,UserName text,Password text, Email text)
3. GradeTable (SID real references StudentLoginTable,FID real references FacultyLoginTable,CourseId real references CourseTable,MidTerm1 double,MidTerm2 double,MidTerm3 double,FinalGrade real)
4. CourseTable (CourseId real PRIMARY KEY,CourseName text)



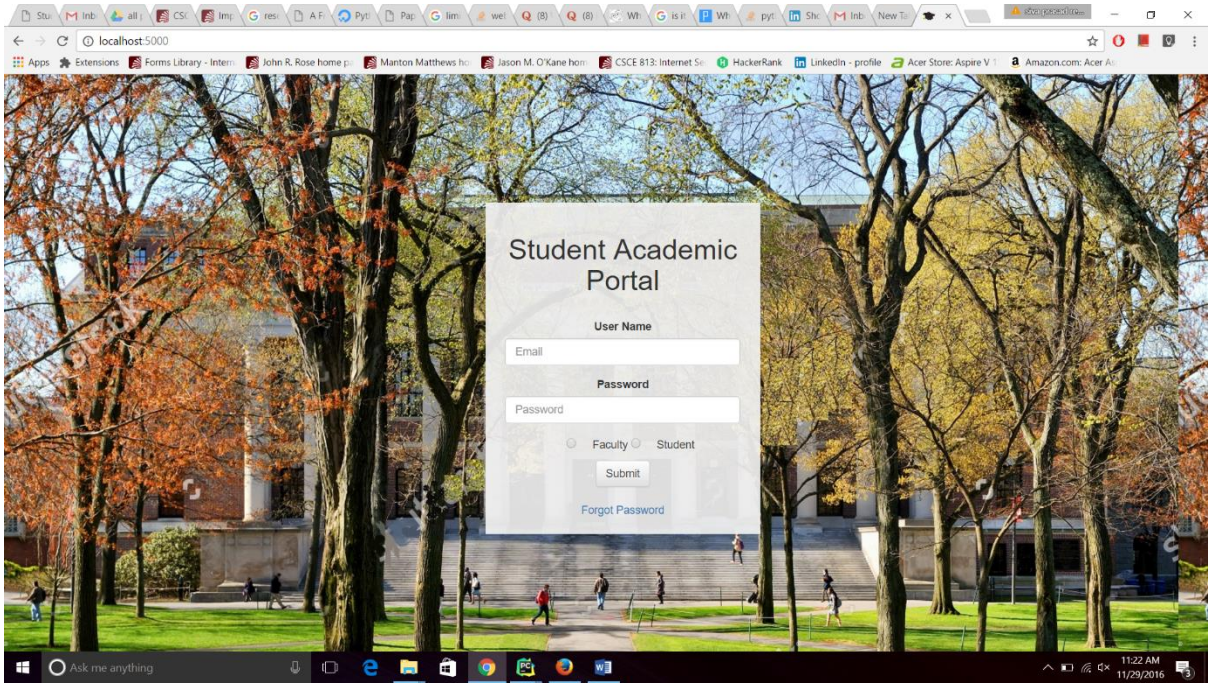
Report By:
Aditya Tripathi, Aman Sharma, Chandra Bhatt, Siva Reddy, Sri Sai Lekha

- **Front End Design and Modules:** We have made different modules in our application which is implements different security features:

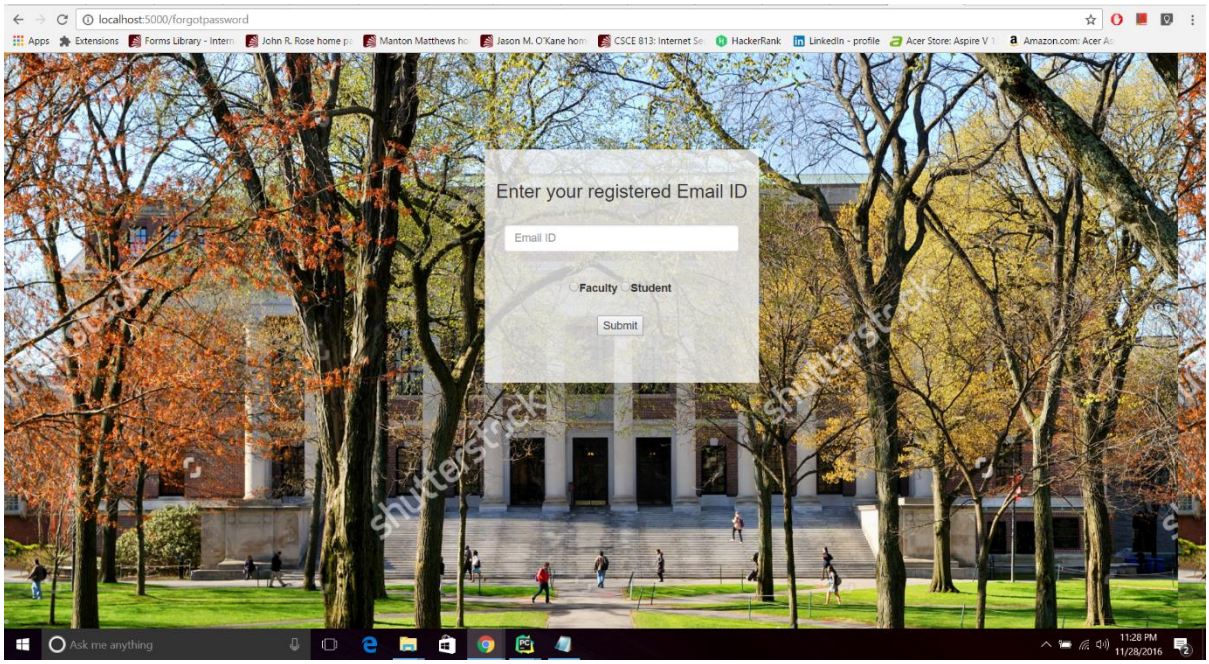
- Home Page – Authentication
- Student Module – Integrity and Confidentiality
- Instructor Module – Integrity and Confidentiality

- **HOME PAGE:** The screenshot below shows the homepage of the student portal which has a login screen accepting user Id's and password, while letting user choose whether he wants to access Student Portal or Faculty Portal. **Authentication** security feature is exhibited by this page in which only the authorized users will be able to access the portal and further login to it. Python program running at the backend query the SQLite database as soon as user hits Submit Button and check for the correctness of credentials entered by user. Different actions are taken on different conditions and few conditions are described below:

1. Any of the field cannot be left blank.
2. Wrong UserName or Password will prompt the user and will not allow access inside the application.

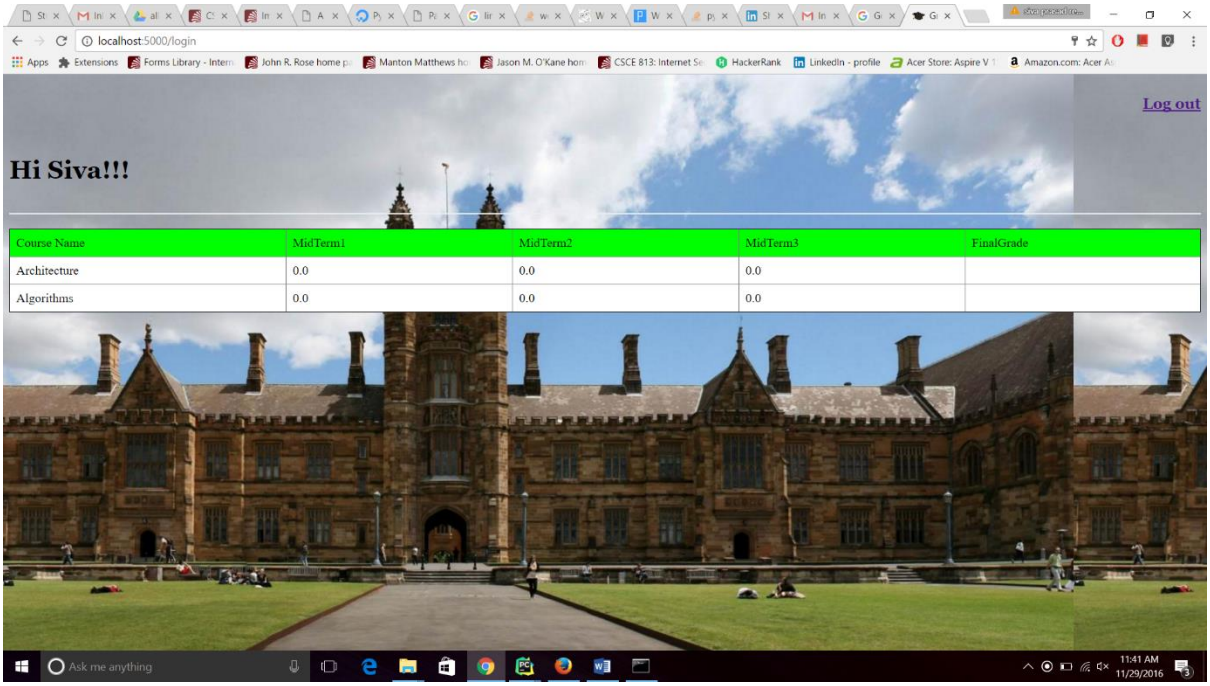


We have also incorporated a Forgot password functionality which helps the authorized user to recover his password using his E mail id. The mail with a password is automatically sent to the user at the registered mail Id present in the database. The below Screenshot shows the forgot password window functionality. Also, this is totally secure because any user who knows registered email ID and has access to that email ID will be able to access the email sent from StudentAcademicportal. This data is not lost anywhere in between.

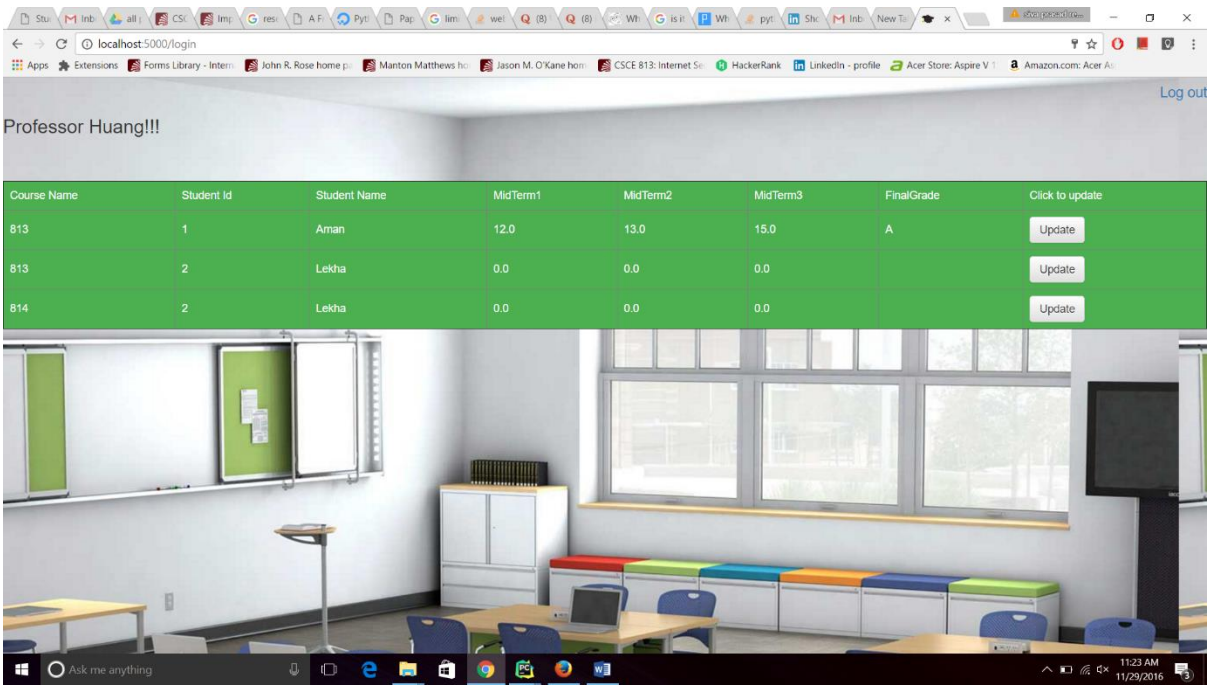


- **Student Module:** The screenshot below shows the student home page post authentication of student credentials at the login screen. As it is clear from the data fetched in the above table from the database that it shows the name of the student, marks obtained in Mid Term1, Mid Term 2, Mid Term 3, and also, the Final grade along with respective course names. This **confidential** information cannot be accessed by any other student taking the same courses and hence it implements Confidentiality aspect of the data security. Also the data which is shown to student is fetched by a Python program from a SQLite database where both program and database are inaccessible to any user and hence cannot be modified. Thus this also includes **Integrity** feature of internet security.

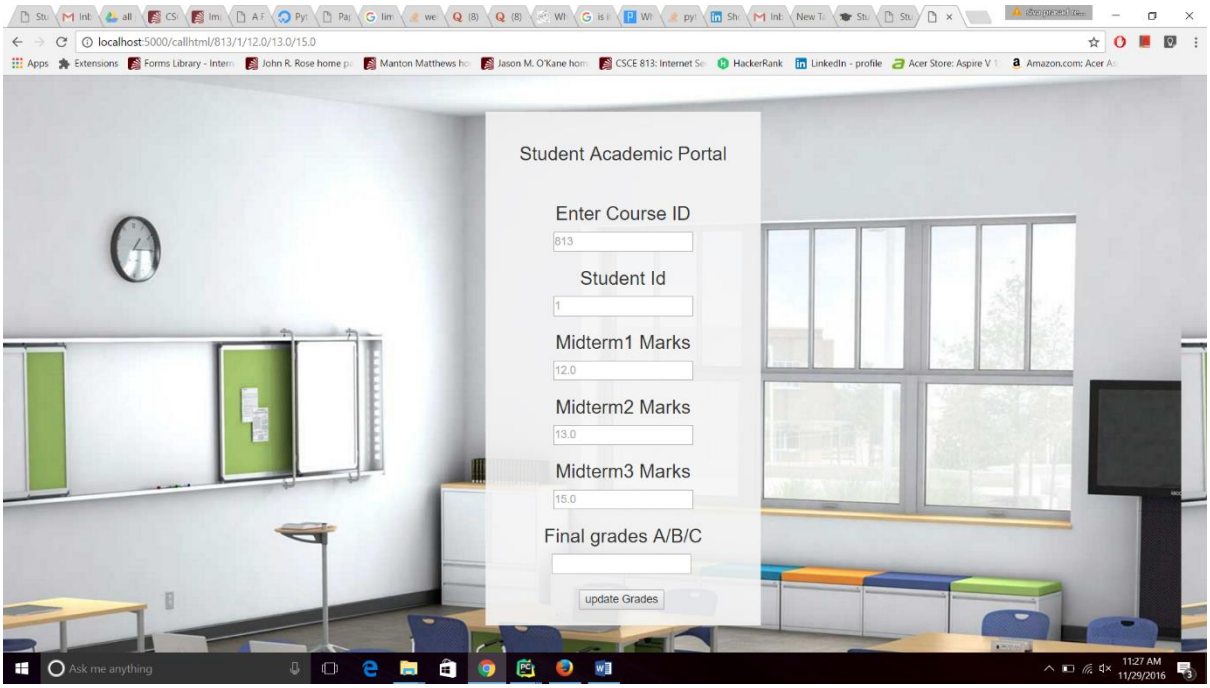
Report By:
Aditya Tripathi, Aman Sharma, Chandra Bhatt, Siva Reddy, Sri Sai Lekha



- **Instructor Module:** The screenshot for instructor module is shown below. Since instructor is responsible to update the grades of students for the courses that he is offering, he is give Write access to database. But since he is authenticated before he can update the database it is secure. Moreover, we are implementing **confidentiality** by making sure that the SQL query running at the back end gives access to instructor to only those records in which Student has taken only courses that he is offering. Instructor is prevented to updated and even Read the grades of students who is not taking his course. Also, the grades that he enters are not modifiable anywhere in the Web application and hence it maintains **Integrity** of the data.



Report By:
Aditya Tripathi, Aman Sharma, Chandra Bhatt, Siva Reddy, Sri Sai Lekha



Conclusion

The application could be categorised as one of the research level projects in which we have implemented security techniques using a python code base with proper authentication measures. Although, more enhanced functionality with robust security techniques can anyways been developed with greater precision.

In this implementation project we have successfully implemented security features into an online student portal using Python programming and Html, CSS web development. The security features currently implemented are Confidentiality, Integrity and Authentication.

Areas of Improvement: A future scope for the proposed student portal can be implementation of more robust and versatile security mechanisms which could deal with diverse malicious attacks. Also, cryptographic techniques and password encryption has not been implemented in our project which can be a future scope of improvement. Also, this student portal has been programmed to suffice the minimal needs to exhibit the basic required security features namely Confidentiality, Integrity and Authentication and is not developed on a commercial level covering all the robust security needs and large student database.

References

1. Research paper on Efficient Way of Web Development Using Python And Flask:
   http://www.aiktcdspace.org:8080/jspui/bitstream/123456789/1367/1/python%20and%20flask.pdf

2. For styling of static web pages
   http://www.bootstrap.com/

3. Image reference for background in project
   https://images.google.com/

4. References for efficient flask programming
   https://www.syncano.io/blog/advanced-concepts-flask/

5. References for secure python libraries.
   https://pypi.python.org/pypi/Flask