**Advance System Project**

**CIS-5690**

**Central High School**

**Name: Siva Kumar Garapati**

**Student ID: 700635222**

**Professor: Dr. Narasimha Paravastu**

**Introduction**

I would like to do Central High School website for my advanced systems project using ASP.Net. The website consists of the Home page where we can find logins system for the student and professor. The website homepage has the schools operation. In homepage, there is login on the top right of the page. In the login page, we have a login for the student, professor.

The homepage of the website provides information about school-related activities and function. The academic and sports page provides an information in the high school. This is normal website such other schooling and university websites. The login page as a link for creates an account to register the new users. The browser as contact us page for information, all details are saved in the database.

The login page is used to login the users with login credentials while provide while registration. The link that can provide to new user to create a new account according to preferred account type. There was validation provide for each textbox in the registration page.

The Document consist of the description of the project in a diagrammatic manner are UML diagrams and Entity relationship diagrams.

**Technologies used:**

**Development Environment (IDE):** Visual Studio2015

**Database platform:** SQL Server 2014.

**Server-Side Technologies:** Asp.net C#

**Client-Side Technologies:** HTML5, CSS, Bootstrap.

**UML Diagrams:**

UML stands for Unified Modelling Language. Which is used in object-oriented software engineering. It's typically used in software engineering it is a rich language that can be used to model an application structure, behavior and even business processes. There are **14 UML diagram types.**

They can be divided into two main categories are structured diagrams and Behavioral diagrams. The scenarios are organized hierarchically and they capture the system functionality at various abstraction levels, including scenario groups, scenarios, and sub-scenarios. Combining scenarios or sub-scenarios can form complex scenarios. Data are also separately identified, organized, and attached to scenarios. This scenario model can be used to cross check with the UML model. The modeling the logic captured by a single use case or usage scenario, or for modeling the detailed logic of a business rule. Graphical notations used in structural things are the most widely used in UML. The main types of UML Diagrams in this project:

Use case diagram

Sequence diagram

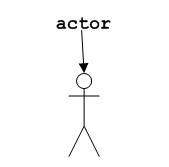
Activity diagram

E-R diagrams

**USE Case Diagrams:**

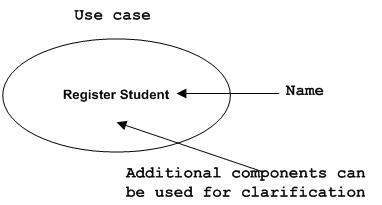
Actor Notation:

An actor can be defined as some internal or external entity that interacts with the system.



Actor is used in a use case diagram to describe the internal or external entities.

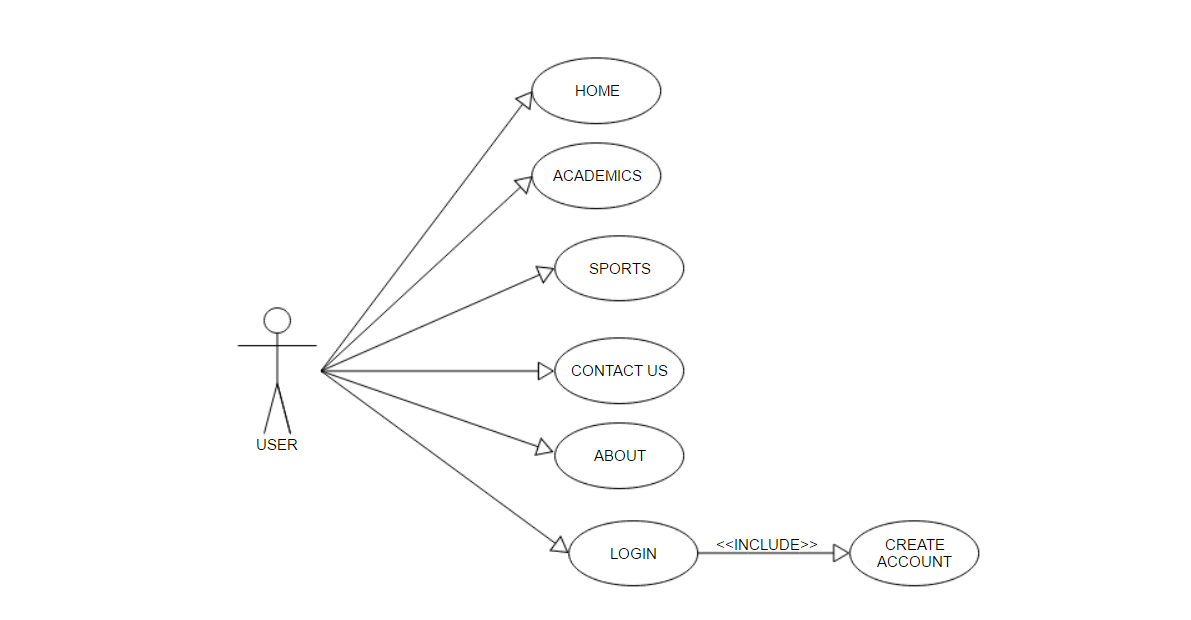
Use case is represented as an eclipse with a name inside it. It may contain additional responsibilities.



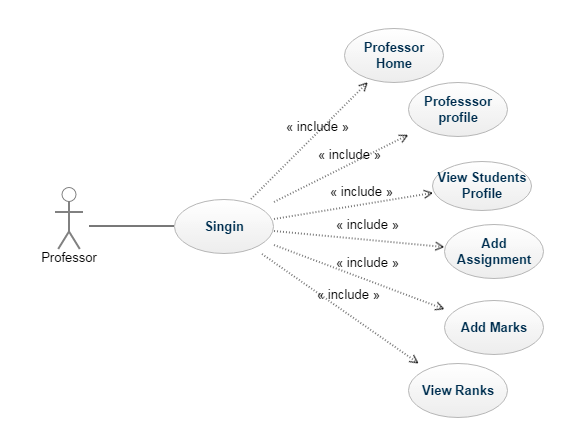
Use case is used to capture high-level functionalities of a system.



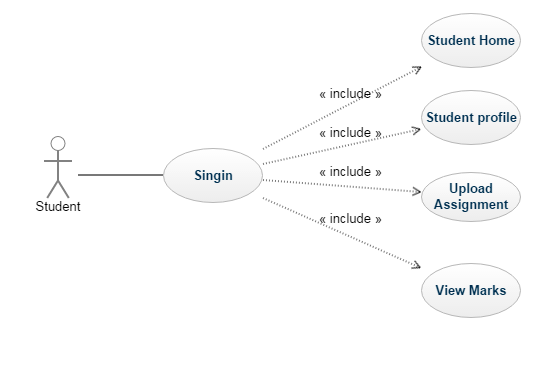
1. User use case diagram:



2. Professor Use case diagram:



3. Student use case diagram:



**Use Cases (Detail Description)**

**Case 1: USER Registration**

**Primary Actor:** User

**Agenda:** The New user can be able to create the account by in the central high school register application.

**Scope:** A User can login to the system application upon successful registration.

**Precondition:** All User should have a valid email id and Password to register in the Central High School application.

**Post Conditions:**

1. The new User can navigate through the information the web pages.
2. Click on the user login for login.
3. Click on create an account to register the new user.
4. Provide appropriate details for a new account for professor and student.
5. Register and access the account depends on the roles.

**Failure Conditions:**

1. Entering the invalid details Will registration.

**Case 2: Student Login**

**Primary Actor:** Student

**Agenda:** Student can be able to login to the application using credentials.

**Scope:** A registered student can access all the allowed menus after successful Login.

**Precondition:** Student should have successfully registered to the application before login.

**Post Conditions:**

1. A student can navigate to the courses using the navigation menu.
2. Login using appropriate emails and password.
3. View and navigate through the available access menu.
4. Logout Successfully.

**Failure Conditions:**

1. Invalid user Email or Password.

**Case 3: Student Operations**

**Primary Actor:** Student

**Agenda:** Student can be able to access the application menus such as Student Home, Student Profile, Upload Assignment and View marks.

**Scope:** A student can access all the allowed menus available in the application.

**Precondition:** Student should have valid login credentials to access the application menu.

**Success Conditions:**

1. A student can navigate to the navigation menu in the student home.
2. Login using appropriate email and password.
3. View and navigate through student menu.
4. Upload Assignments.
5. View Marks.
6. Logout successfully.

**Failure Conditions:**

1. Invalid user Email or Password.

**Case 4: Professor Login**

**Primary Actor:** Professor

**Agenda:** Professor can be able to login to the application with Email-id and Password.

**Scope:** Admin can access all the allowed menus available in the application without any restrictions.

**Precondition:** Admin should have valid login credentials and URL to access the application menu.

**Success Conditions:**

1. Professor user can navigate to the navigation menu in the Professor home.
2. Login by providing appropriate Email-id and Password.
3. All the navigation view through the available menu.
4. Logout Successfully.

**Failure Conditions:**

1. Invalid user Email or Password.

**Case 5: Professor Operations**

**Primary Actor:** Professor

**Agenda:** Professor can be able to access the application menu such as professor profile, Add Assignment, Add marks and View ranks.

**Scope:** Professor can access and edit all student profile and navigation menu available in the application.

**Precondition:** Professor User should have to provide valid login email id and Password to access the application system.

**Success Conditions:**

1. Professor user can navigate to the navigation menu in the Professor home.
2. Login by providing appropriate Email-id and Password.
3. View and edit student profile the available in the system.
4. Add Assignment and Add marks for a particular student.
5. View ranks are displayed for student dependence on total marks.
6. Logout Successfully.

**Failure Conditions:**

1. Invalid user Email or Password.

**Entity-Relationship diagrams:**

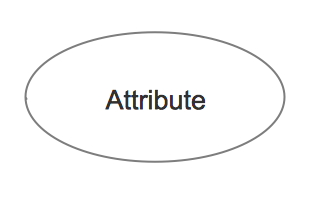
An entity–relationship model (ER model) describes composed of entity types and specifies relationships that can exist between instances of those entity types. The Semantic modeling is modeling data structures, based on the meaning of these data. Different variants of the entity-relationship diagrams are used as a tool for the semantic modeling. ER-model based diagrams have three main components: an entity, a relation and attributes. An entity is a class of similar objects, information about which should be considered in the model.

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business. It is usually drawn in a graphical form as boxes (entities) that are connected by lines. An ER model is typically implemented as a database.

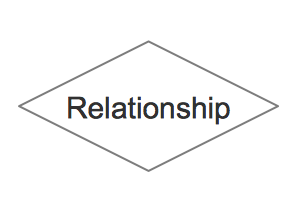
**Entity**: An entity is represented by a rectangle which contains the entity’s name.



**Attribute:** In the Chen notation, each attribute is represented by an oval containing an attribute's name



**Relationship:** A relationship where the entity is existence independent of another entity. A strong relationship is represented by a single rhombus. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database, a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity.

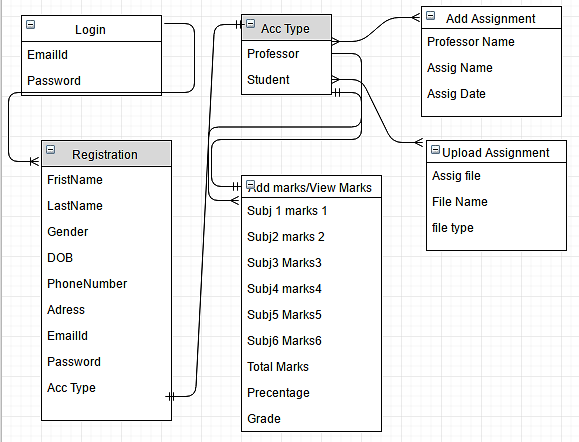


****

**Simple view of interaction of Professor and student through browser Diagram:**

****

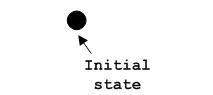
**Connection in database:**

****

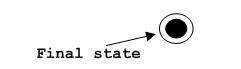
**Activity Diagram:**

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow form one activity to another activity.

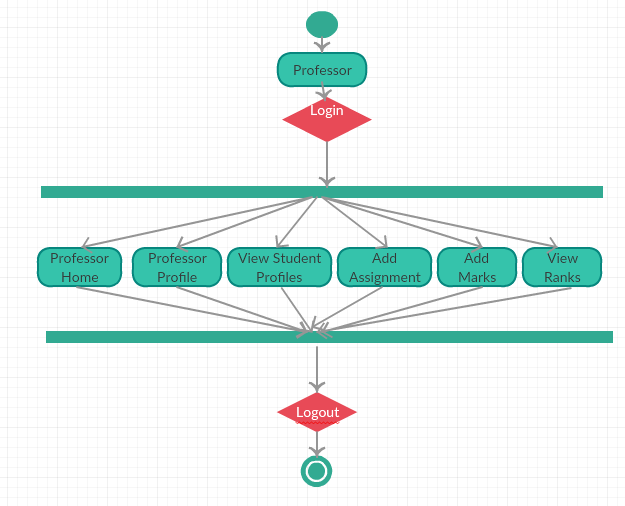
**Initial node**. The filled in circle is the starting point of the diagram. The initial state is defined to show the start of a process. This notation is used in almost all diagrams. The usage of Initial State Notation is to show the starting point of a process.



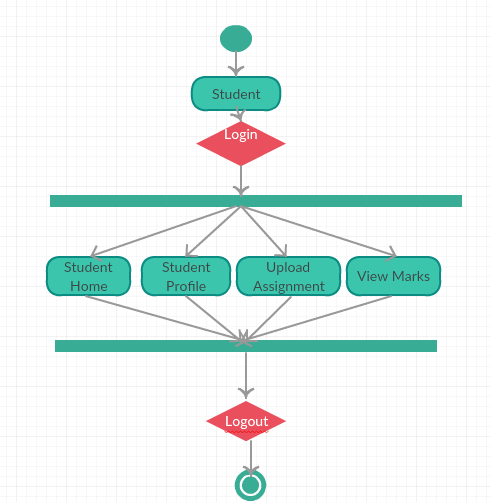
**Activity final node**. The filled circle with a border is the ending point. An activity diagram can have zero or more activity final nodes. This notation is also used in almost all diagrams to describe the end. The usage of Final State Notation is to show the termination point of a process.



.**Professor activity diagram:**



**Student Activity diagram:**

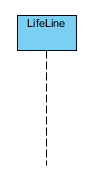


**Sequence Diagram:**

The sequence diagram shows how objects communicate with each other in terms of a sequence of messages. Interaction diagrams, a subset of behavior diagrams, emphasize the flow of control and data among the things in the system being modeled.

**Lifeline:**

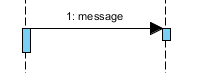
A lifeline represents an individual participant in the Interaction.



**Send message:**

A message defines a particular communication between Lifelines of an Interaction.

Send message is a kind of message that represents the start of execution.



**Return Message:**

A message defines a particular communication between Lifelines of an Interaction.

Return message is a kind of message that represents the pass of information back to the caller of a corresponded former message.

https://www.visual-paradigm.com/VPGallery/img/diagrams/Sequence/ReturnMessage.png

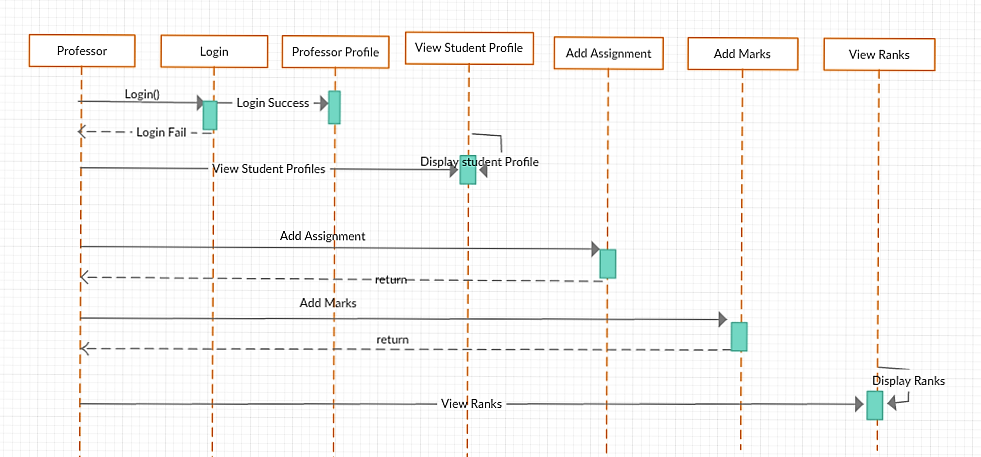
**Self Message:**

A message defines a particular communication between Lifelines of an Interaction.

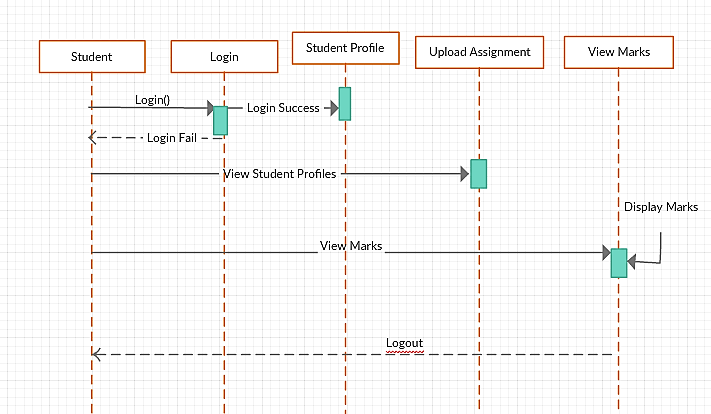
Self message is a kind of message that represents the invocation of message of the same lifeline.



**Professor Sequence diagram:**



**Student sequence diagram:**

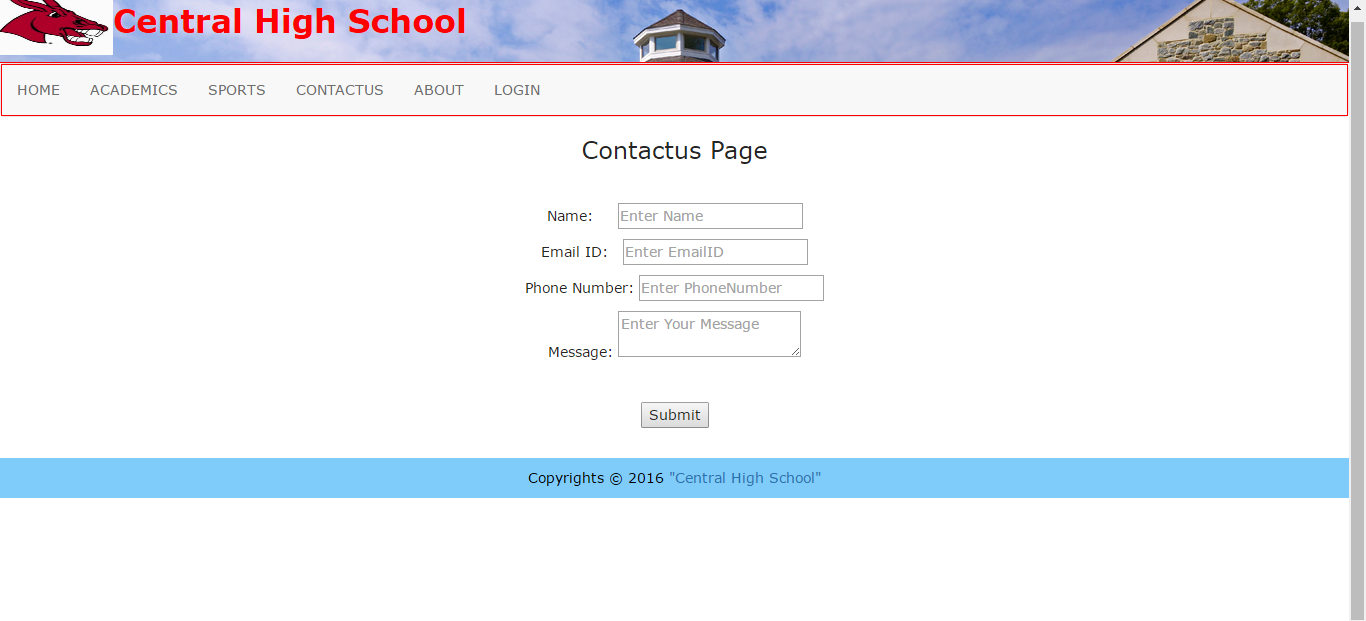


**Website Screenshots:**

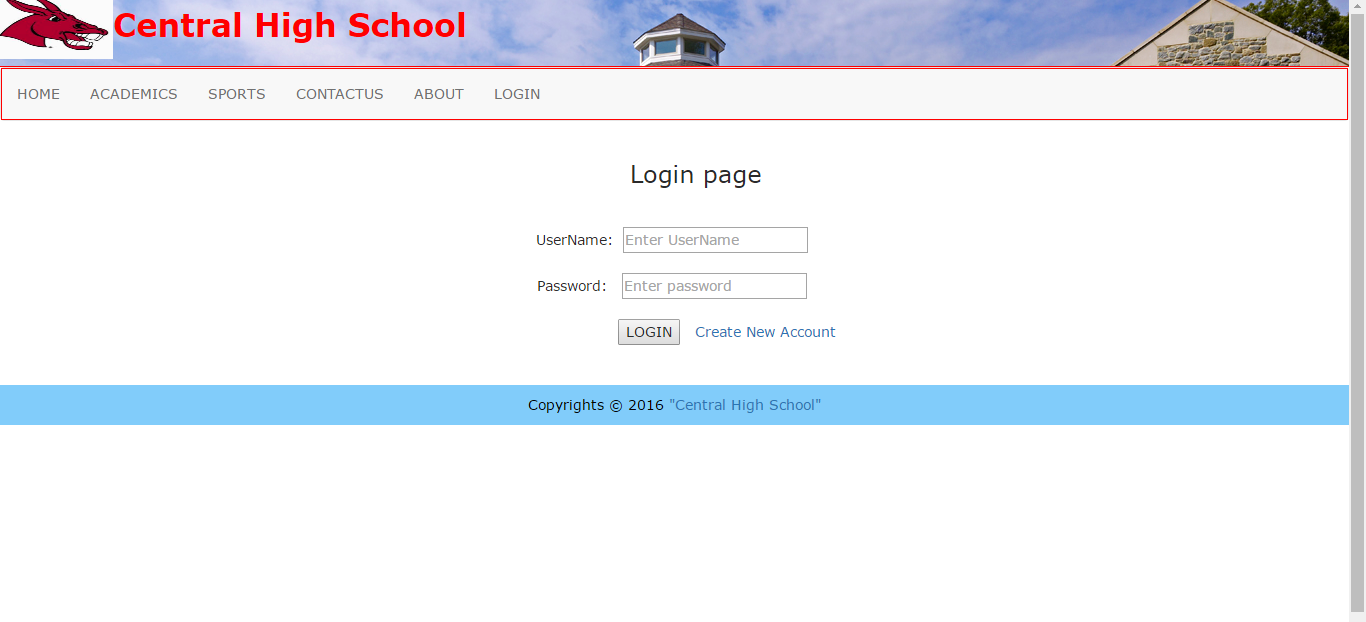
Home page:



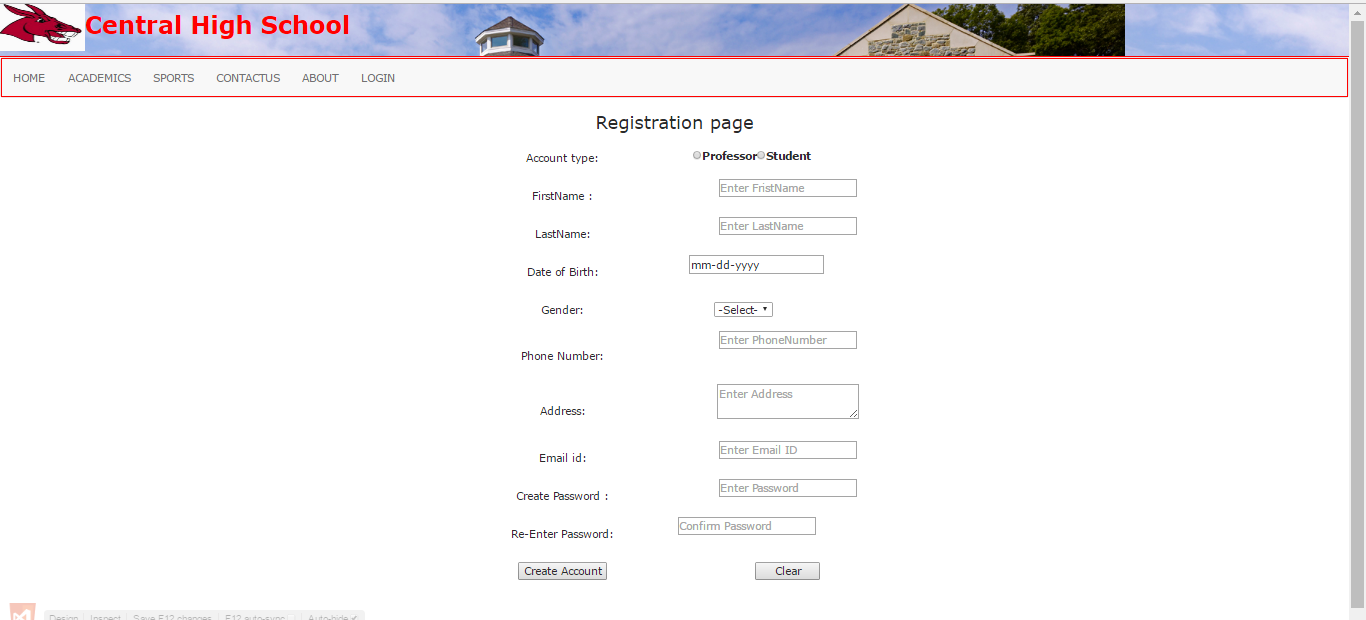
Contact Us page:



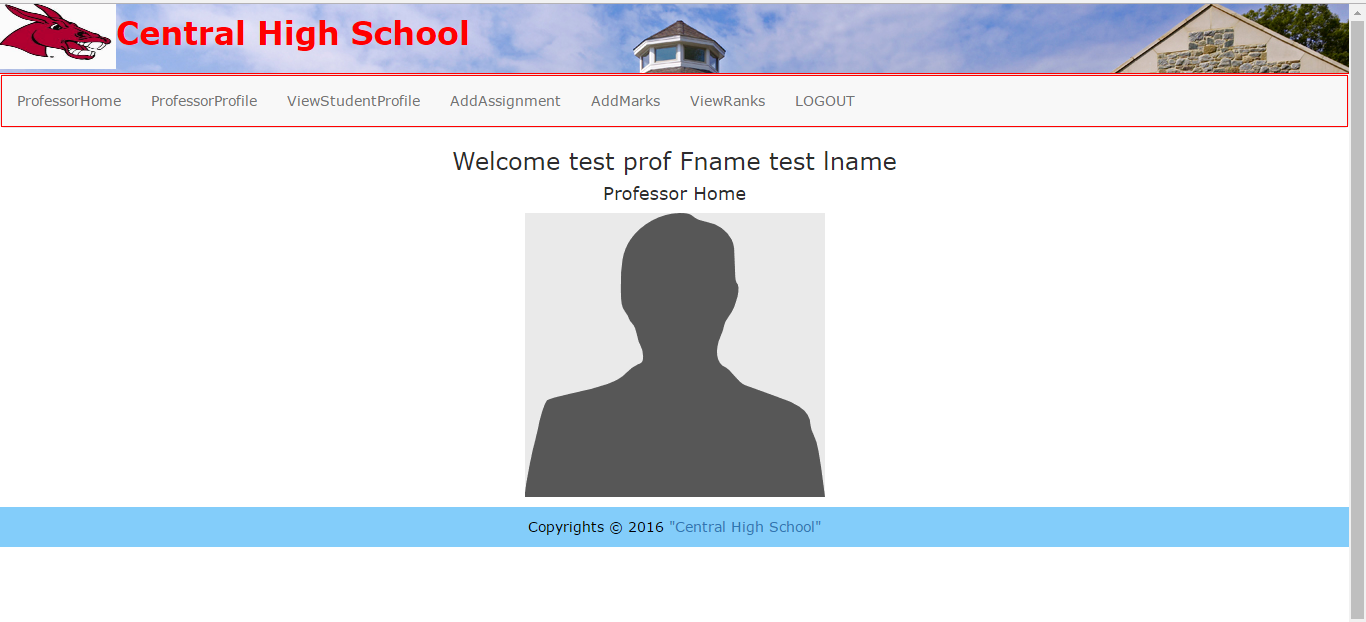
Login page:



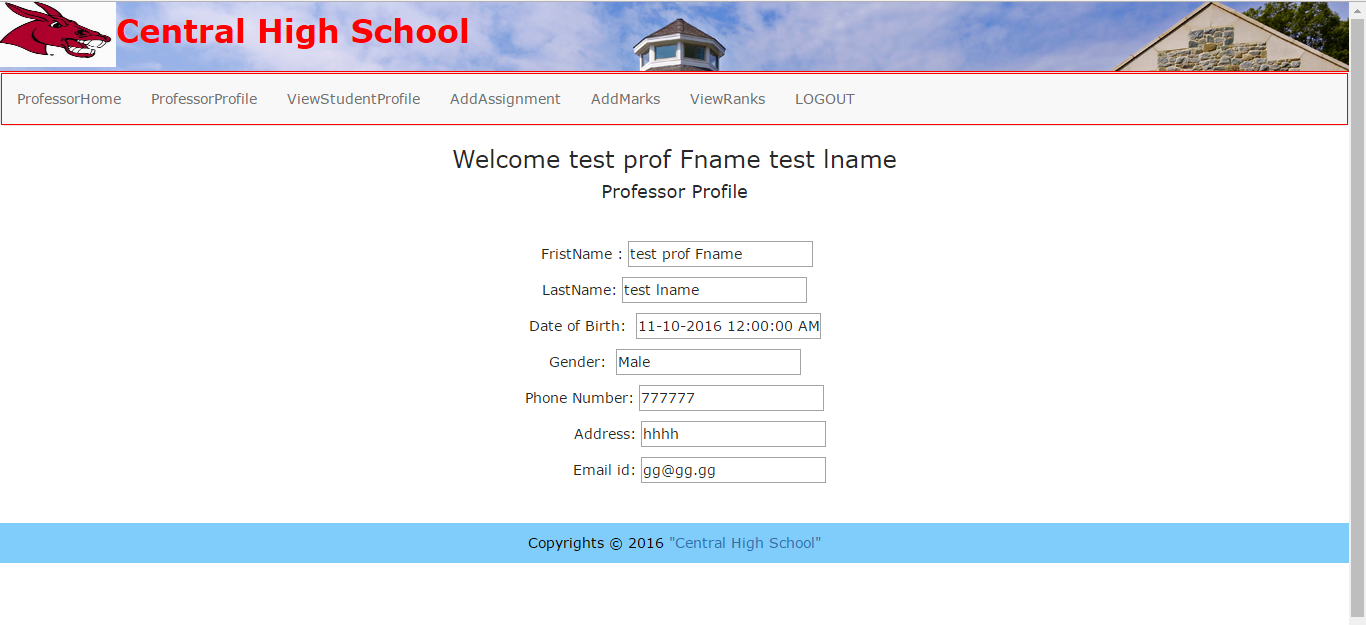
Registration page:



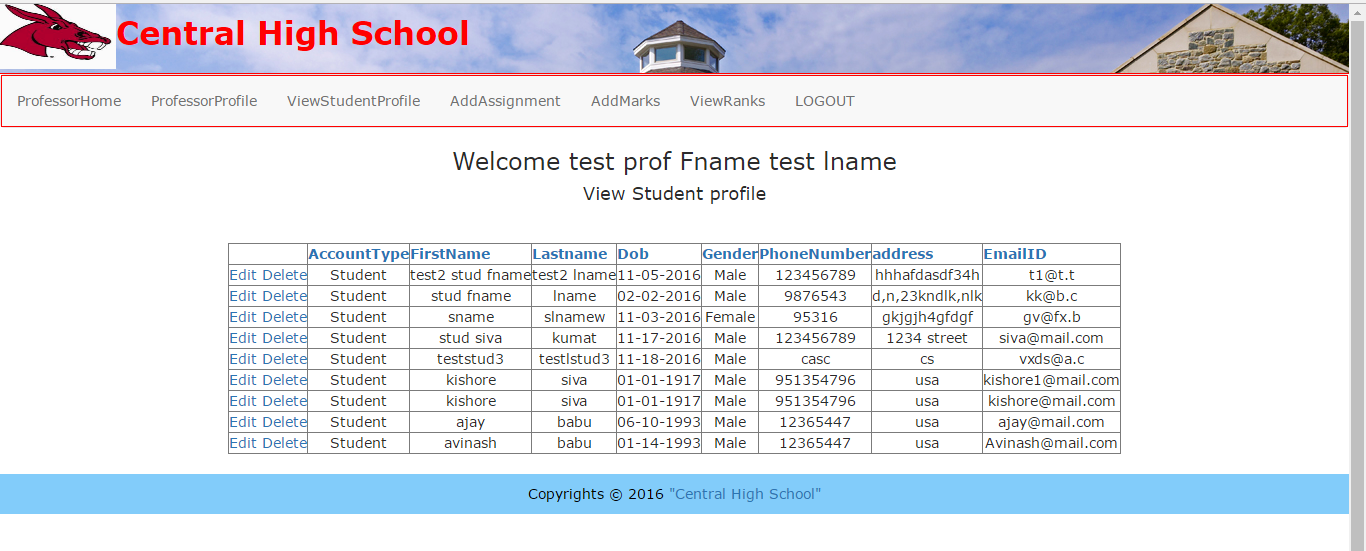
Professor Homepage:



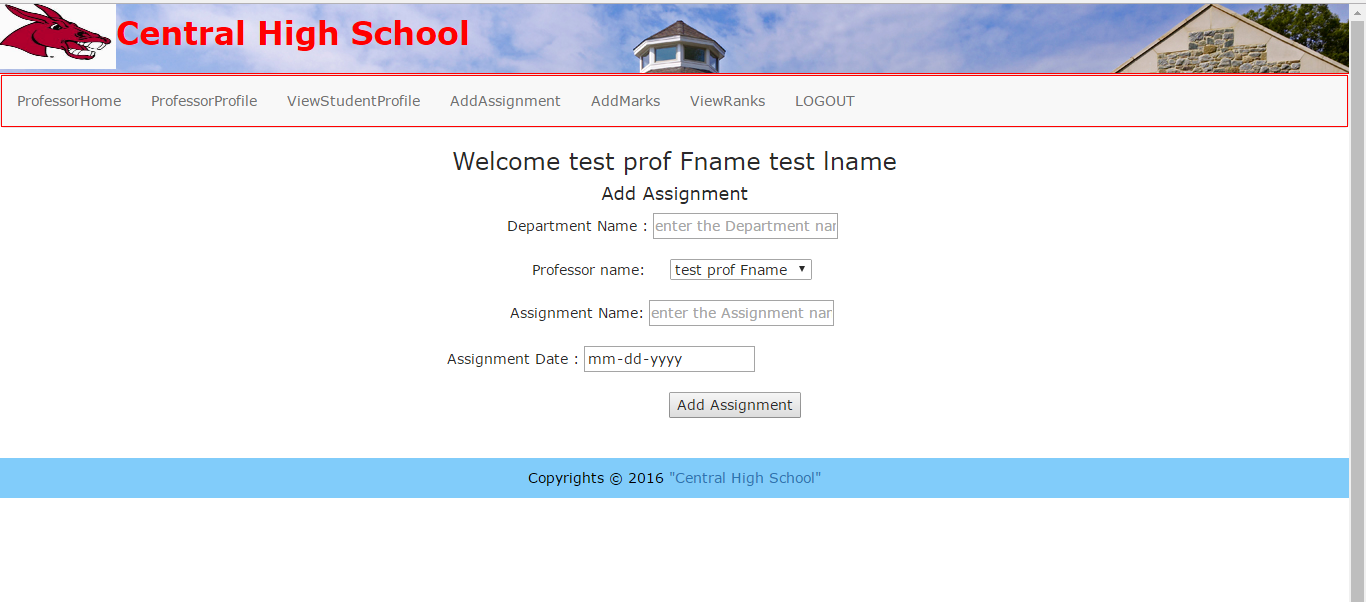
Professor Profile page:



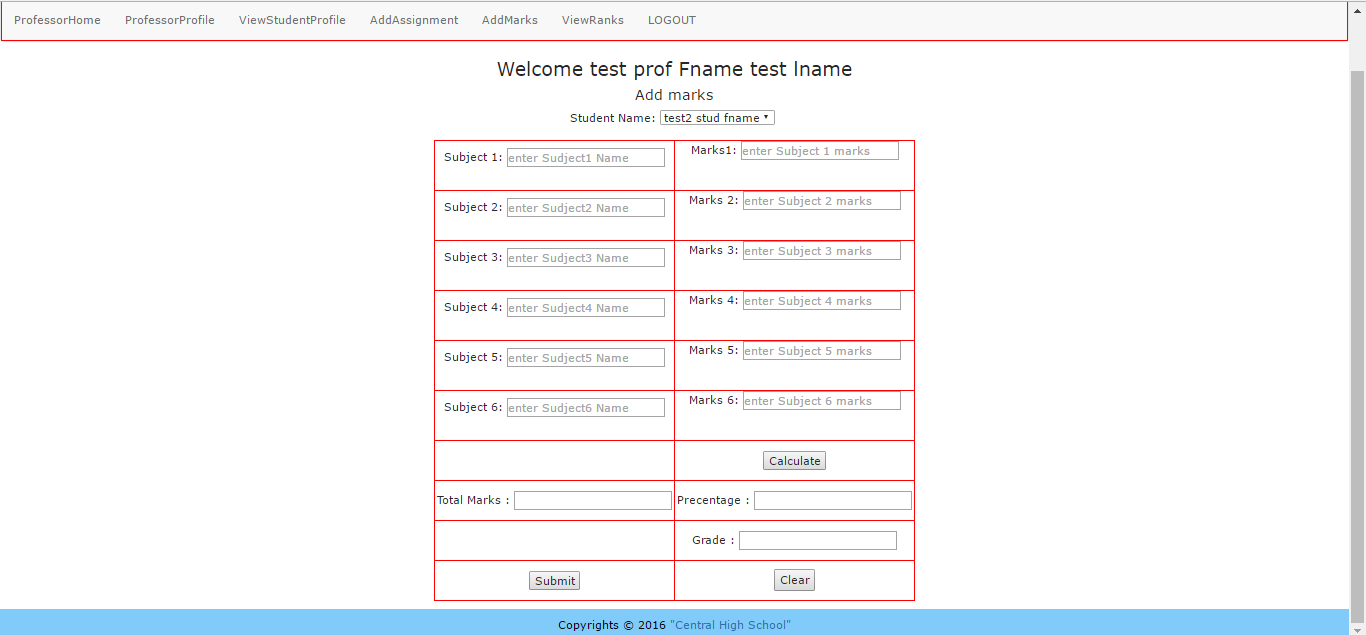
Professor View Student page:



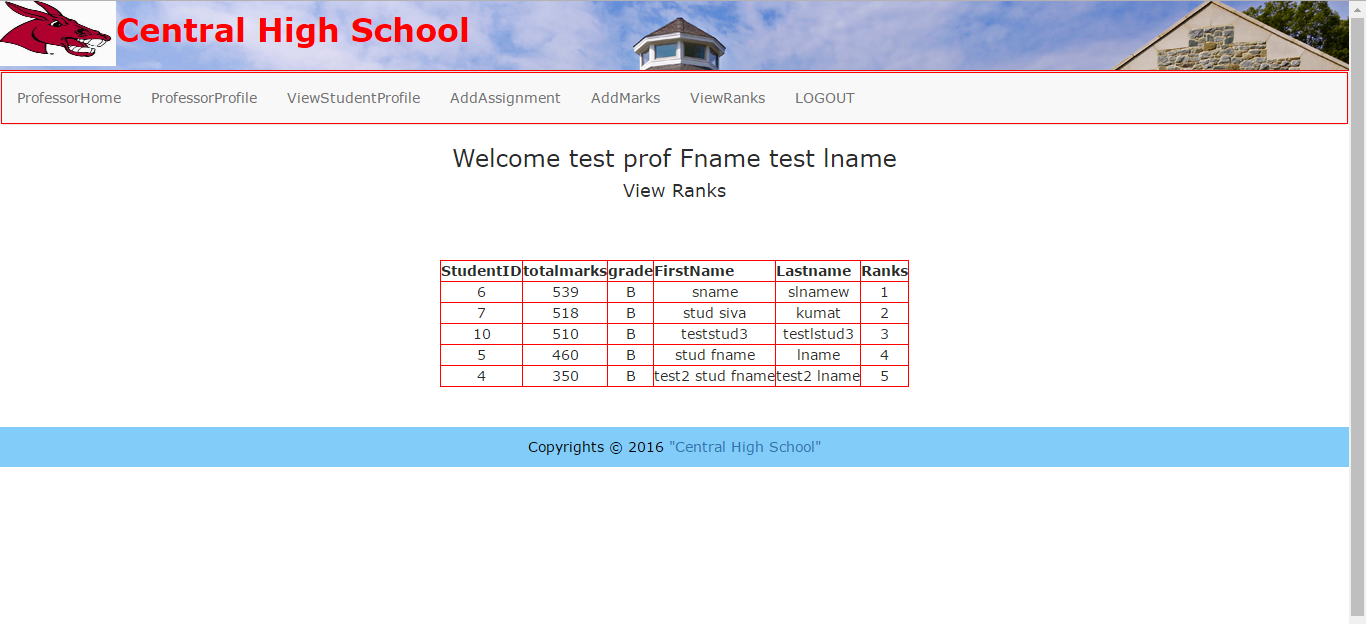
Professor Add Assignment page:



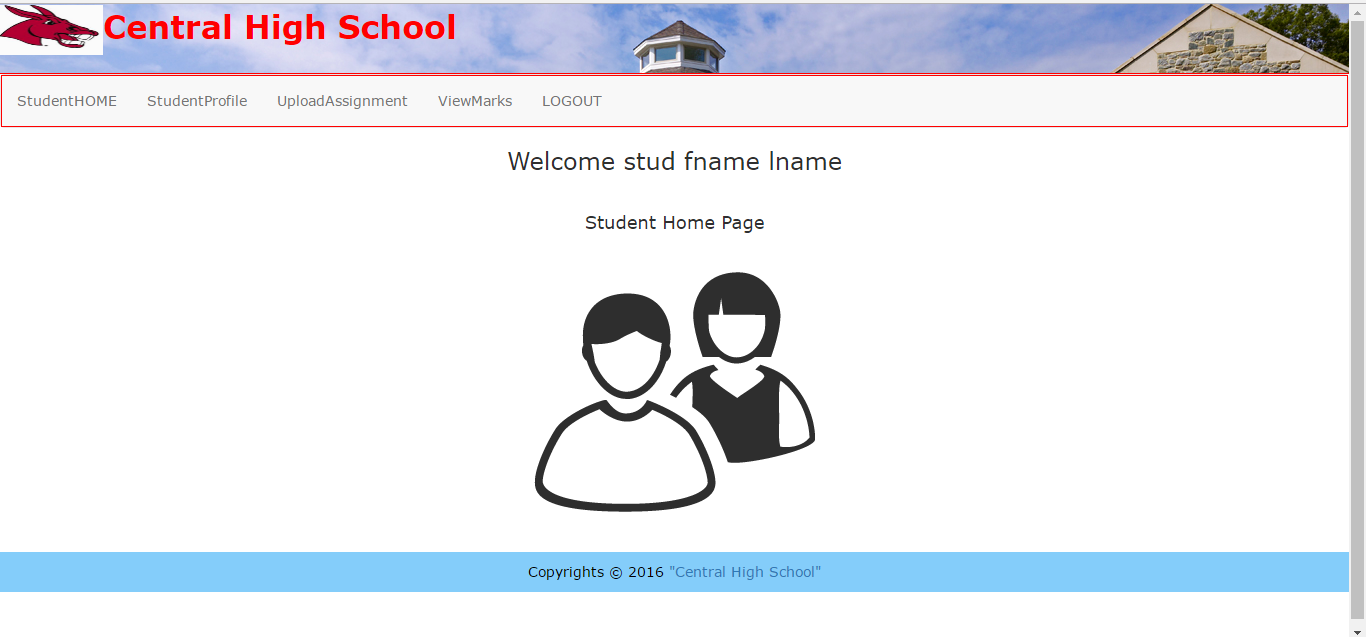
Professor Add Marks page:



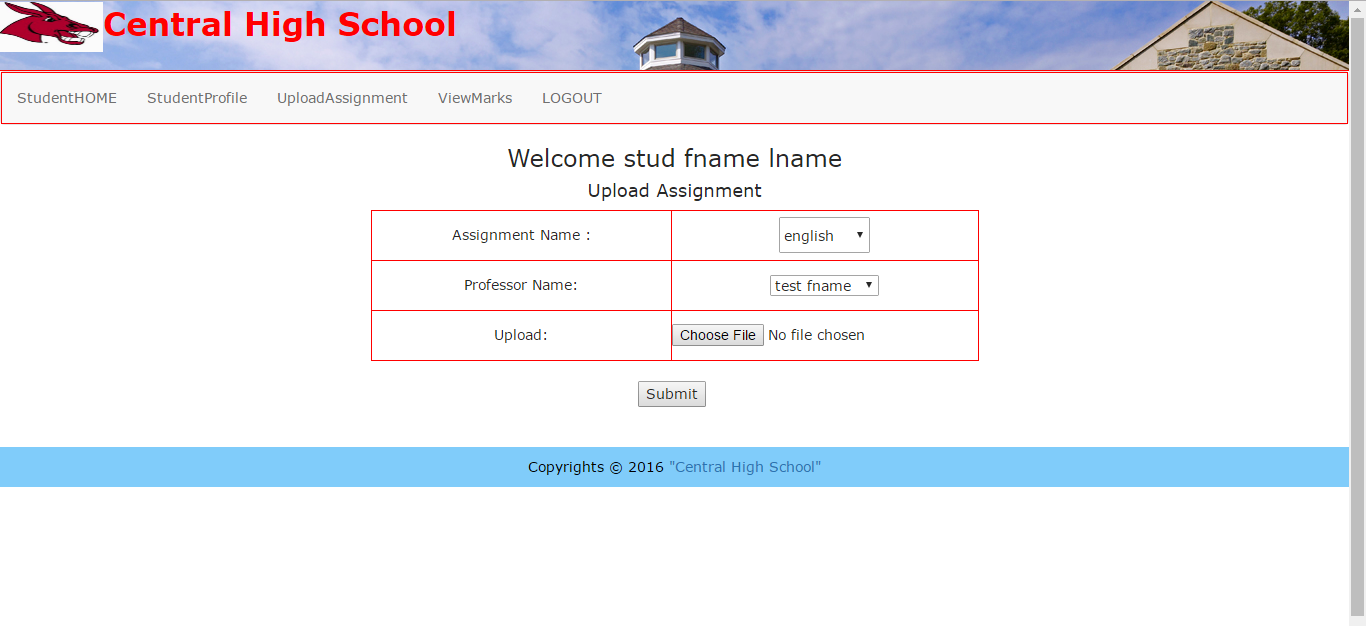
Professor View ranks page:



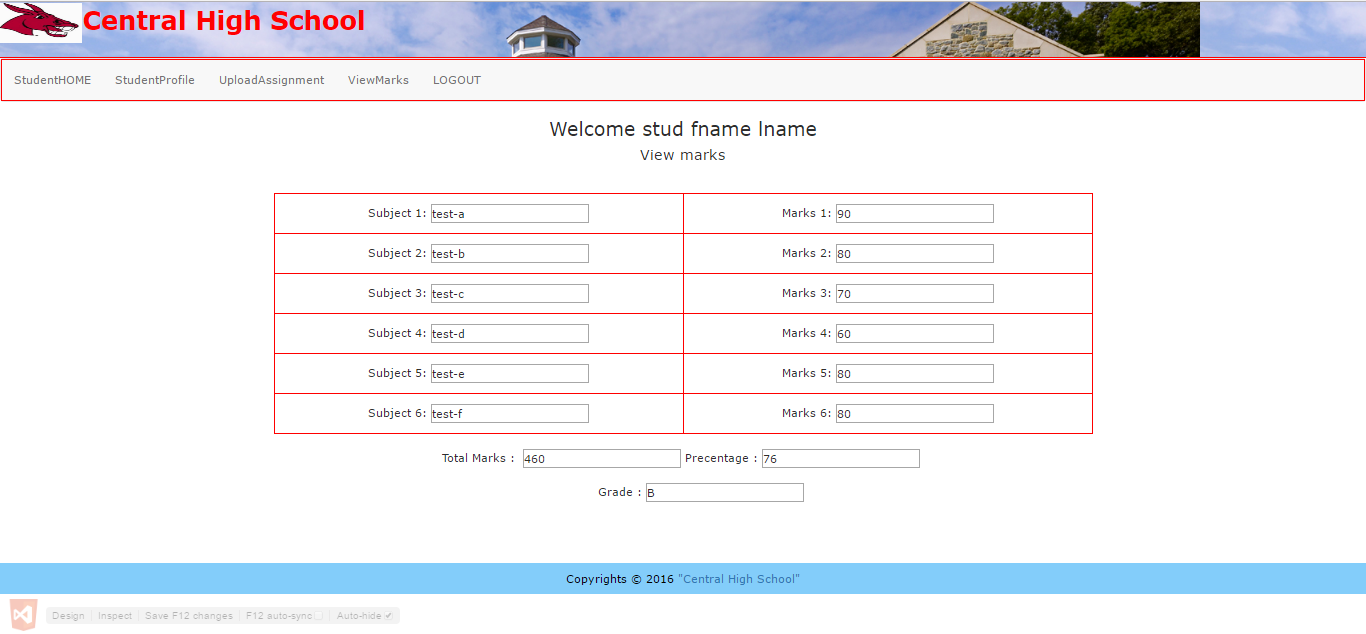
Student Homepage:



Student Upload Assignment page:



Student View Marks page:



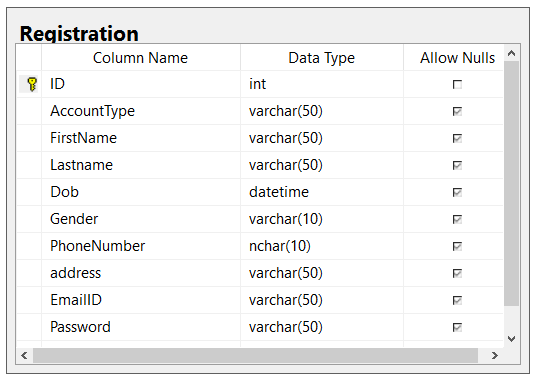
**SQL Server Screen Shots:**

The SQL Server consisting of five table:

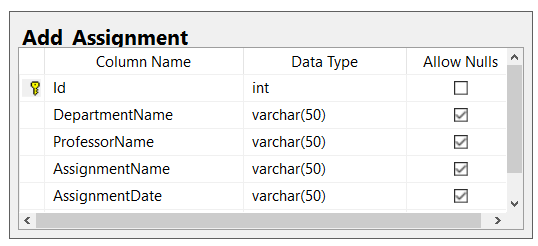
1. Registration Table – Used while registration of new account to the database

Used while verify the login credential from the users.

Used to retrieve the user information from database to the profile view page.

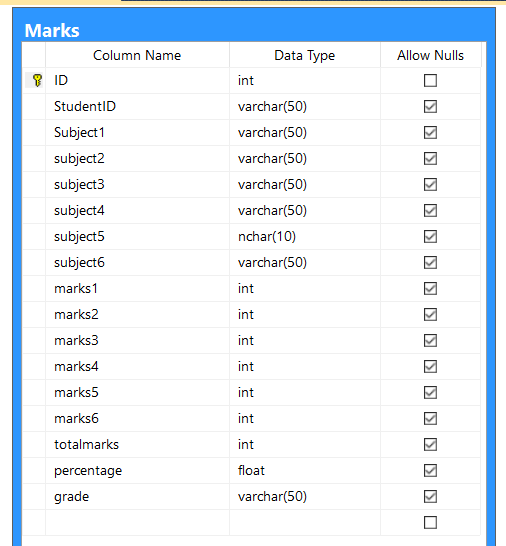


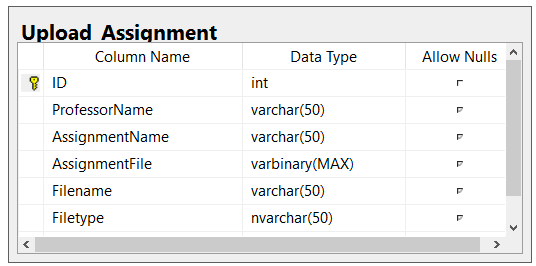
1. Add\_Assignment Table – Used to store the Assignment name for the professor from the add Assignment page.



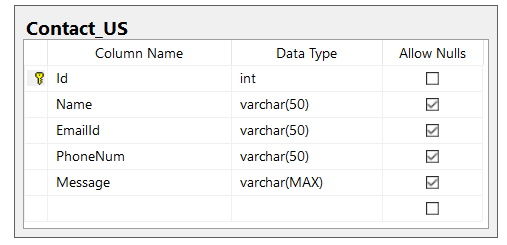
1. Marks table – used to store the mark & grades of individual student was upload by professor in the Add Marks Page.

Used to retrieve the marks & grades of particular student from the database to the View Marks page.

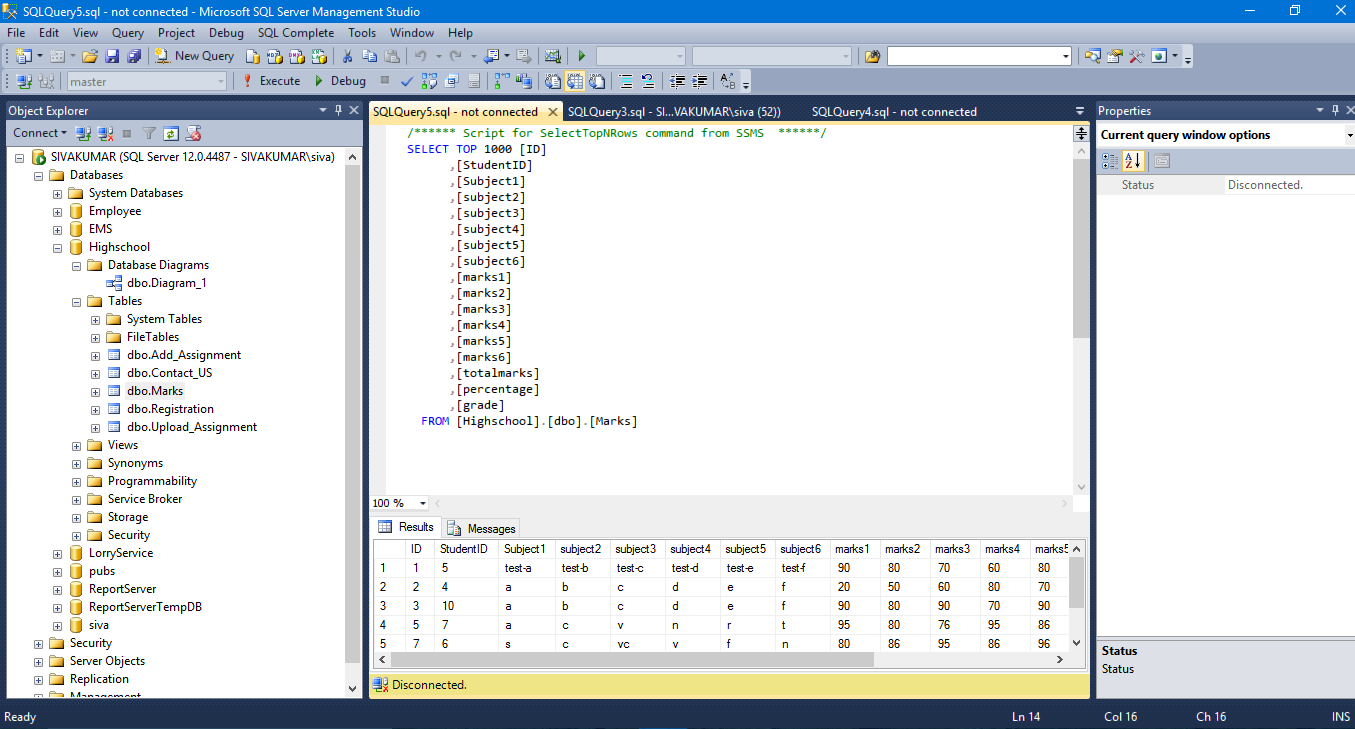


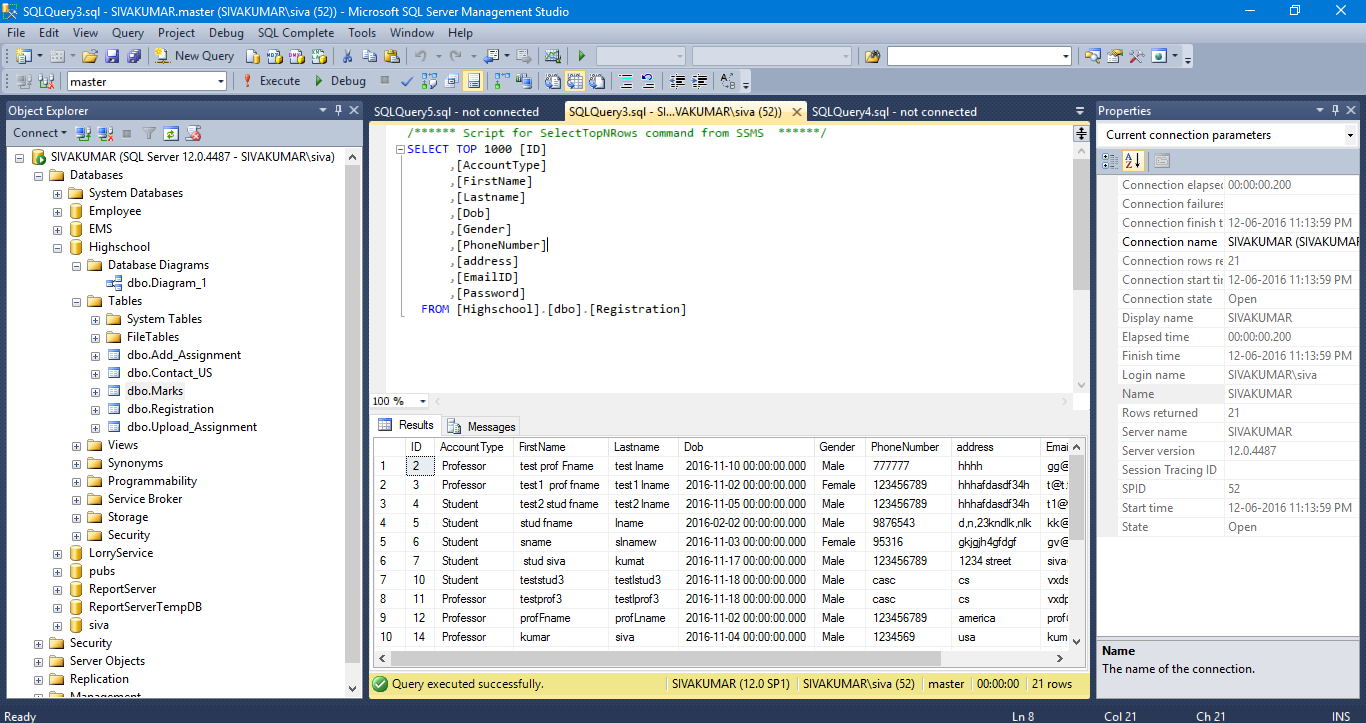
4) Upload\_Assignment Table- Used to Upload assignment file by the Student it stores the Assignment file in binary format in the database.it store the Assignment file(Binary format), Filename and Filetype. 

5) Contact\_US Table – Used to store the information for Contact US page.



Screen shot of SQL Server:





Conclusion:

The Central high school website gives an easy access of interaction the student and professor in the school. The view rank page provides an instant update to professor based on the change of marks and grade of student on the database. If there was an upgrade functionality to the application, it provides more feature to the application.

Reference:

1) http://www.w3schools.com/html/default.asp

2) http://www.w3schools.com/css/default.asp

3) <http://www.w3schools.com/bootstrap/default.asp>

4) <http://www.w3schools.com/sql/default.asp>

5) <http://www.w3schools.com/asp/>

6) <https://www.asp.net/web-pages>

7) UML sequence diagram - <https://creately.com/>

8) UML use case diagram - <https://www.draw.io/>

9) UML use case diagram - https://www.gliffy.com/uses/uml-software/