COMP 5531 Databases

Final Project

Lecture Section: NN

Lab Section: NNNI

Professor: Dr. bipin c. DESAI

Marker: yogesh YADAV

Due: Saturday April 23rd 2022

Group Name: COMP 5531_group_4

Group Leader: michael POULLAS, 21917730, m_poull@encs.concordia.ca

adriana DANKO, 40029681, a_dank@encs.concordia.ca

christopher SAVOV, 40106262, c_savov@encs.concordia.ca

shafiq IMTIAZ, 40159305, s_mtiaz@encs.concordia.ca

TABLE OF CONTENTS

Project Description

Our CGA Course Management System is created using CSS/HTML/JavaScript/ and PHP/MYSQL. The purpose of this project is to design a web based user interface to gather information regarding instructors, TA's and students from an institution. A variety of web pages and database content is created for different role activities within the system. These include; the system administrator, professor, TA's and students. Each of these user roles will have their own entry to a specific user mode where they will be capable of executing tasks according to their defined role. The web page content and data access was designed for each user mode as they have certain tasks they are eligible for. For instance, the higher the user status i.e: system administrator or content creator, the greater accessibility and functionality a user may have access to in the system. The system administrator has full privilege and full access to all the services of the system. Some of the many include insertion/deletion of a course, editing of course sections, term dates, etc. Overall the CGA system includes but is not limited to obtaining details on students, groups, files uploaded, details on TAs and instructors regarding their access, etc.

The assumptions

According to the project description, our team had concluded a few assumptions while putting together and polishing our ER diagrams. Some of many include: each course has a group, each course has sections, each course has a forum and announcement, etc. A total of seven assumptions were made and they are noted below:

1. Each user is part of a course (professor) or course, section (ta, student).
2. Each course has a group.
3. Each course has sections.
4. Each course has a forum and announcement.
5. Each course has a task (marked entity).
6. Each task has a solution.
7. Each grade has a solution and student.

The limitations

Several limitations have been encountered while developing this database management system. Some of these include security breaches, destructive records, limitations of overall interactivity of the site and cascading of architecture.

1. Security breach: Due to our core development language being PHP, this project may be susceptible to security breach. This means other unauthorized users may obtain forbidden access into the system.

2. Destructive records: As we are using database tables with the DELETE function for the purpose of easing the development, this as a result will lead to destructive records. In other words, once the records are deleted there are no other alternatives to recover them.

3. Limitations of overall interactivity: basic HTML/CSS/JS and PHP/MYSQL is used. This as a result limits the functionality of the overall web pages. The overall interactivity and website experience should not be compared to projects with implemented frameworks.

4. Cascading of architecture: our database architecture is cascading. Thus, the constraints must be satisfied from top to bottom (start from role - ends at - grades).

The applications supported

Several different applications were used for this project and to ensure the delivery of the code was done in a smooth manner to the rest of the team. The applications used were Github, GitDesktop, Jira and Visual Studio Code (VS Code). Our developers would create a branch in Github and begin coding their respective tasks. After completing this portion, the rest of the team would update the code in VS Code and view the code on their end.

In the developer mode HTML, CSS, JavaScript, MySQL and PHP were used. HTML, CSS and JavaScript were used to design the web pages in the front-end aspect of this project. On the other note, MySQL and PHP were used to create the database. Furthermore, emails are delivered from the CGA platform which leads us to use SMTP (simple mail transfer protocols). The SMTP is an application used by email servers to send and receive emails between the users who send and receive (1). In order to have the email functionality working within the application, there must be an open SMTP connection on port 25. For the purpose of this project, port 25 is used as it is one of the primary ports for SMTP (2). According to the research done by our group, it is discovered that ENCS does indeed allow these emails as long as the user who is browsing on the user interface is logged into the ENCS VPN.

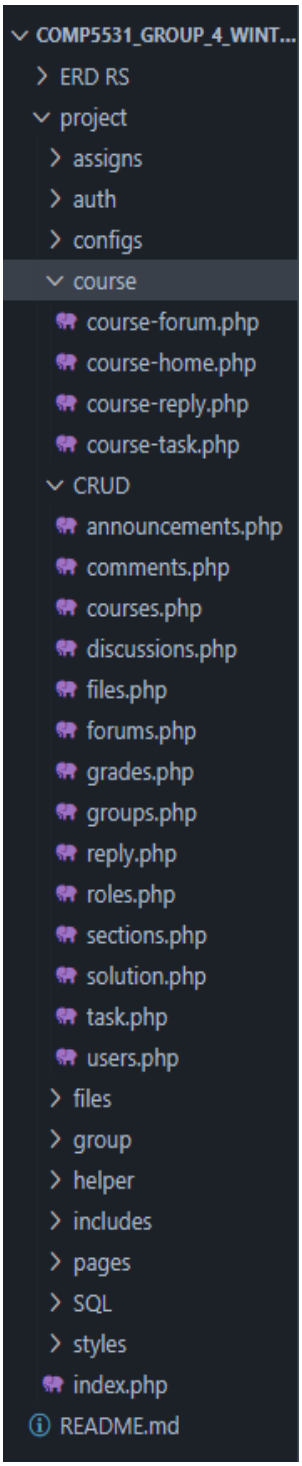
Github was mainly used to undergo testing requirements. A team member would clone a repository from Github and push a commit to the code. For the purpose of this project, simple tests were performed. For instance, automated testing was enabled when testing the code for this project. If a code encountered a problem, it would be directly revealed and opened into Github. The problem would then be fixed and pushed again. In addition to this, team members from the group would test the overall interface of the webpages and ensure the functionalities were working properly.

Architectural design

Our website uses a three-tier client-server architecture. There is the user interface tier on the client side which consists of the web browser and dynamic HTML pages. There is the database

management tier on the server side which is implemented using MySQL. We configured PHP to use the MySQLi interface to connect to MySQL. The processing management tier links both together using PHP as the scripting language. CSS is involved in the user interface tier to add style to our website. JavaScript is used to implement form validation in order to reduce latency and remove unnecessary processing on the network.

Our codebase's folder structure is done in a way that each folder contains files that are related and serve similar functions. There is a separation of concerns which makes the architecture logical and clean.



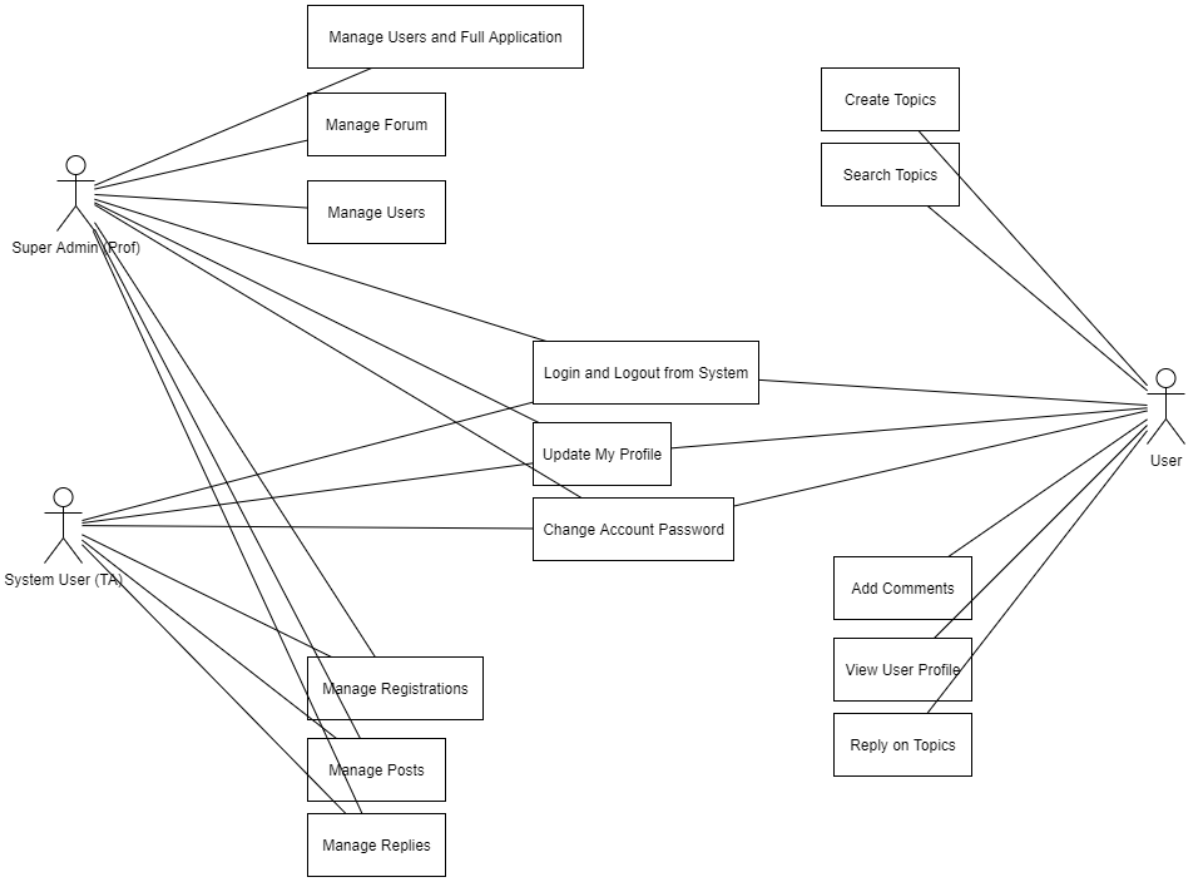
The most important folder is the CRUD folder. Each file inside has the same structure where we can CREATE, READ, UPDATE, or DELETE data in the database.
Below is an example of an ADD (CREATE) functionality.

```php
// ADD
if (isset($_POST['add_course'])) {

    // receive all input values from the form
    $course_name = mysqli_real_escape_string($conn, $_POST['course_name']);
    $course_number = mysqli_real_escape_string($conn, $_POST['course_number']);

    // form validation: ensure that the form is correctly filled ...
    // by adding (array_push()) corresponding error unto $errors array
    if (empty($course_name)) {
        array_push($errors, "Course Name is required");
    }
    if (empty($course_number)) {
        array_push($errors, "Course Number is required");
    }

    if (count($errors) == 0) {
        $add = "INSERT INTO course (course_name, course_number) VALUES('$course_name', '$course_number');";

        if (mysqli_query($conn, $add)) {
            array_push($success, "Course added Successful");
        } else {
            array_push($errors, "Error adding course: " . mysqli_error($conn));
        }
    }
}
```

We partially implemented a Model-View-Controller (MVC) pattern, specifically the controller to better manage and simplify routing of the various pages for each role type within each section.

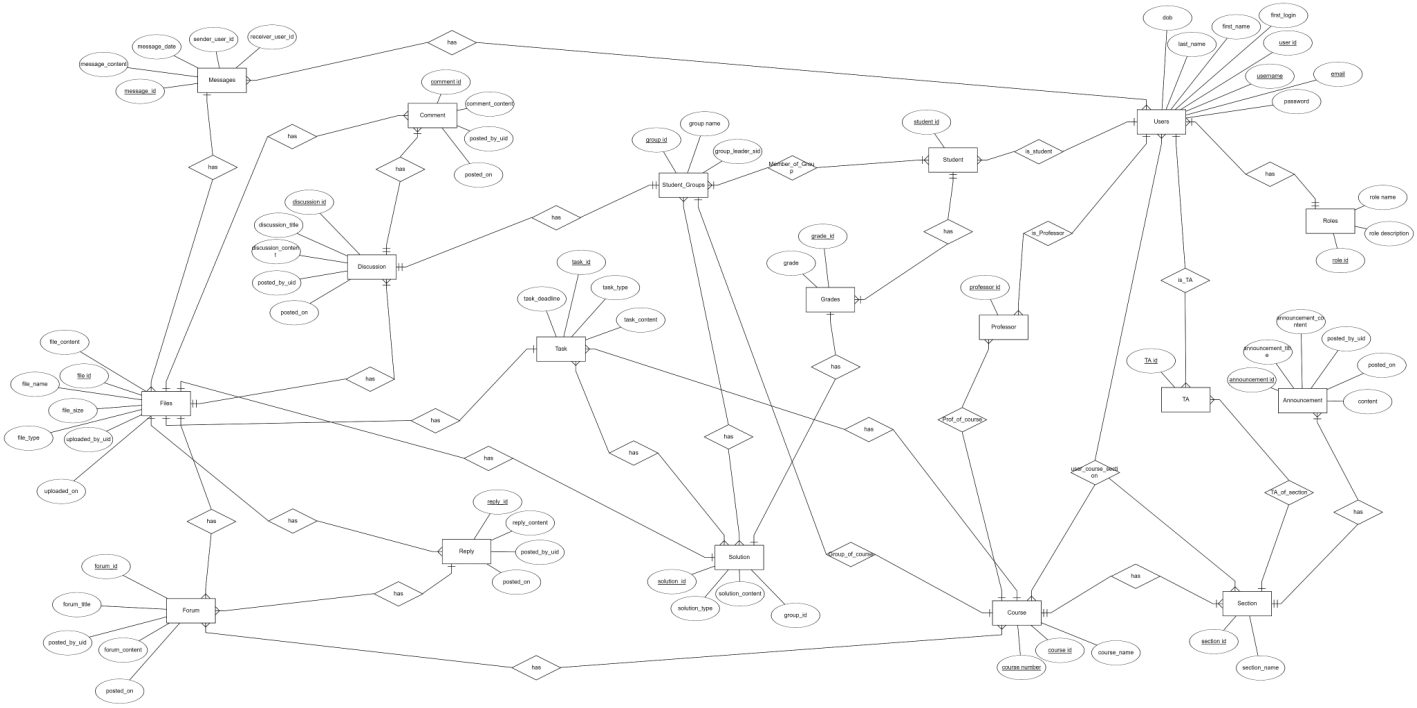
Use Case Diagram

The following use case diagram represents a graphical representation of the different user roles along with their interactions within the CGA system.



E-R Diagrams

The ER diagram was used to model the overview of our database design. The purpose of this is to get a bigger picture and understand how different entities relate to one another within the system.

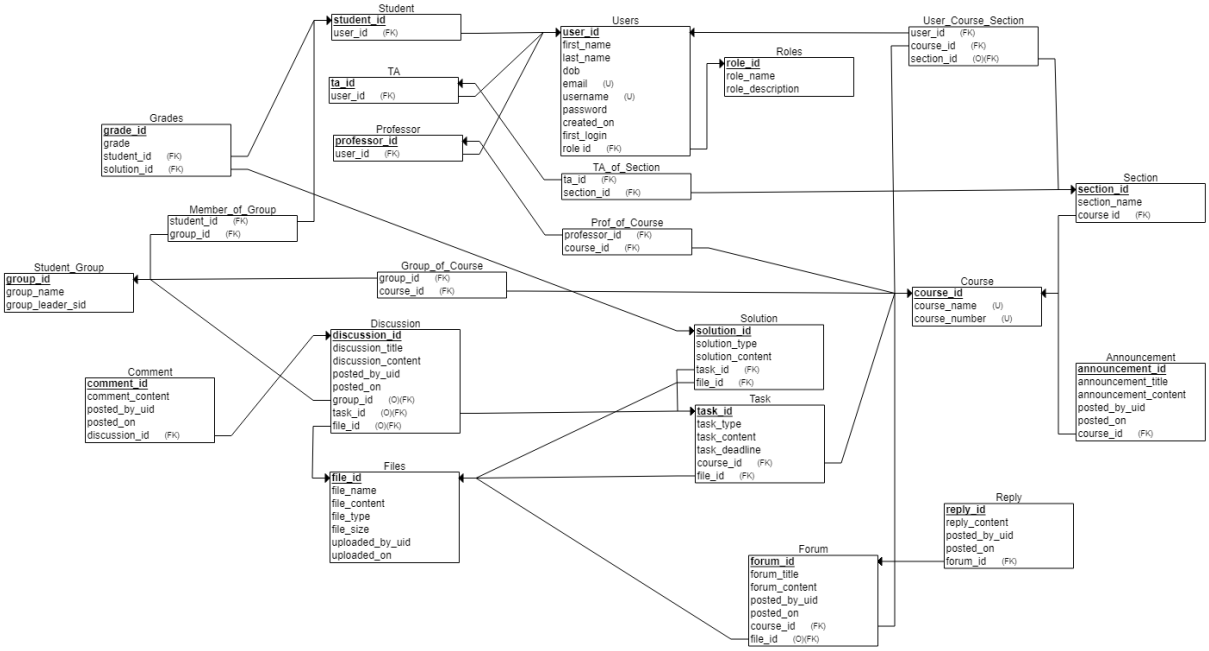


Link to the actual file to get a better resolution:
https://github.com/mikepoullas/COMP5531_GROUP_4_Winter_2022/raw/main/project/files/CGA%20ER%20Diagram.png

Relational Database Design

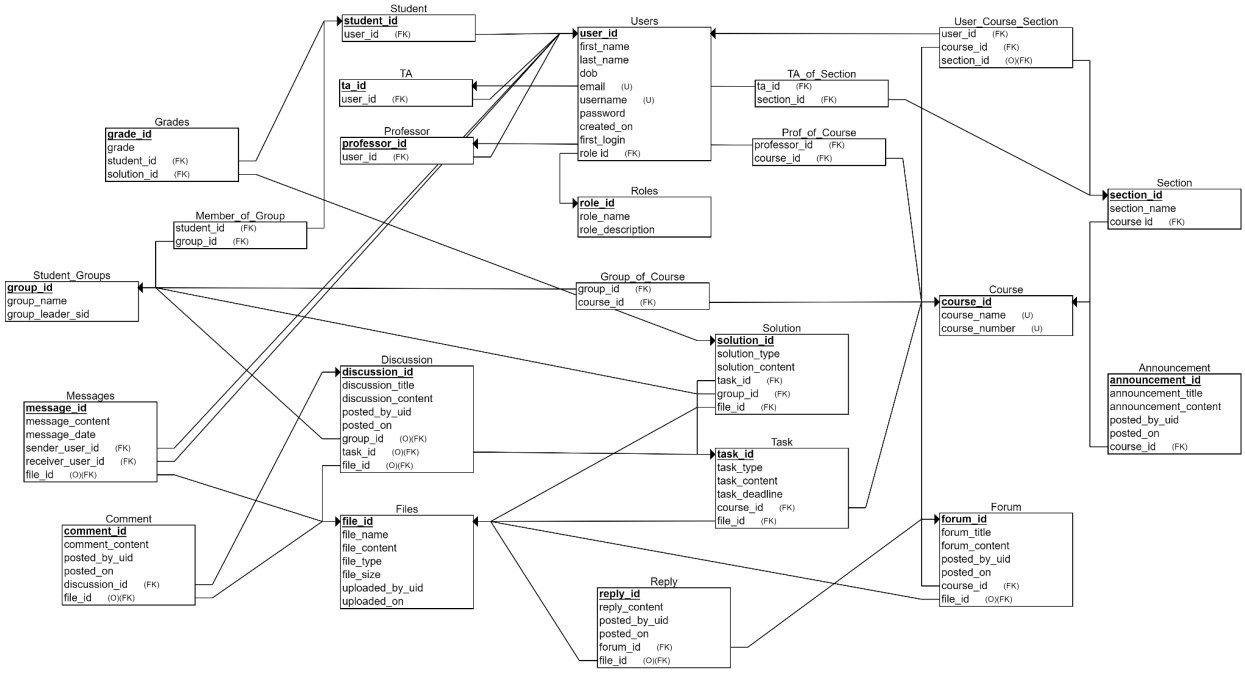
The relational schema of our database was created from the conversion of our ER diagram. This design was completed before any code was written for our website. The foreign key referencing is used to ensure proper linkage of data between tables. Non-null values are required as they were used to maintain proper relations. For instance if any errors, redundancies or anomalies were introduced, these would be prevented. To ensure a properly working database it is important to keep the data and information as clean as possible.

Before:



After:

-Main changes were the TA_of_Section and Prof_of_Course with Users.



Link to the actual file to get a better resolution:
https://github.com/mikepoullas/COMP5531_GROUP_4_Winter_2022/raw/main/project/files/CGA%20Schema.png

3NF Solution

Our initial design started with basic requirements to capture user information, configure specific roles to each user and then assign courses to those users. We would later add Group, Discussion and Comment capability all tied to individual user input. However, maintaining a large few simple tables in 1NF or 2NF form would introduce high redundancy issues as well as update and delete anomalies.
The update anomaly is caused by a transitive dependency. We need to remove such dependencies by moving to 3NF. This is about removing the transitive dependencies by identifying the determinant of each transitive dependency, which then becomes a primary key of a new table and then assigning the appropriate attributes. The new table name needs to be properly chosen and the dependent attributes need to be removed from the original table while the determinants remain and become foreign keys.
So if our USER table originally has a STUDENT_ID, TA_ID or PROFESSOR_ID attribute, they would all be removed to form their own independant table (STUDENT, TA, PROFESSOR) with STUDENT_ID, TA_ID and PROFESSOR_ID as the PRIMARY KEY respectively and USER_ID (determinant) as the FOREIGN KEY in each of those TABLES.
Bottom line, we identified transitive dependencies and created tables as needed to eliminate update and delete anomalies.

Group Members Responsibility & Contribution

Shafiq: contributed in ER design, coding the website and integrating the SQL database.
Mike: contributed in ER design, coding the website, and testing the web server.
Christopher: contributed in ER design, testing the web server and documenting the project.
Adriana: contributed in ER design, testing the web server and documenting the project.

Detailed Analysis of Coding the Website

A few of our files will be analyzed below in order to give a better understanding of our code:

login.php

Implementation overview: This is the login page where a user will add their username and password. If the individual does not enter their username or password, the same page will reload to prompt the user to add the required information. The same algorithm is applied for incorrect usernames or passwords. Overall in the login.php file, conditional algorithms are used (if/else statements) as the file must have a decision made between the two actions.

Variables

$username = $passwords =” ”

-The variables username and password are defined and initialized with an empty string.

Algorithms

if (isset($_POST['login_user']))

-$username: user adds their username. The '$_POST' is used to post the username into the database.

-$password: user adds their password. The '$_POST' is used to post the password into the database.

-if (empty($username)) & if (empty($password)):

-Sub conditional statements were made within this conditional statement to verify if the username and password were entered.

logout.php

Implementation overview: The logout.php file will destroy the current session and redirect the user back to the login page.

session_start(); is a function used to start and initialize the session.
$_SESSION = array(); is a function used to unset all of the session variables
session_destroy(); is a function used to destroy the session
header("location: login.php"); is used for the user to be redirected to the login page

The Interface Design Rationale

As a group, we decided to keep the same interface as the current CrsMgr website. The purpose of this was to allocate more effort and focus on the backend of the project rather than the frontend. Other functionality and web pages were created and tuned according to the required documentation. The style.css file is found in the styles folder.

Queries

All of our queries are role driven. They are dynamically generated, and prevent SQL injection by implementing parameterized SQL statements. All of our queries perform one of 4 actions: SELECT/INSERT/UPDATE/DELETE. Below is an example of code from our announcements.php page.

```
if (isAdmin()) {
    $query = "SELECT a.*, u.username, c.course_name FROM announcement as a
    JOIN users as u ON a.posted_by_uid = u.user_id
    JOIN course as c ON c.course_id = a.course_id
    ORDER BY announcement_id ASC";
} else {
    $query = "SELECT a.*, u.username, c.course_name FROM announcement as a
    JOIN users as u ON a.posted_by_uid = u.user_id
    JOIN course as c ON c.course_id = a.course_id
    JOIN user_course_section as ucs ON ucs.course_id = c.course_id
    JOIN users as us ON us.user_id = ucs.user_id
    WHERE us.user_id = $session_user_id
    ORDER BY announcement_id ASC";
}
```

User Manual

In this section we will discuss in detail the overview of each user role along with all their specific functionalities. As a brief overview, the login page is where the user will login, and it is linked to the index file. Thus, if a user is not already logged in they will be automatically redirected to the main login page. If the user has forgotten their password, they may recover it using the forgot password option below to recover it using their linked email address. We will first begin with the most restricted interface (students) and make our way up to the most privileged interface (admin).

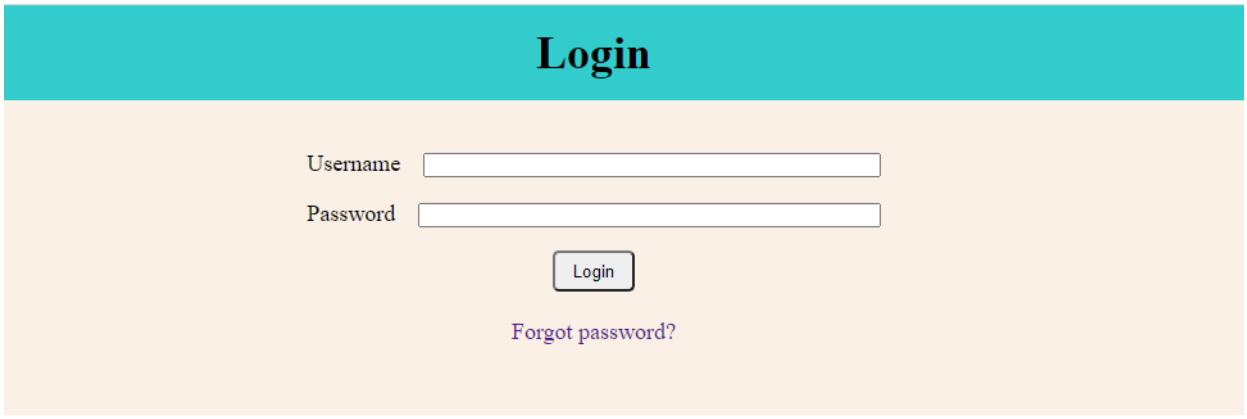


Figure 1. Main login page to access the CGA interface

Section 1 - Overview of Student User Mode

Students are the users with the most limitations on the CGA user interface. The student is able to login to the system with their credentials. When the student logs into their account, they will be directed to the home page where they will see general information. The home page for the student is similar to the professor and TA. The students home page consists of the courses the students are enrolled in along with their group information and announcements. On the top left corner, the student will see their role. The student has the ability to view the courses they are enrolled in as well as view their groups for their classes. From there they can manage the forum and tasks available for each class they are enrolled in. Lastly, the student has the ability to redirect to another page to reset their email, reset their password or simply log out of their account. The image below summarizes the home page overview for a student.

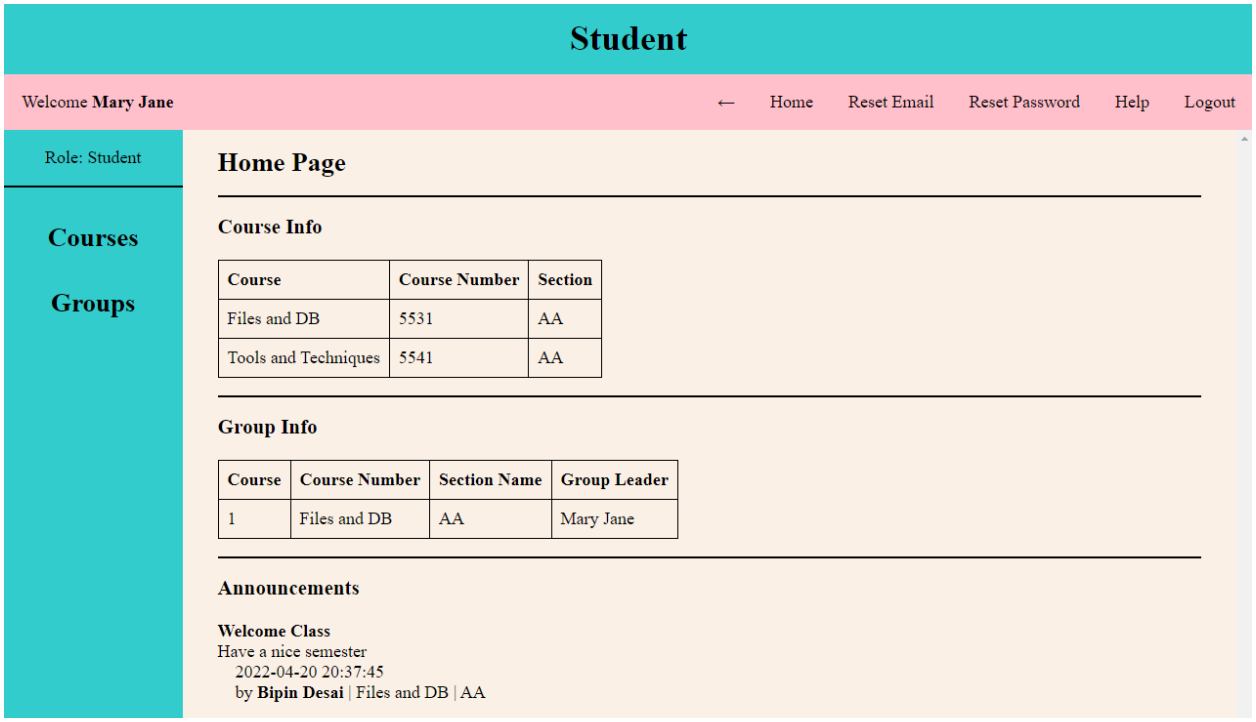


Figure 1.1 Overview of student homepage

1. Course Section

In the courses section, the student has a summary of the courses they are currently enrolled in as well as the recent forums posted by their fellow classmates. The functionalities for the students on this page is their ability to view and manage new forums as well as to view the tasks that are assigned to them.

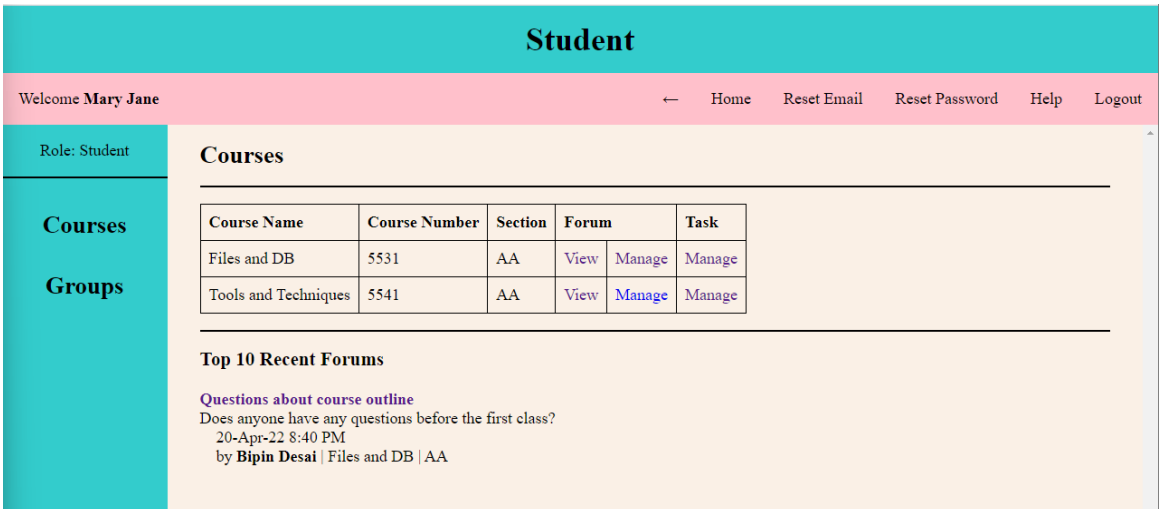


Figure 1.2. Overview of student course interface

Forum view & manage

Upon clicking on view, the student can view all the forums posted that has been posted with the course name respectively.

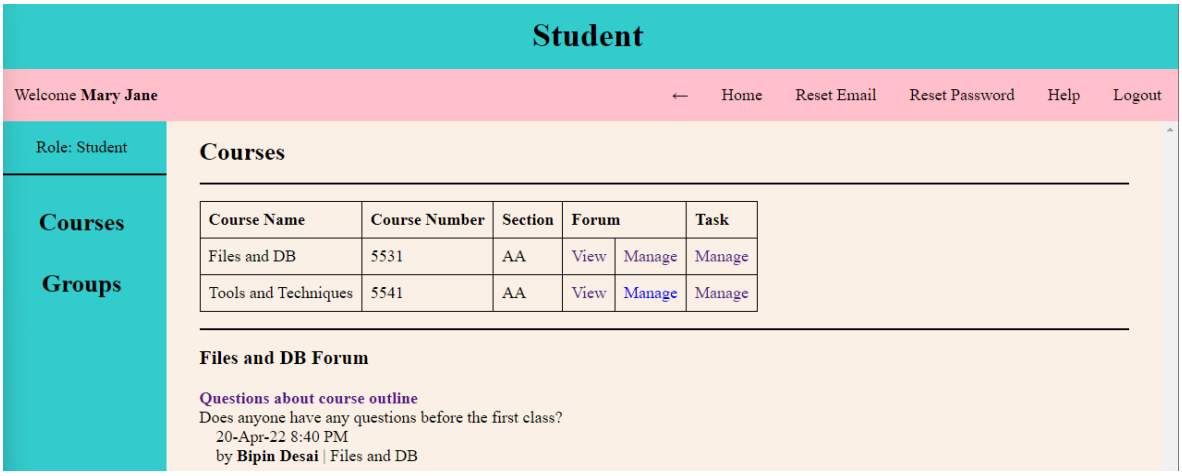


Figure 1.3 View courses forums

Furthermore, in the manage section for forums, the student can post a discussion as well as update/delete their comments once they are posted.

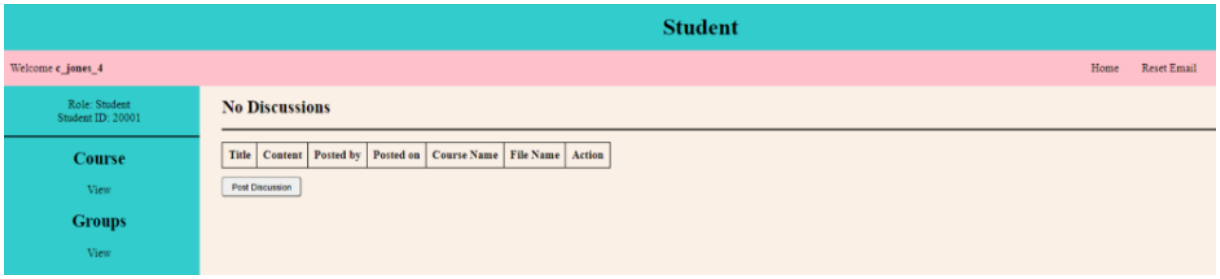


Figure 1.4 Overview of student discussions

Tasks

Upon clicking on the “manage” functionality for tasks, the student will be able to view the tasks assigned to him. Many different attributes are used to describe the tasks. These include the type of task, whom the task was uploaded by, the date the task was uploaded on, the name of the file, the solution as well as the action that can be taken.

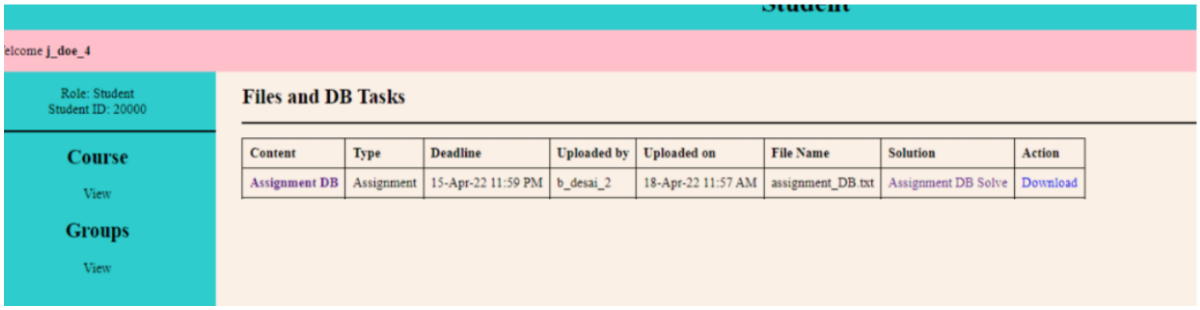


Figure 1.5 Overview of student tasks

2. Groups Section

The groups consist of similar functionalities as those from the course section. The student is able to see the groups they are currently in. The information they see are the name of their group, the name of their group leader, what is their section number, for what course this group is for as well as all the members that are in the group.

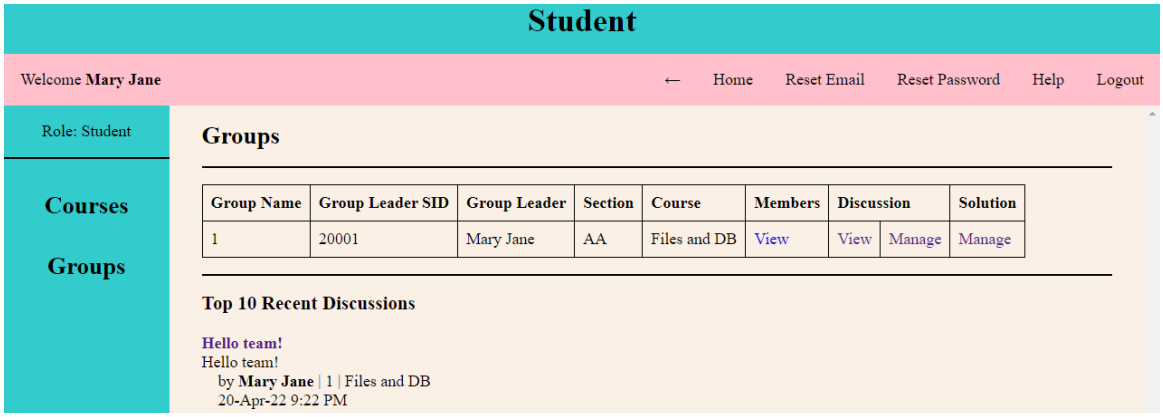


Figure 1.6 Overview of student groups interface

Groups - Solutions

In the solutions section, the student is able to see what they have submitted. The information the student can see is the name of the assignment they have uploaded, by whom it was posted, the date it was posted, etc. The student can also download, update or delete the file.

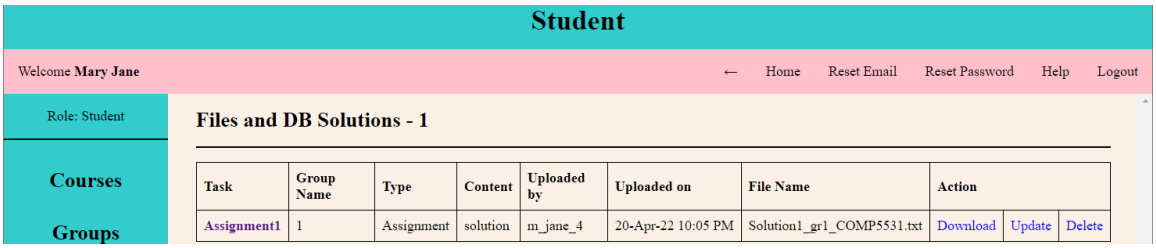


Figure 1.7 Overview of submission files page

Inbox

In the inbox section, students are able to send a message to other students. Students may also add attachments along with their messages.

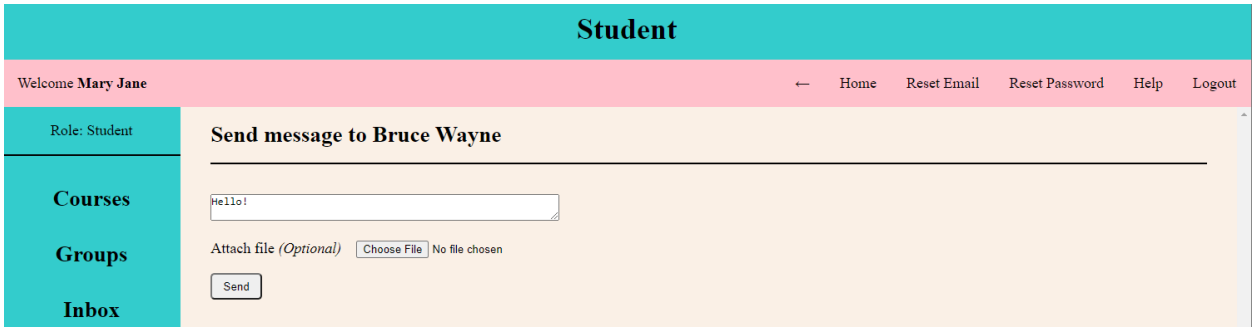


Figure 1.8 Overview of the student inbox page

Section 2 - Overview of Teaching Assistant User Mode

Teaching assistants are able to log into the system with their credentials. Once they are logged in, they will be redirected to their homepage. Similarly to the students interface, TAs have two directories where they can click and view the courses they are TAs for, as well as view the groups and contribute to group discussions. Details about the functionalities will be described below.

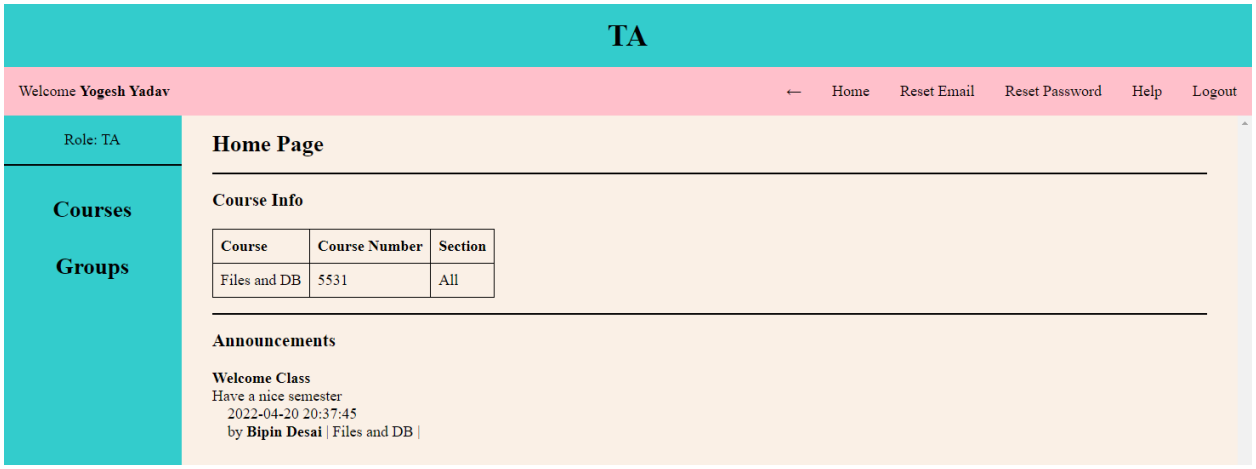


Figure 2. Overview of TA home page

1. Course Section

In the course section, the TA can view courses and interact with both students or professors in the class forum. The TAs can add and post new forums or reply in existing ones where he can update or delete his comments. The TA may also view the assignments that the professor has posted.

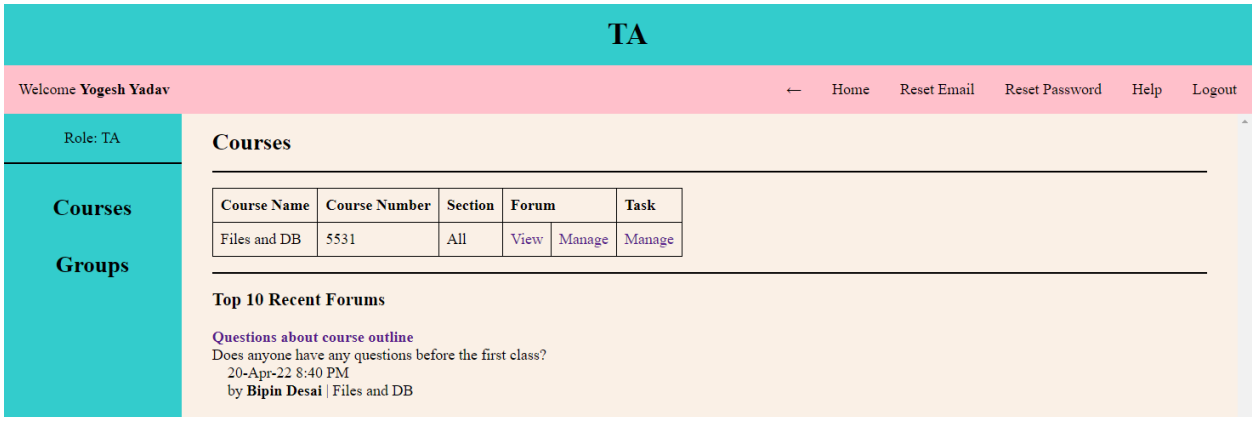


Figure 2.1. Overview of TA course interface

Forums

The TA can reply and post new forums as shown in the following two figures below.

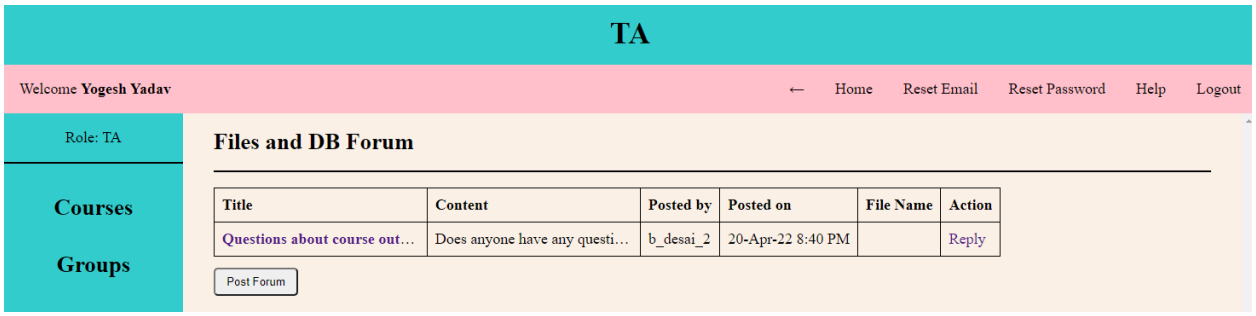


Figure 2.2. Functionality where TA can reply and post new forums

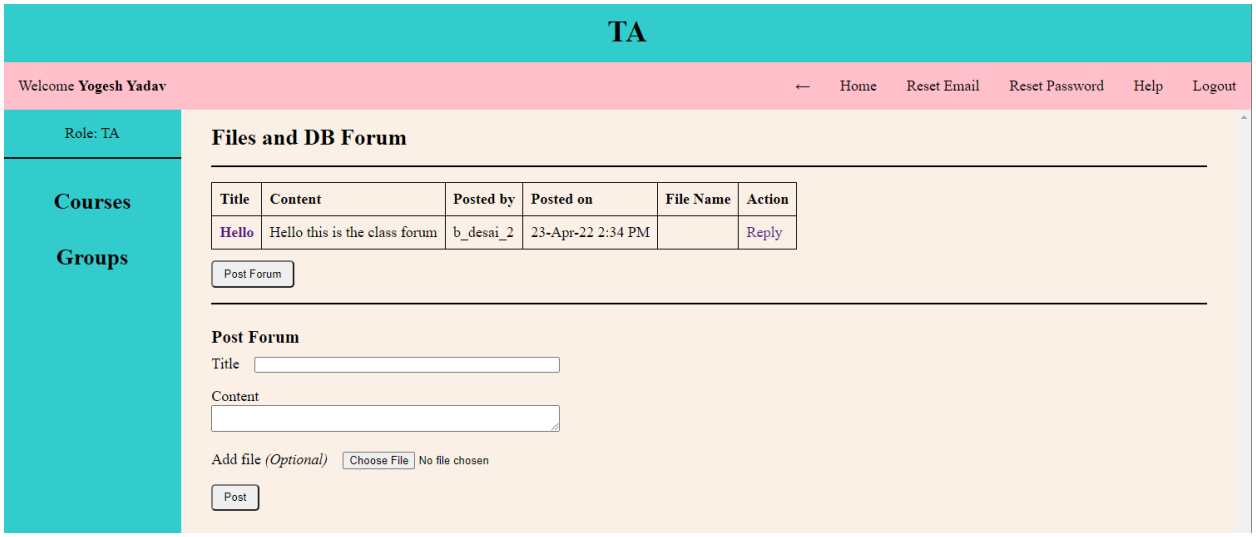


Figure 2.3. Functionality where TA can add and post new forums or reply in existing ones

2. Groups Section

The TA can contribute to the group's discussions by commenting on them. However, the TAs do not have the ability to create new threads and may also view the solutions posted by the group.

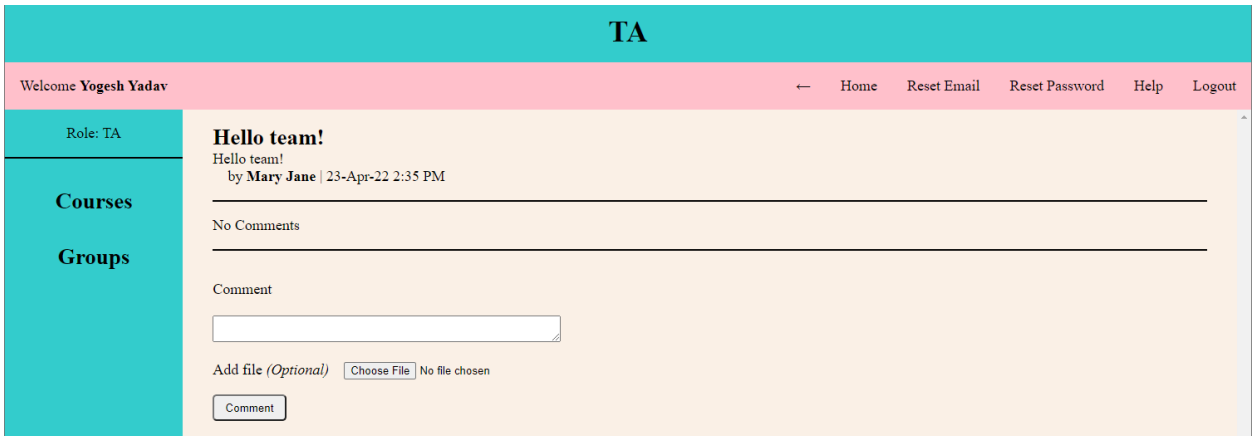


Figure 2.4. Overview of TA contributing to group discussions

Section 3 - Overview of Professor User Mode

Professors have a larger variety in terms of their capabilities and roles within the interface. The professor has many more privileges in the system in comparison to teaching assistants and students. On the professor's home page, they are able to see the courses they are currently teaching along with all the announcements related to their courses. Their roles consist of but are not limited to managing forums, managing courses, tasks, discussions, managing groups, managing announcements, etc. The professor's specific functionalities will be described in detail below.

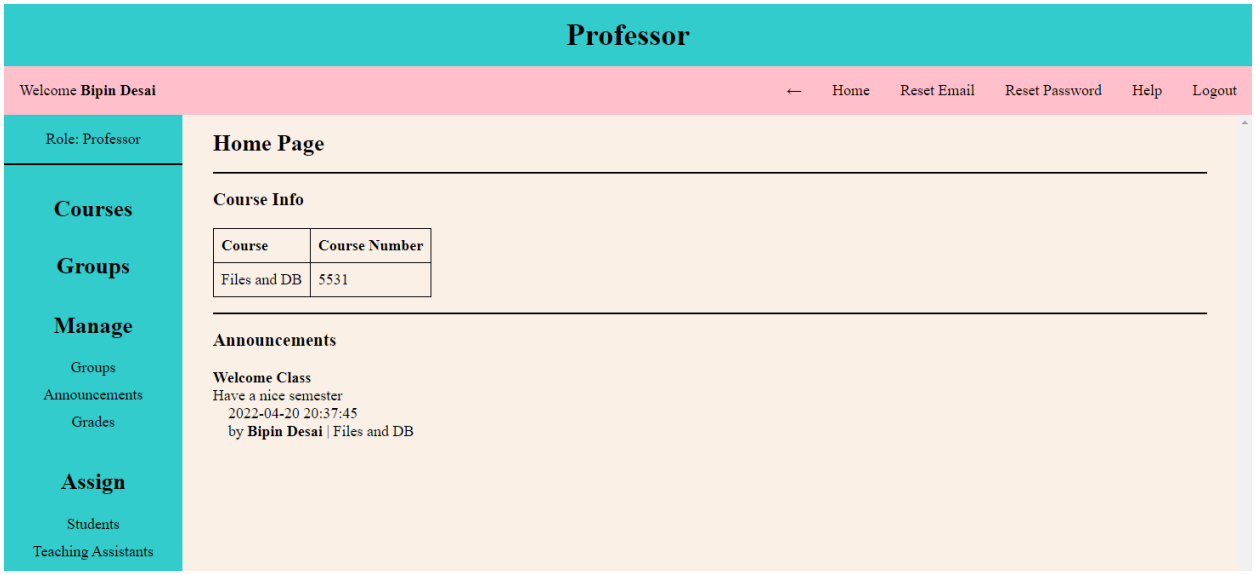


Figure 3. Overview of professor home page

1. Course Section

Forums

In the course section, the professors can view the courses they are teaching and from there they can view and manage the forums which is a posting across every class.

Professors can start new forums by adding a title and some content. Afterwards, they can update its content or delete it altogether.

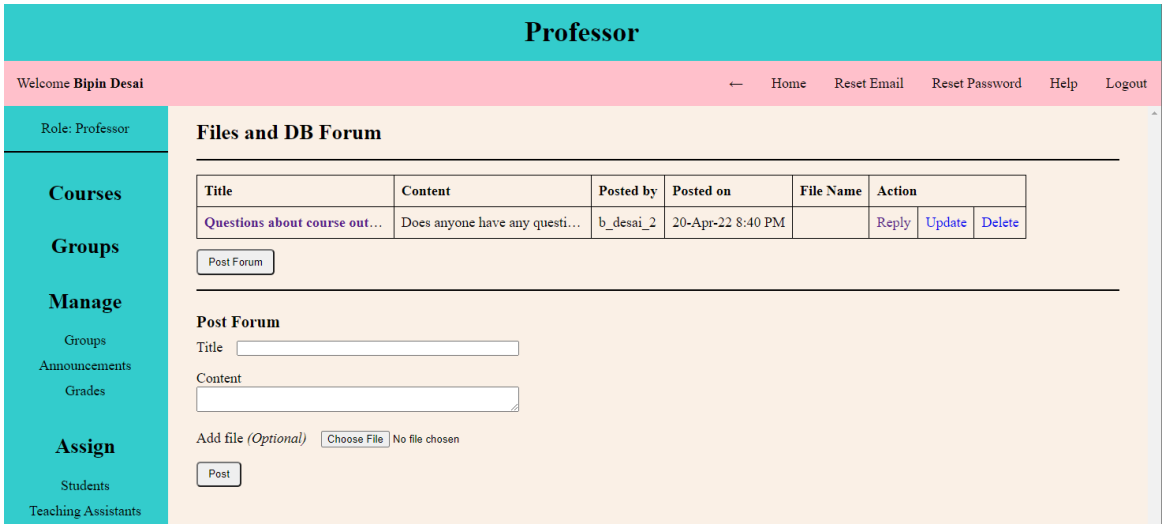


Figure 3.1 Functionality where the professor can view, manage and edit content of forums

Tasks

In tasks, the professors can upload a new assignment. The professor can manage different attributes such as add a deadline as well as add/delete the assignment.

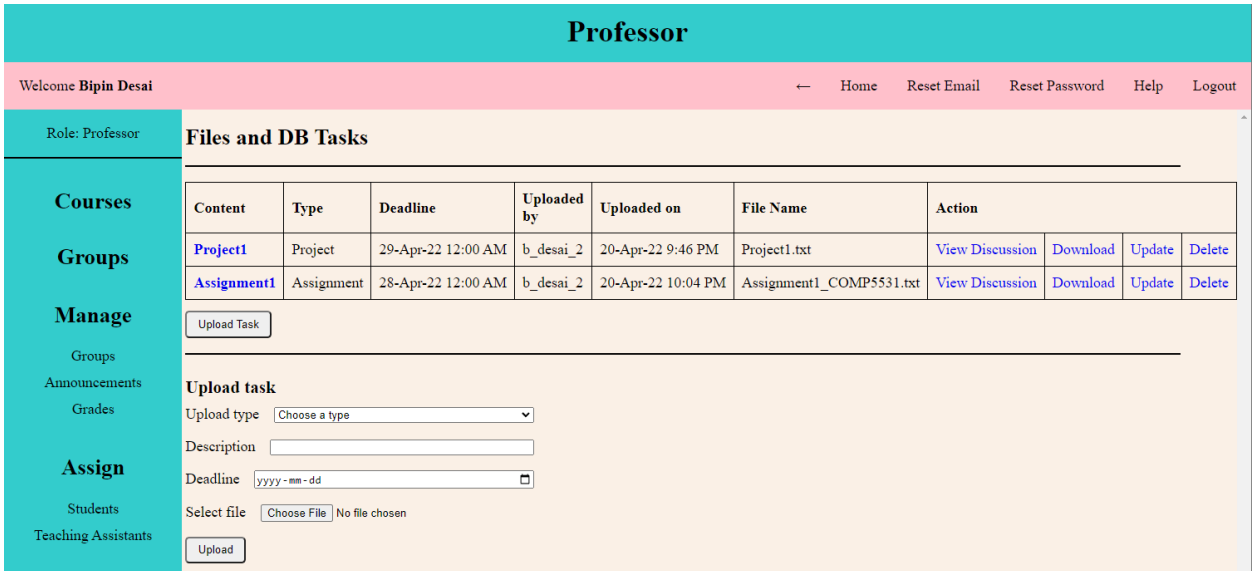


Figure 3.2 Functionality where the professor can upload/manage attributes for the assignments

2. Groups Section

In groups, the professor has the same discussion functionality however it is group specific. The professor can also view the solutions from the students and grade them. The students can update or delete the grades under the managed grades.

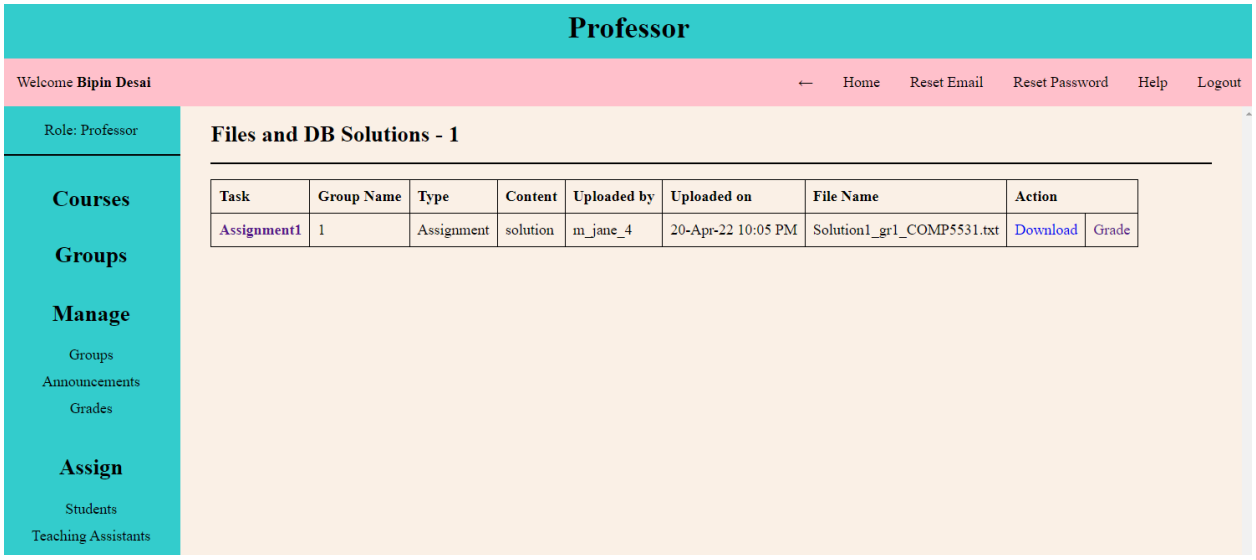


Figure 3.3 Functionality where the professor can view solutions

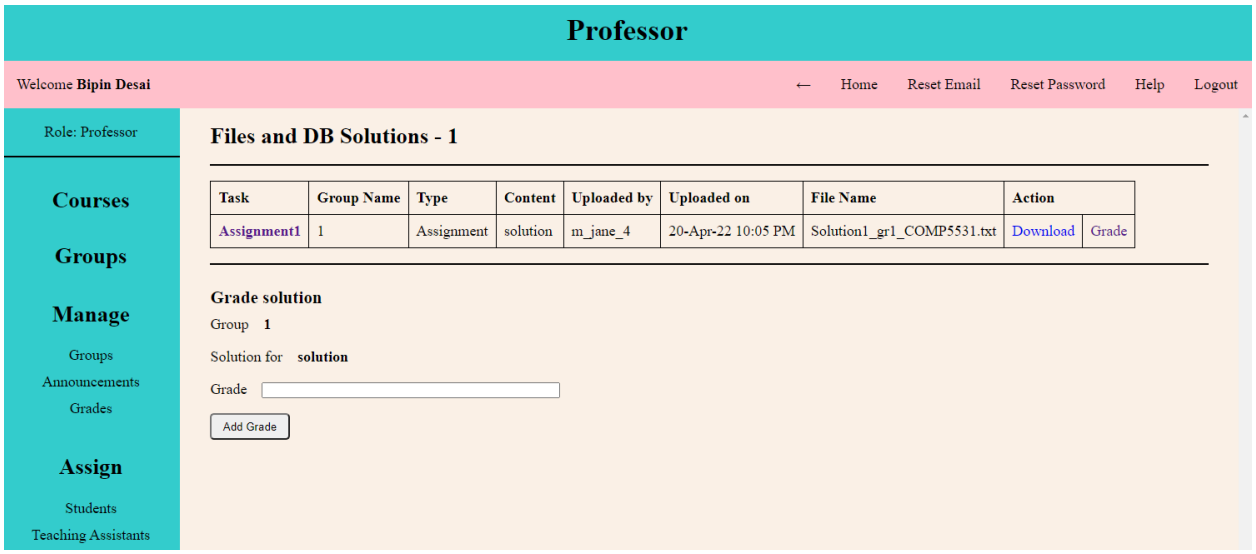


Figure 3.4 Functionality where the professor can grade

3. Manage Section

In the manage section, professors can manage student groups, announcements as well as students grades.

Groups

The professors can manage groups where he can update or delete the name and the group leader. Furthermore, the professor has the ability to add/delete/change the group members.

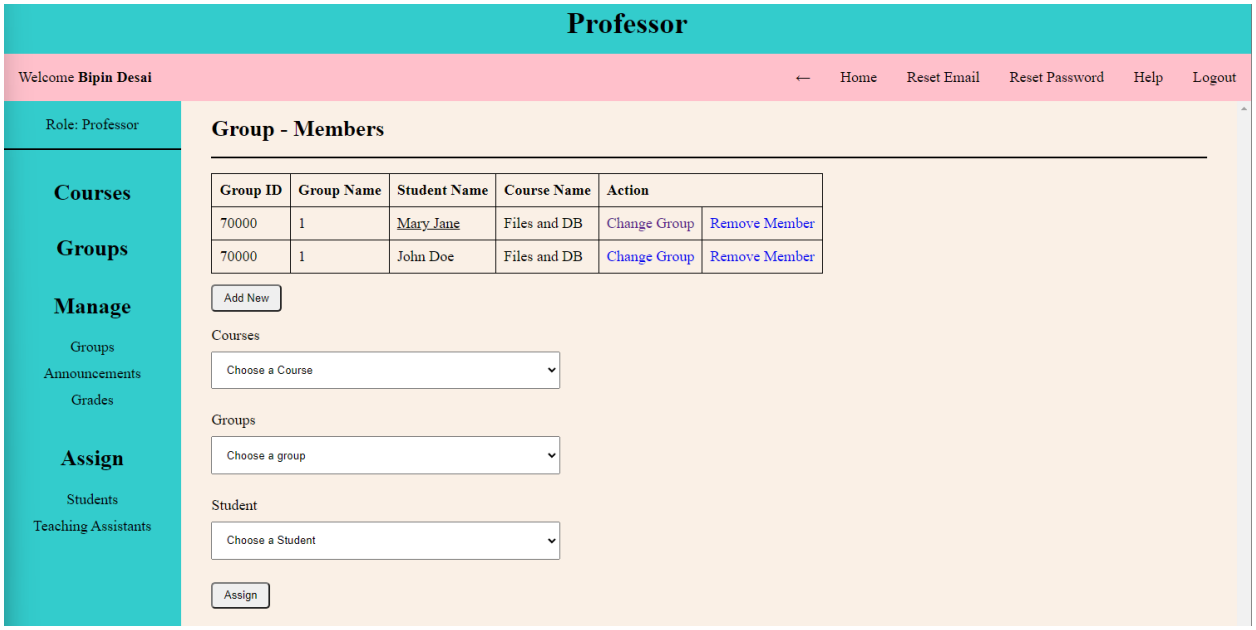


Figure 3.5 Functionality where the professor manage student groups

Announcements

In the announcements page, the professor can add announcements to the class and also update or delete them.

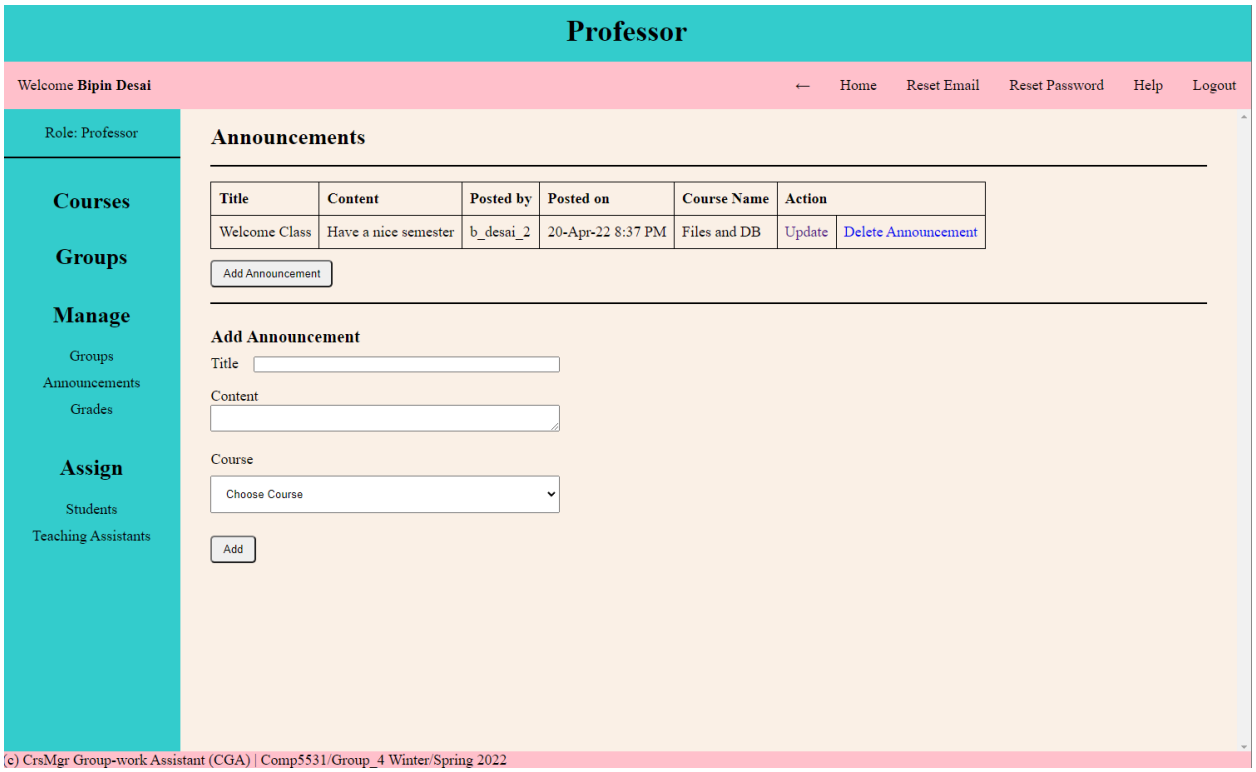


Figure 3.6 Functionality where the professor add/edit announcements

Grades

The professors can update or delete individual grades, even though group assignments transfer the overall to each student in the group.

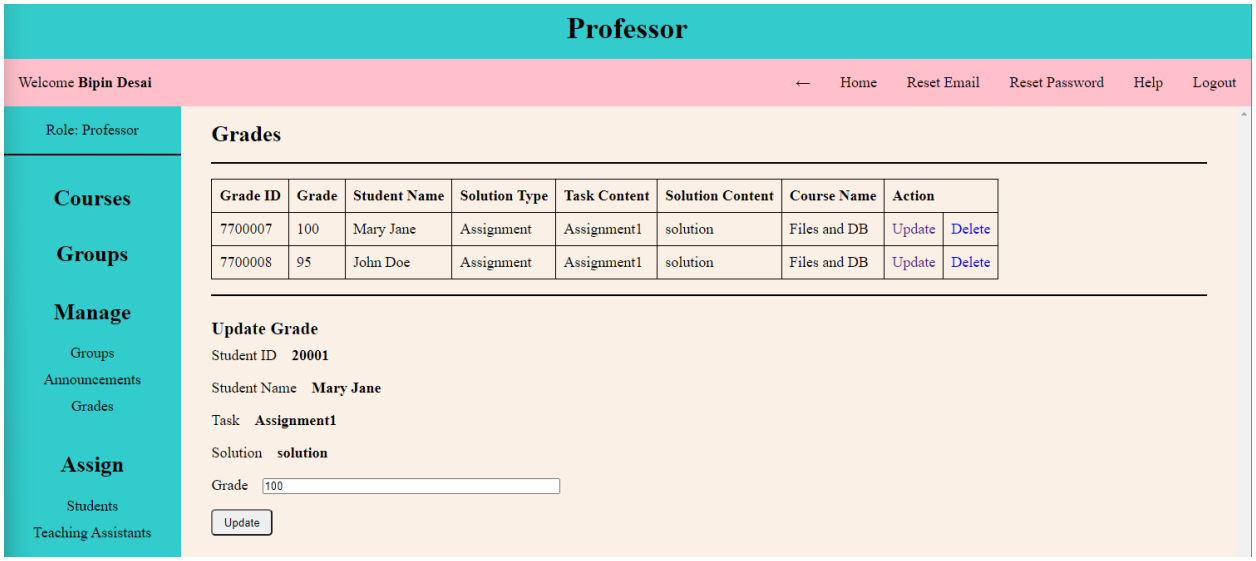


Figure 3.7 Professor individual grade edit/delete

Assign Section

Professors can also assign students and teaching assistants to their classes and to specific sections.

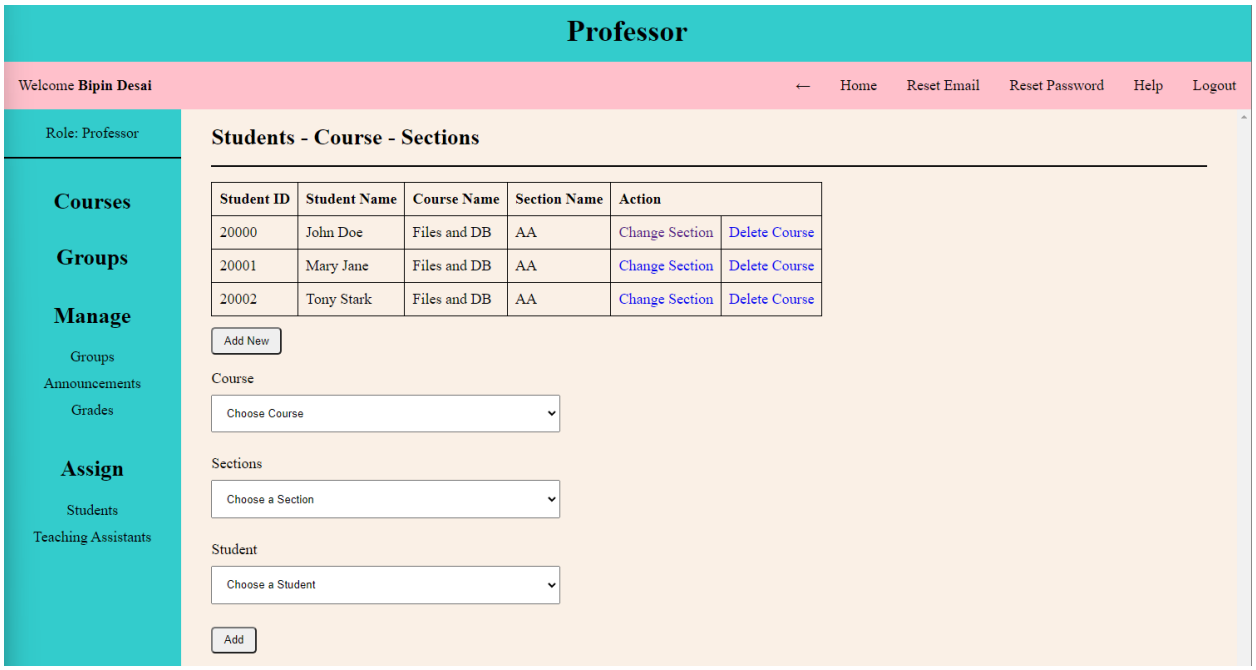


Figure 3.8 Professor assign students

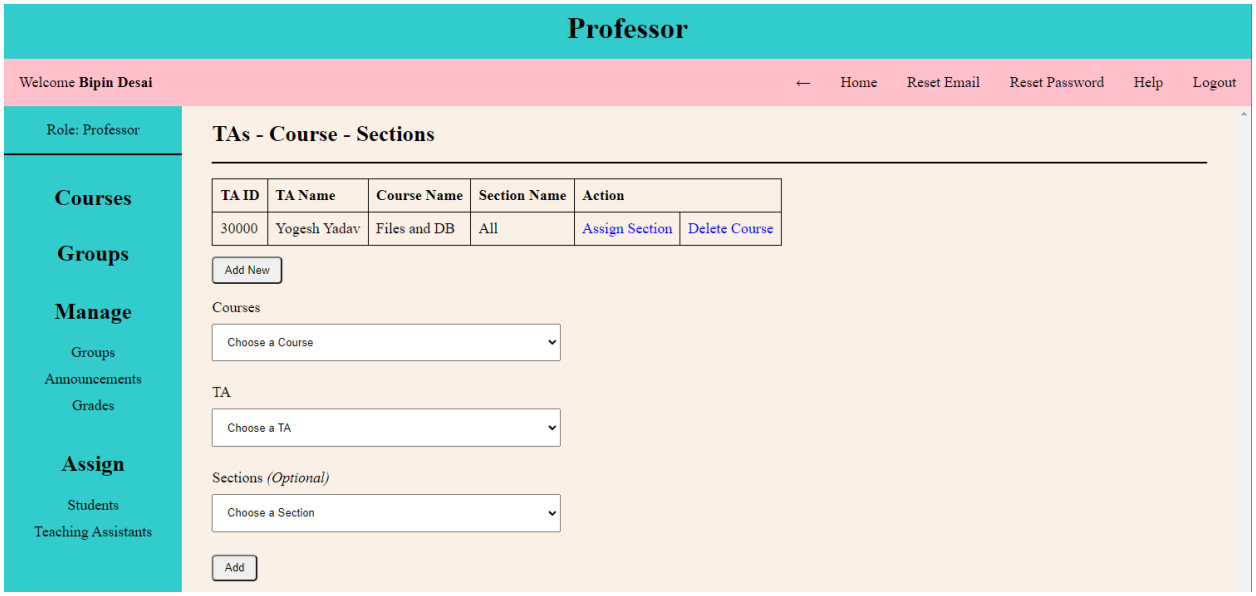


Figure 3.9 Professor assign teaching assistants

Section 4 - Overview of Admin User Role

The admin is the most powerful role in the CGA system. The admin has the privilege and the entitlement to add or modify whatever they would like in the system. The admin can click on any of the components on the sidebar to go and tweak the system which is directly linked to the database. The home page of the admin is as follows where all the functionalities are shown.

Figure 4. Overview of the admin home page

1. Manage

The admin can manage users, roles, courses, course sections, groups, files, announcements, forum, reply, discussions, comments, tasks, solutions and grades. The specific functionality of each will be described below.

Users

The admin can view/add/update/delete all of the users existing in the CGA system database. The admin can change any information related to the user simply by going to the “update” page. However, the admin is not able to change their roles.

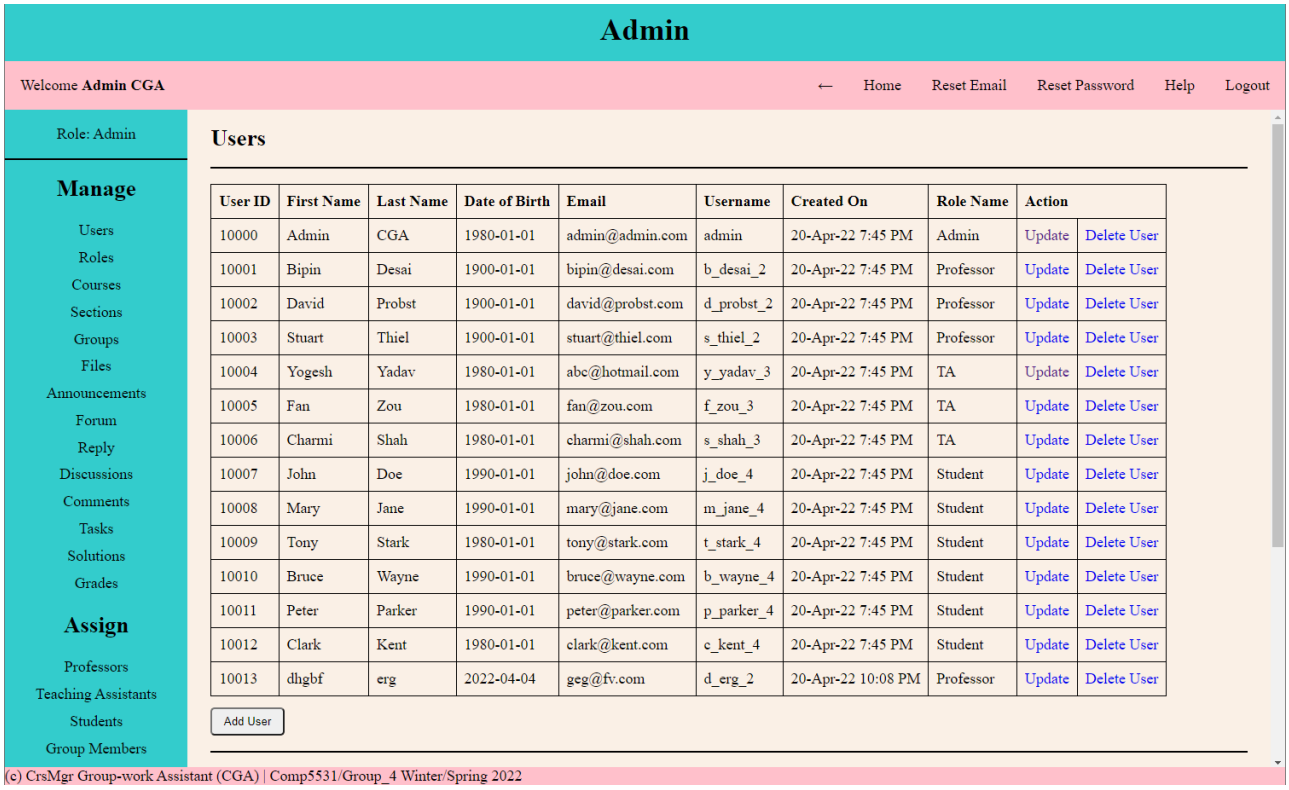


Figure 4.1 Admin users page

Roles

The admin has the privilege to view/add/update/delete all of the different roles existing in the database.

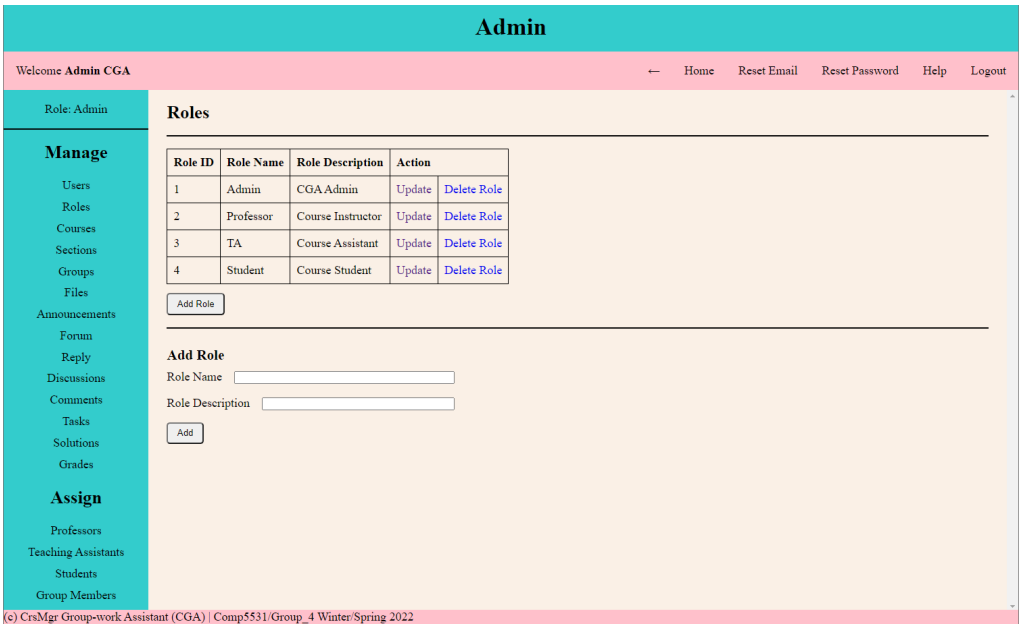


Figure 4.2 Admin roles page

Courses

When the admin clicks on the courses tab, they will be redirected to a page where they may update, delete or add a specific course. The attributes the admin will need are the course ID, course name and the course number.

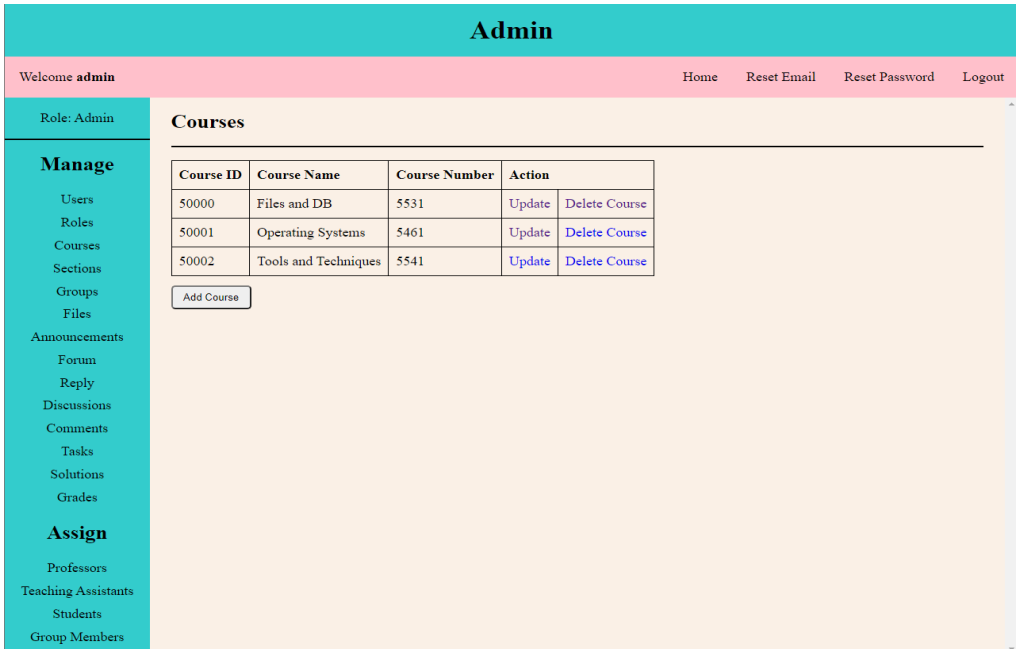


Figure 4.3 Overview of the courses for the admin page

Sections

Similarly to the courses page, the admin can view/add/update/delete sections. When doing so, the admin has the ability to add information. These information includes section ID, section name and course name.

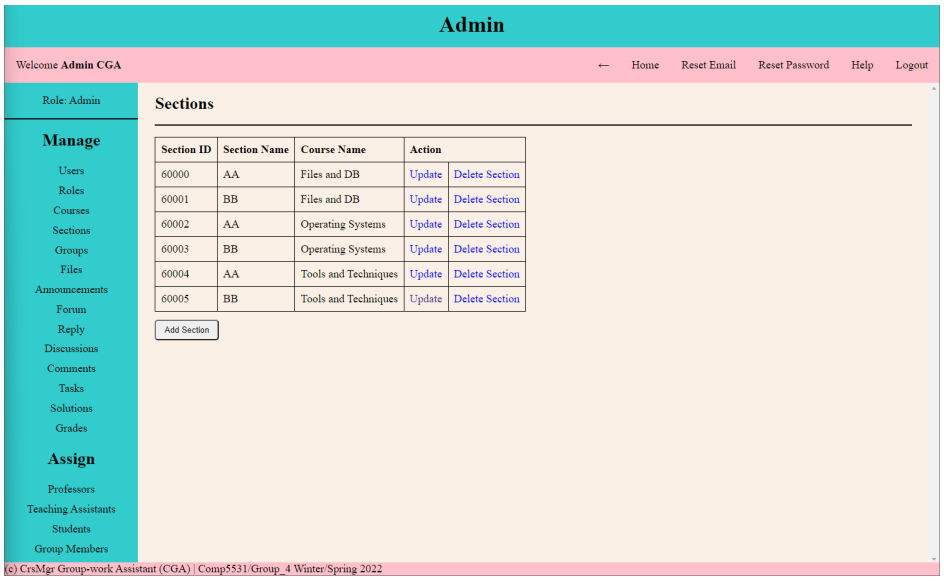


Figure 4.4 Overview of the courses for the admin page

Groups

The admin has the ability to add groups to specific classes as well as assign a group leader.

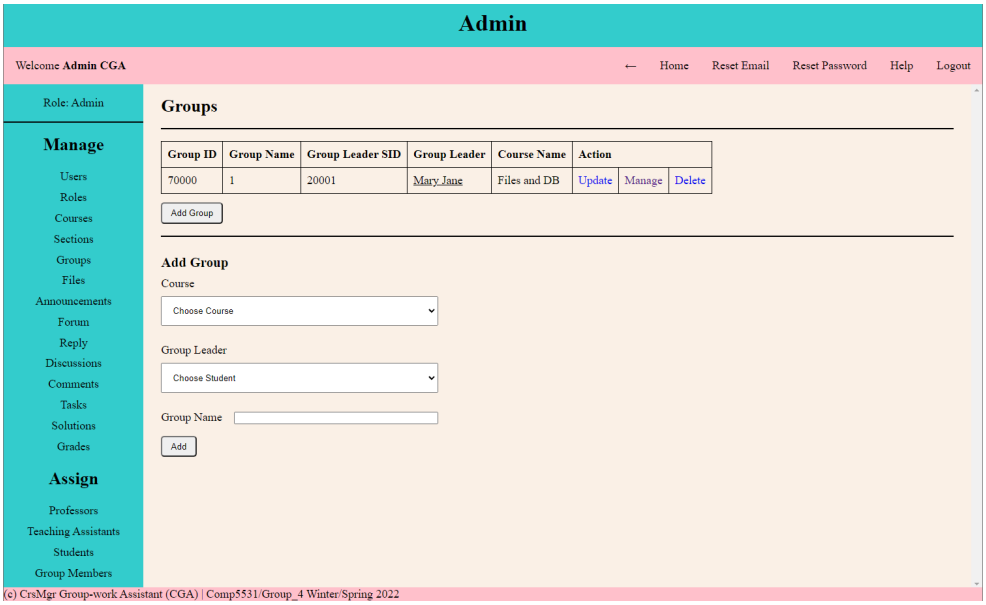


Figure 4.5 Overview of the groups for the admin page

Files

The admin can view all files that are stored in the database. It has info on file ID, the name, the content description, the file type, its size, who it was uploaded by and on what date and time it was uploaded. The admin can view/change/delete the file.

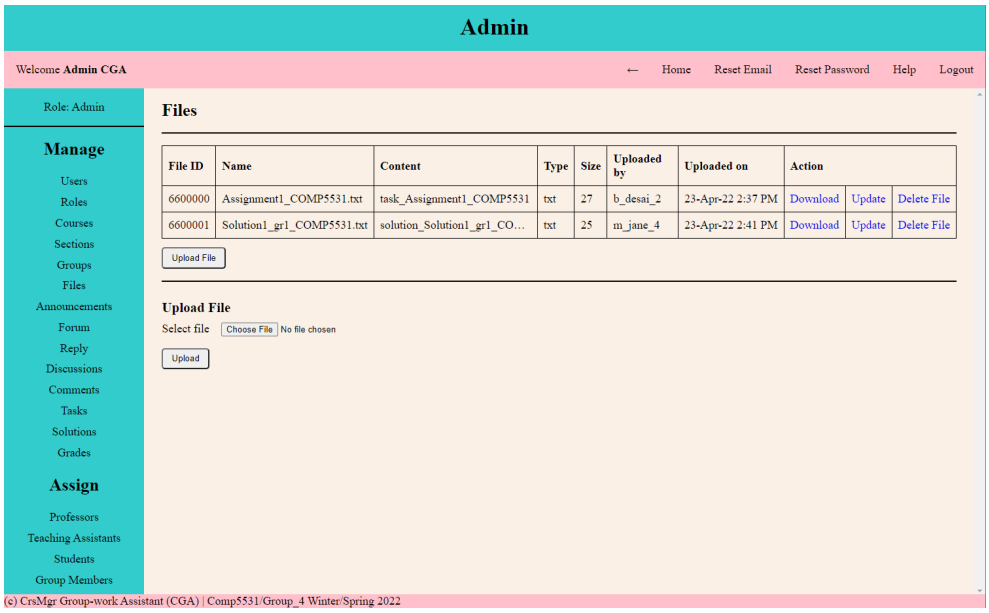


Figure 4.6 Overview of the files for the admin page

Announcements

The admin can add, update and delete the announcements. The admin can also do a global announcement to all users in a specific course. The admin would select do all courses for all users.

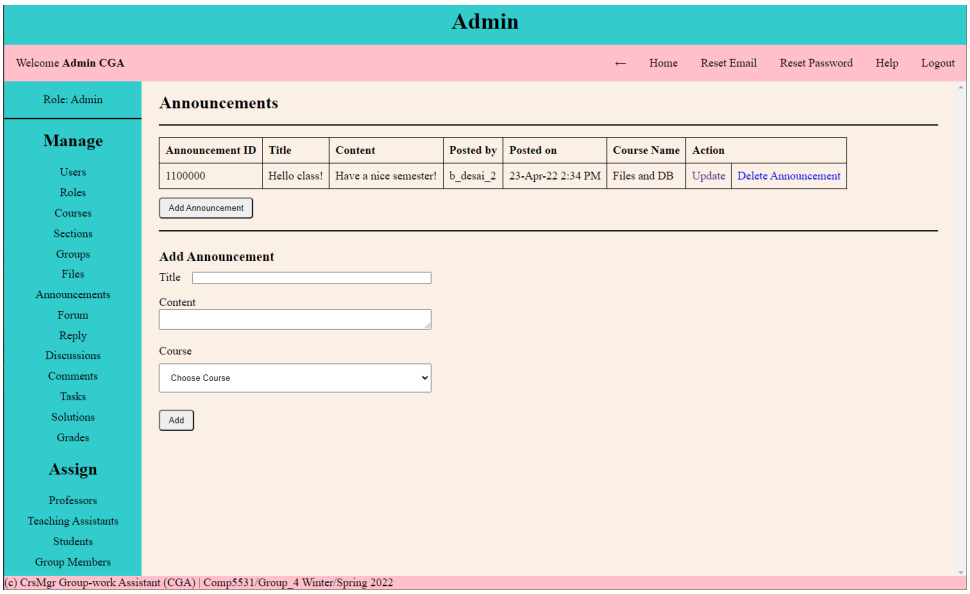


Figure 4.7 Overview of the announcements for the admin page

Forum

In the forums, the admin can only view and delete the forum. The admin is able to see the entire forum content as well as by whom it was posted, the date it was posted and the from which course it was posted for.

Figure 4.8 Overview of the forums for the admin page

Reply

The admin is able to view all the replies posted. The details the admin is able to see are the content, whom it was posted by, when it was posted, and for which course. The admin has also the ability to remove a reply.

Figure 4.9 Overview of the replies for the admin page

Discussions

The admin is able to view all the discussions that have occurred throughout the interface. The admin will be able to see the title of the discussion, whom it was posted by, when it was posted etc. The admin is not able to physically modify the content from another user, however the admin can indeed delete the discussion.

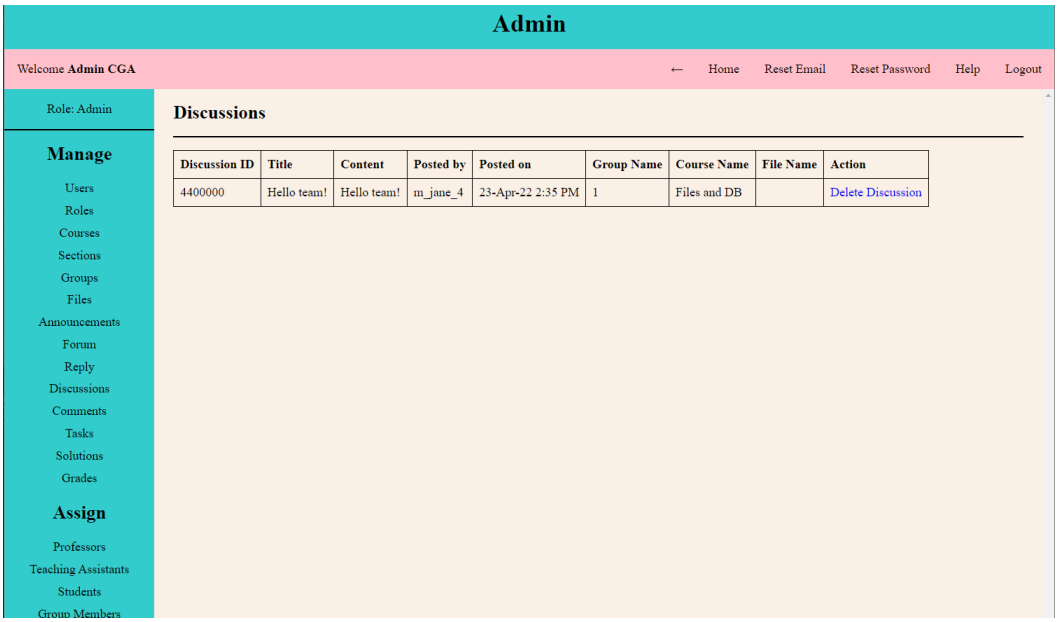


Figure 4.10 Overview of the discussions for the admin page

Comments

The admin may view all the comments posted by different users. Although the admin cannot modify a comment from the interface, they are able to at least delete comments.

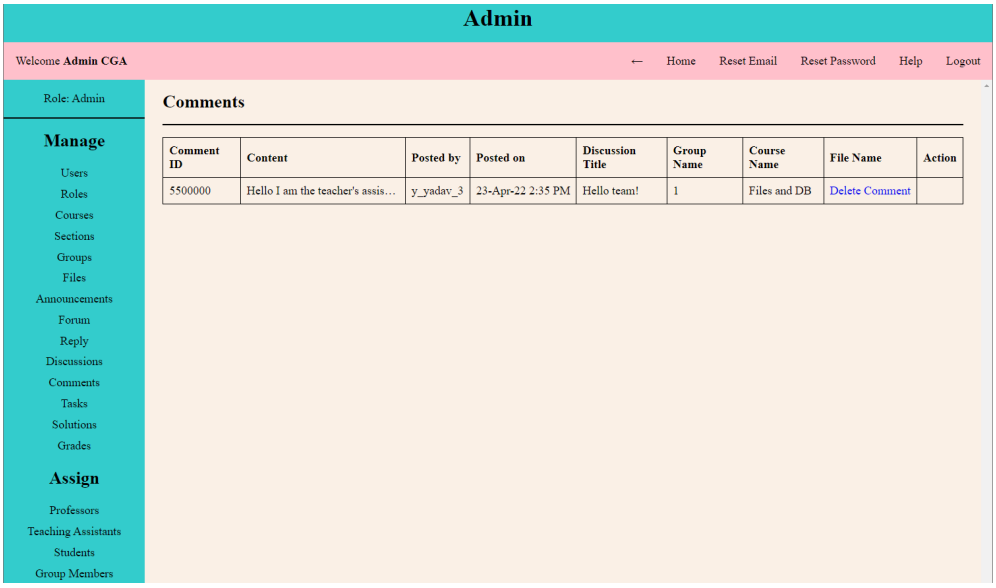


Figure 4.11 Overview of the comments for the admin page

Tasks

The admin is able to manage all the tasks that appear on this page. The admin is able to see what type of tasks there are along with their content, whom it was uploaded by, when it was uploaded as well as for which course it is for.

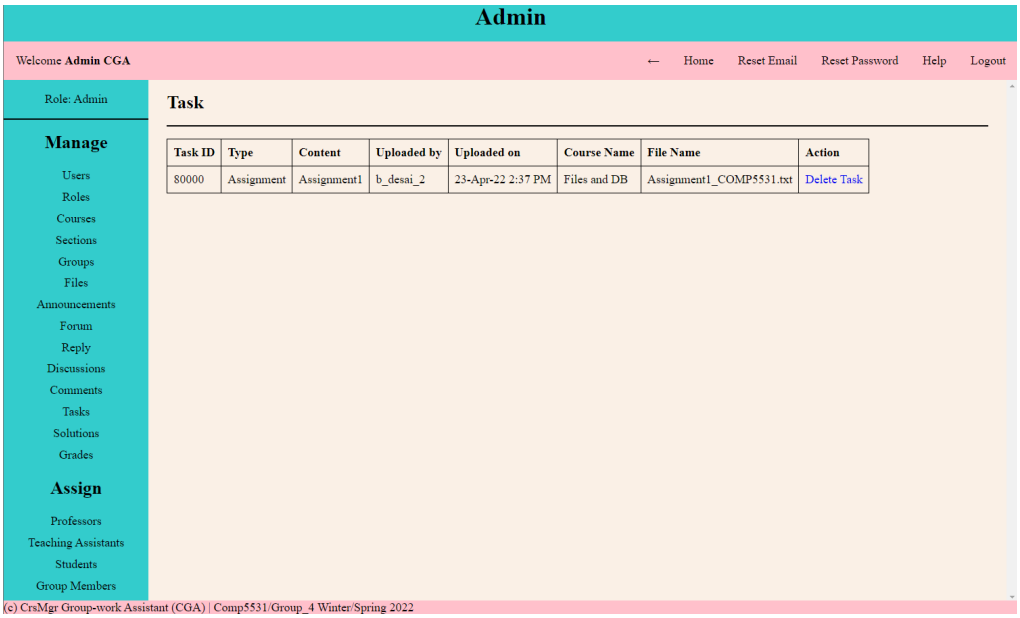


Figure 4.12 Overview of the tasks for the admin page

Solutions

Similarly to all the other tabs, the admin is able to see all the solutions that are posted and from which user. If needed, the admin is also able to remove it.

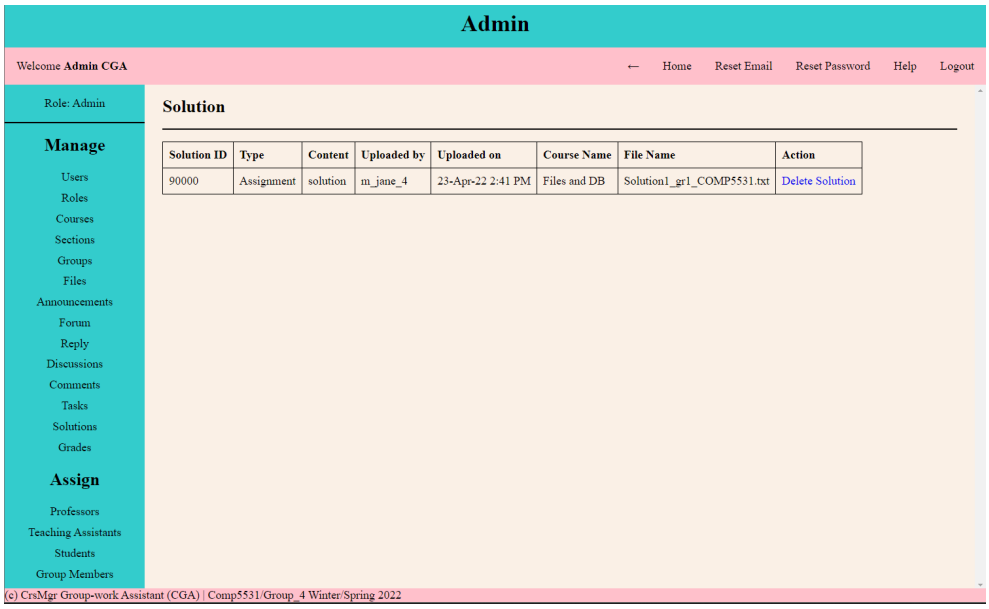


Figure 4.13 Overview of the solutions for the admin page

Grades

The admin is able to see the grades assigned to all students. Furthermore, the admin may also update and or delete the grades of students.

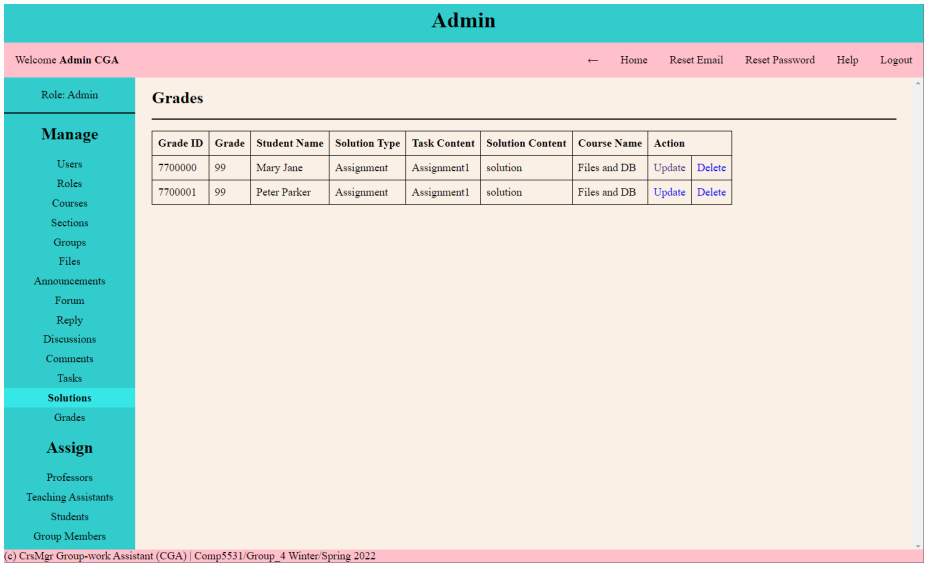


Figure 4.14 Overview of the grades for the admin page

2. Assign

The admin has the ability to assign professors, TAs, students and group members.

Professors

The admin is able to assign professors to a course. The admin simply would need to select the course as well as a professor to the designated sections.

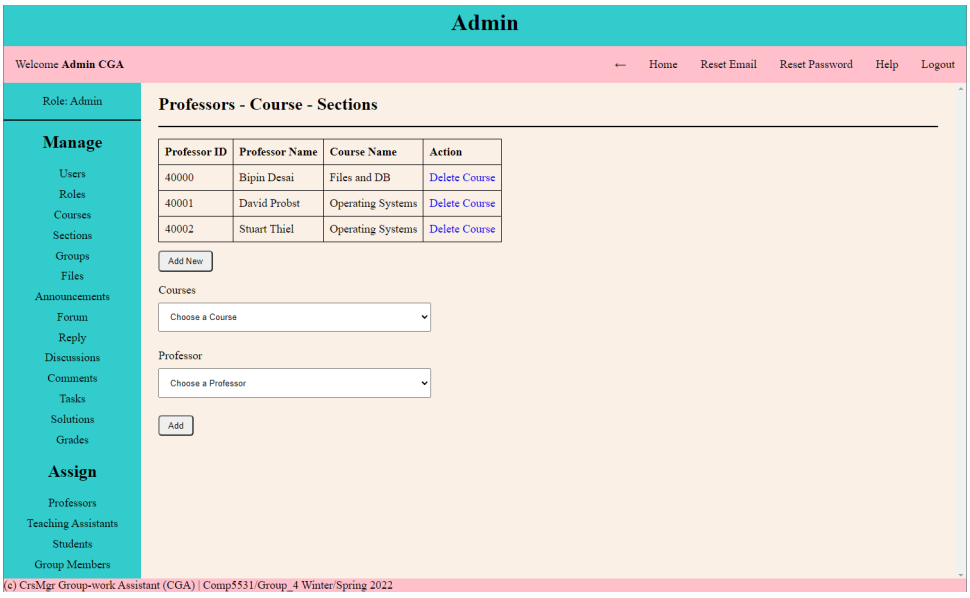


Figure 4.15 Overview of assign professors for the admin page

Teaching Assistants

The admin is able to assign professors to a course and a section (if any). The admin also has the ability to remove a TA from a course section and/or add a new TA to a course.

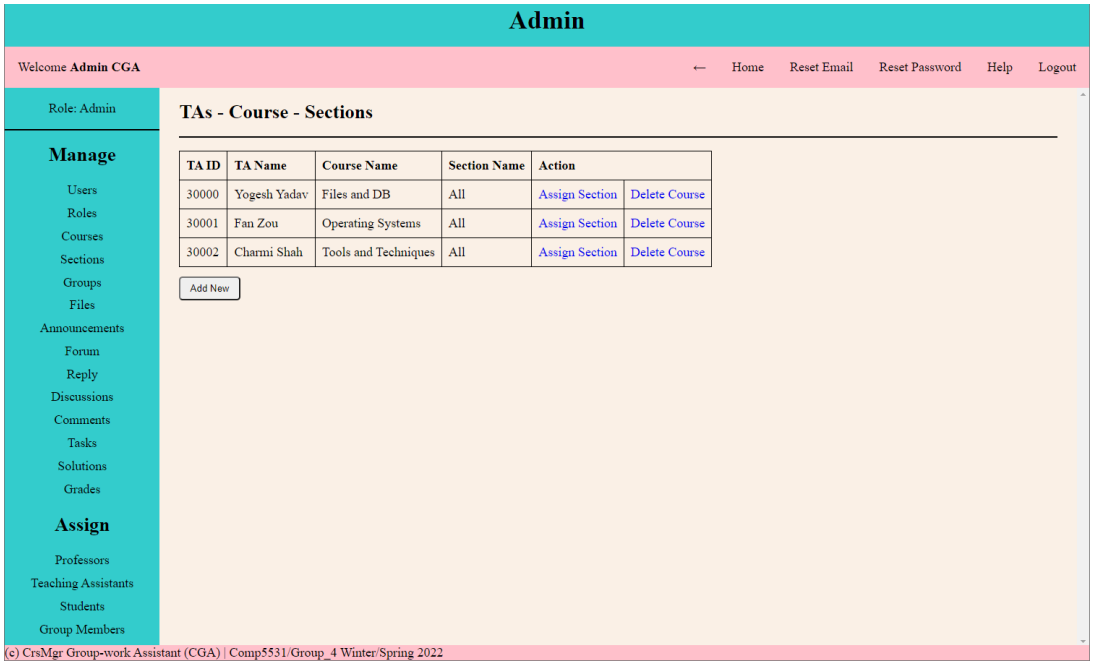


Figure 4.16 Overview of assign teaching assistants for the admin page

Students

The admin has the ability to change a students section and or delete a student from a course. The admin may also add a new student to a course and a section.

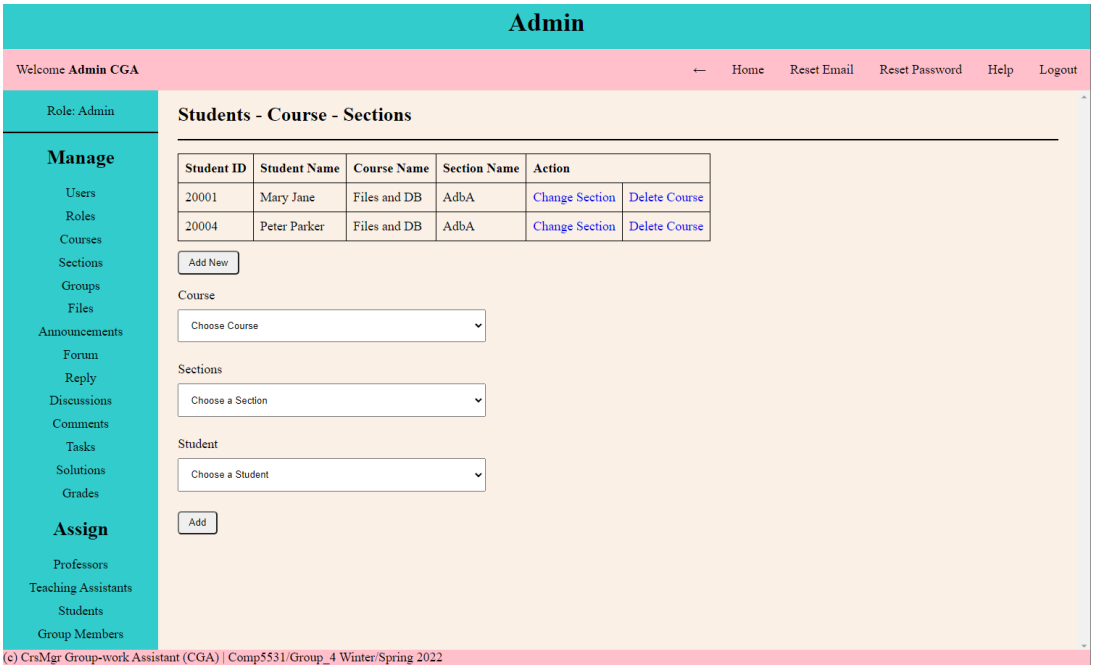


Figure 4.17 Overview of assign students for the admin page

Group Members

In this section, the admin can add a student to a group. The admin will simply fill in the information such as courses, choose a group and select a student to assign to a course.

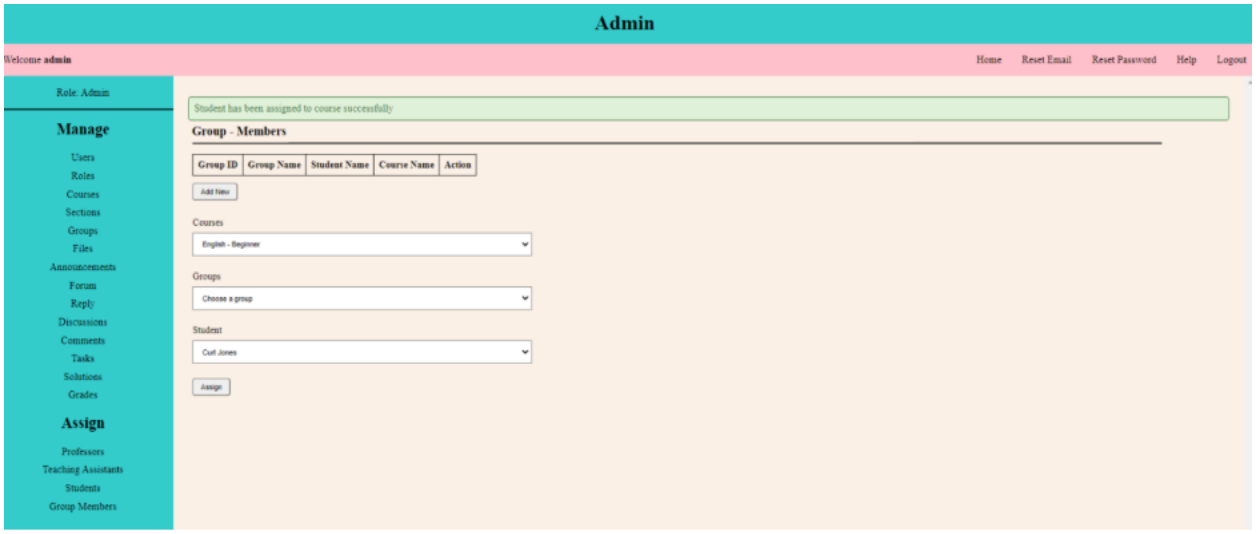


Figure 4.18 Overview of assign students for the admin page

Open Source Tools Used

The following tools and versions were used for the construction of this project:
PHP 12.0.0, MySQL version 8.0.28, Apache 2.4 and SMTP.

Authorized use Permissions

As different users have different authorizations, please refer below in the overview of the user interface to see the permissions and functionalities each role has.

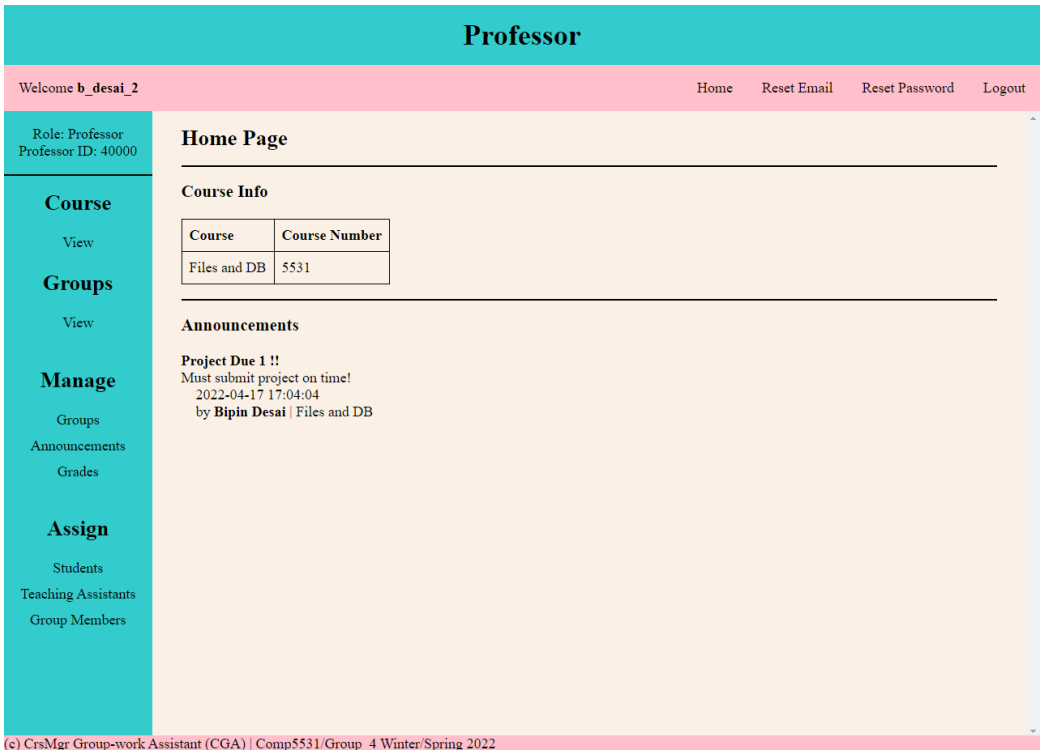
System Summary / Overview of the User Interface

There are four access roles all with different permissions. They are:

- Admin/Content-Creator: Acts as a super-user. Can add/delete/update roles, users, courses, sections, groups, files, and announcements. Can assign/delete/update professors, teaching assistants, students, and group members. Can delete forums, replies, discussions, tasks, solutions, and grades.
- Professor: Can view and manage courses. Can view/manage/update/delete groups. Can add/delete/update announcements. Can view/delete grades. Can add/delete/update students, teaching assistants, and group members.
- Teaching Assistant: Can view and add to courses, groups, and discussions.
- Student: Can view courses enrolled in, view groups, view and add discussions,comments and assignments.

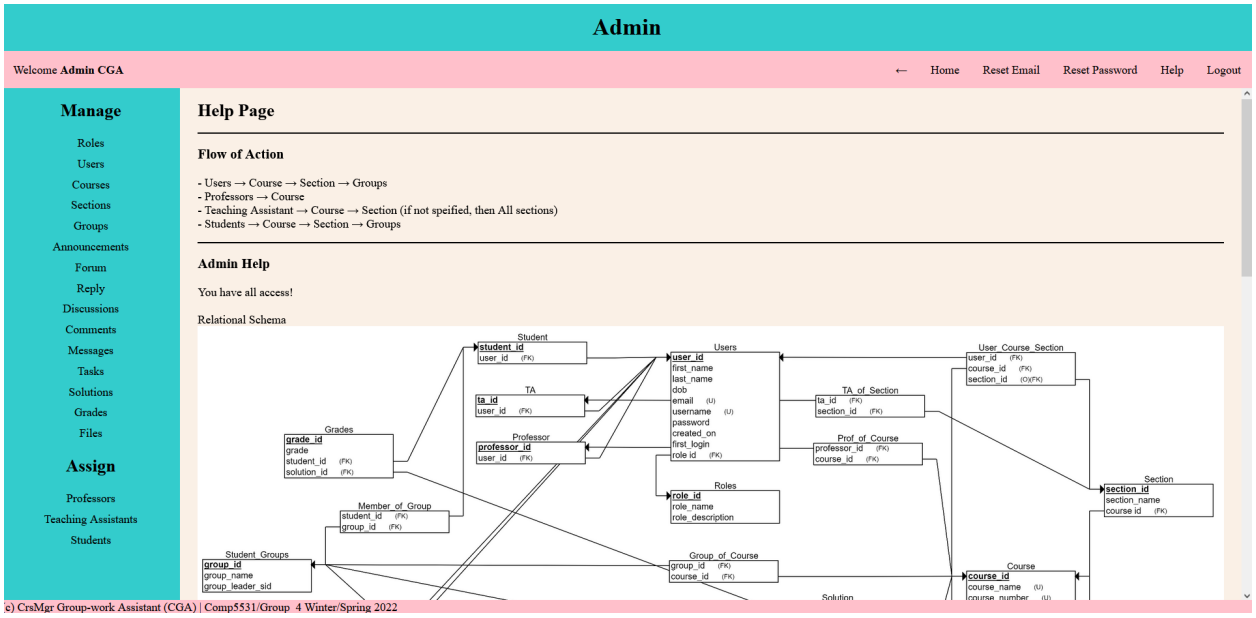
Home Page

The home page has the same structure for all of the roles. In the example below, we have the home page for a professor. The info for all courses assigned to that user and the announcements are shown in the main body. On the top banner, we can see the user's name, and buttons for: home, email reset, password reset, and logout. On the left banner, we have information of the user's role, its ID, and then buttons to the user's functionalities. Refer to *Overview of the User Interface* for all of the role's functionalities.



Help Page

Each user role has a help page. The help page consists of information on what the user can and should do.



Database

The database has been populated in such a way that all of our systems functionalities can be tested. Each table contains multiple records so the system acts as intended like in a real-world use case scenario. The database is split into two files mentioned in the section below to ensure all information is properly communicated.

Script to Create the Database and Populate it

There are two .sql files that are found in the SQL folder. The first file, create_db.sql is used to create the database and the tables. The second, data_db.sql is used to populate the database.This can be done simply by executing the scripts in MySQL Workbench.

References

1. Duff. W. (2019) *What is an SMTP Server?*. Twilio SendGrid. https://sendgrid.com/blog/what-is-an-smtp-server/

2. The Mailgun Team.(2020) *Which SMTP port should I use? Understanding ports 25, 465 & 587*. https://www.mailgun.com/blog/email/which-smtp-port-understanding-ports-25-465-587/