Teacher Website

William Guyott

Senior Project – Spring 2017

Sunday, May 07, 2017

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author/Reviewer** | **Description** |
| 2/25/2017 | 1.0 | Will Guyott | First Status Report |
| 4/16/2017 | 1.1 | Will Guyott | Second Status Report |
| 5/7/2017 | 1.2 | Will Guyott | Final Status Report |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Overview 4](#_Toc481949158)

[1.1 Current Progress 4](#_Toc481949159)

[1.1.1 Database 5](#_Toc481949160)

[1.1.2 Website Infrastructure 5](#_Toc481949161)

[1.1.3 Redirect LocalHost to TeacherWebsite 6](#_Toc481949162)

[1.2 Constraints & Assumptions 6](#_Toc481949163)

[1.3 Definitions, Acronyms and Abbreviations 6](#_Toc481949164)

[1.4 References 7](#_Toc481949165)

[2. Website 8](#_Toc481949166)

[2.1 Design 8](#_Toc481949167)

[2.2 Implementation 9](#_Toc481949168)

[2.2.1 login.php 9](#_Toc481949169)

[2.2.2 studentWelcome.php 10](#_Toc481949170)

[2.2.3 studentProfile.php 10](#_Toc481949171)

[2.2.4 studentClasses.php 11](#_Toc481949172)

[2.2.5 studentHomework.php 11](#_Toc481949173)

[2.2.6 studentAnnouncements.php 12](#_Toc481949174)

[2.2.7 teacherWelcome.php 12](#_Toc481949175)

[2.2.8 teacherProfile.php 13](#_Toc481949176)

[2.2.9 teacherClasses.php 13](#_Toc481949177)

[2.2.10 manageTeacherClass.php 14](#_Toc481949178)

[2.2.11 manageTeacherHomework.php 14](#_Toc481949179)

[2.2.12 manageClassAnnouncement.php 15](#_Toc481949180)

[2.2.13 teacherStudents.php 15](#_Toc481949181)

[2.2.14 manageTeacherStudent.php 16](#_Toc481949182)

[2.2.15 manageGlobalAnnouncement.php 16](#_Toc481949183)

[3. Database 17](#_Toc481949184)

[3.1 Design 17](#_Toc481949185)

[3.2 Implementation 17](#_Toc481949186)

[3.2.1 database creation script.txt 17](#_Toc481949187)

[3.2.2 deletion script.txt 18](#_Toc481949188)

[3.2.3 population script.txt 19](#_Toc481949189)

[3.2.4 table creation script.txt 24](#_Toc481949190)

[4. Improvements for Future Versions 28](#_Toc481949191)

# Overview

For my senior project, I am going to create a data driven website that allows a teacher to setup, view and maintain the classes they are teaching, the students in each class and school related information about each student, such as the grade each student is in. The website will also allow for posting assignments and other notes on a class by class basis. The website will allow students to see what classes they are in and view the information that the teacher has posted about each class.

The website will support three roles, Teacher, Student, and Guest.

Supported Roles:

* Teacher
  + Able to login with full website access.
  + Has CRUD access to their personal data, announcement data, class data, and student data.
* Student
  + Able to login with student website access.
  + Has read-only access to general announcements, the classes they are currently in, class announcements, and class homework.
  + Has CRUD access to their personal data.
* Guest
  + See the login page and attempt to login.

The database behind the website will not actually delete any objects stored in the database. Instead each row in each table in the database will have a create data and a delete date. To tell if an object in the database, like a class or a student, is still valid you will need to ensure that the delete date is NULL. Once an object in the database is “deleted” the delete date is set to the date on which it is was “deleted”. This will allow database administrators to determine when an object was deleted and do things like support queries that could tell you what classes were offered for a certain semester and year. So even if a class is no longer available today the database administrator could craft a SQL query to tell you when that class was available based on its create and delete dates. This will allow for future website enhancements that would permit browsing historical data to see when a class was offered, what teacher taught which classes in which semesters, what classes a student took over the course of their entire academic experience, etc.

To implement my teacher website, I will be using the XAMPP PHP development environment. XAMPP is a free Apache HTTP Server distribution containing MariaDB (MySQL), PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.

All of the code and supporting files (such as SQL scripts) for the website will be stored in a GitHub project at: <https://github.com/wguyott3843/TeacherWebsite>.

## Current Progress

With the ending of this semester Version 1.0 of the teacher website is complete. There are still many improvements that can be made to the website. See the section titled “Improvements for Future Versions” for a brief discussion of known issues and desired enchancements.

### Database

The database design and implementation are completed at this time. Please see the “Database” section later in this document for a detailed description of the work that has been completed concerning the teacher website database.

Please see the files in the files in the “sql scripts” folder in my GitHub project (<https://github.com/wguyott3843/TeacherWebsite>) for the actual SQL code used to create and manage the database.

### Website Infrastructure

A prototype login page was one of the first pages to be implemented. This prototype helped me to figure out how to connect HTML code with PHP code and how to connect to the MySQL database from the PHP code. It also helped me to understand how to create and use “session state” for maintaining information in my web application as the user moves from web page to web page. At this point in time a final login page is in place and fully functional.

You can see the actual source files by going to my GitHub project (<https://github.com/wguyott3843/TeacherWebsite>). Most of the website source files are in the root folder of my GitHub project. The php files that specifically interact with the database are located in the “database php” sub folder.

#### loginpage.php

The login page allows the user to enter their username and password and then submit the page for authentication.

login.php includes config.php which establishes our database connection. It then calls session\_start() which starts our server side session, which I will use to keep track of information for this user on our website.

After setting up our environment login.php queries the database with the given username and password to see if the user is valid. If the user is valid login.php then determines if this login is for a teacher or a student and populates appropriate session state variables for use on later pages.

#### config.php

config.php contains the database configuration and login information. This page also initializes the connection to the MySQL database. The PHP code for each web page should include config.php in order to open a connection to the database server so that the PHP code can then access the underlying database.

#### welcomeStudent.php and welcomeTeacher.php

The welcome.php pages include session.php to setup the environment and to confirm that this user is still valid.

The welcome.php pages then display a welcome message, confirming that the user was successfully logged in, and displays a link for logging out.

#### session.php

session.php should be included by the PHP code for all of the other pages besides the login page (login.php). session.php establishes the database connection by include config.php and then reconnects to the session that was started in login.php.

After getting the environment setup, session.php checks to see if this user is a teacher or a student and validates that the login is still valid. If the login for this user is not still valid the user is redirected to the login page to enter new, valid credentials.

#### logout.php

logout.php does not include session.php because the page does not need to setup the user’s environment. logout.php simply closes the current session and redirects the user to the login page.

### Redirect LocalHost to TeacherWebsite

I have also put in place the code needed to direct anyone coming to “localhost” to the teacher website login page. This is done by having an index.php file in the main HTTP Server folder that redirects the web request to the TeacherWebsite folder. In the TeacherWebsite folder there is an index.php file that redirects the web request to the login web page (login.php).

I have put the index.php file for the main HTTP Server folder in my github root directory named as “index.htdocs.php”. You would have to rename this to index.php once you have copied it into the correct location.

## Constraints & Assumptions

The initial implementation will only allow for one teacher per website. The database will be designed in such a way as to allow multiple teachers to be supported in a single website but the website itself will not initially take advantage of this functionality. However, it should be straight forward to enhance the website to support multiple teachers in a future version.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| XAMPP | A PHP Development Environment including Apache, MariaDB, PHP, and Perl |
| Apache | Apache HTTP Server |
| MariaDB | MariaDB Server – An Enhanced Drop-In Replacement for MySQL |
| PHP | General-Purpose Scripting Language Suited to Web Development |
| Perl | A Programming Language |
| CRUD | Create, Read, Update and Delete |
| Bootstrap | An HTML, CSS and JavaScript Framework for Web Development |

## References

* XAMPP  
  <https://www.apachefriends.org/index.html>
* Apache HTTP Server  
  <https://projects.apache.org/project.html?httpd-http_server>
* MariaDB  
  <https://mariadb.org/about/>
* PHP  
  <http://php.net/>
* Perl  
  <https://www.perl.org/>
* Bootstrap  
  <http://getbootstrap.com/>

# Website

## Design

The website will start with a login page. Once the user has logged in they will see different pages on the website depending on whether they are a teacher or a student.

Once a student has logged in they will be sent to a page that shows all of the general announcements that are available and they will be able to choose to go to a page that allows them to view their classes or to manage their personal information.

If they choose to view their classes they will go to a page that shows a list of their current classes. They may then select “homework” to view the homework’s they have for their classes and they may also select “announcements” to view current class announcements for their classes.

The user website pages will consist of the following:

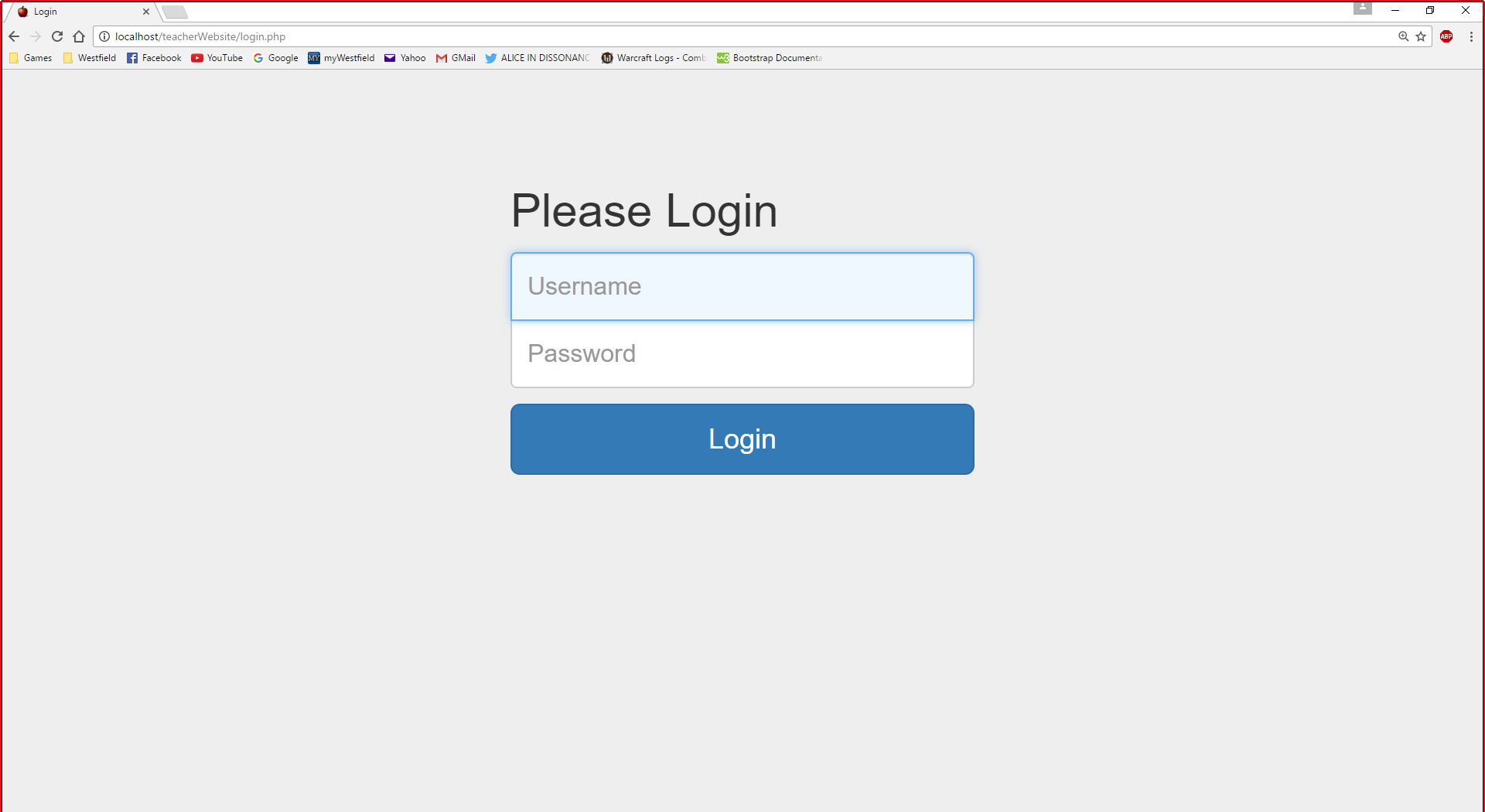
* Login
  + View General Announcements
    - View List of Classes
      * View Class Homework
      * View Class Announcements
    - Manage Personal Data
* Logout

Once a teacher has logged in they will be sent to a page that shows all of the general announcements that are available and they will be able to choose to go to a page that allows them to manage classes, manage students, manage general announcements, or manage their profile.

* Login
  + View General Announcements
    - View Class List
      * Manage Classes
        + Add Class
        + Delete Class
        + Update Class
    - View Homeroom Student List
      * Manage Students
        + Add Students
        + Delete Students
        + Update Students
    - Manage General Announcements
      * Add General Announcements
      * Delete General Announcements
      * Update General Announcements
    - Manage Profile
* Logout

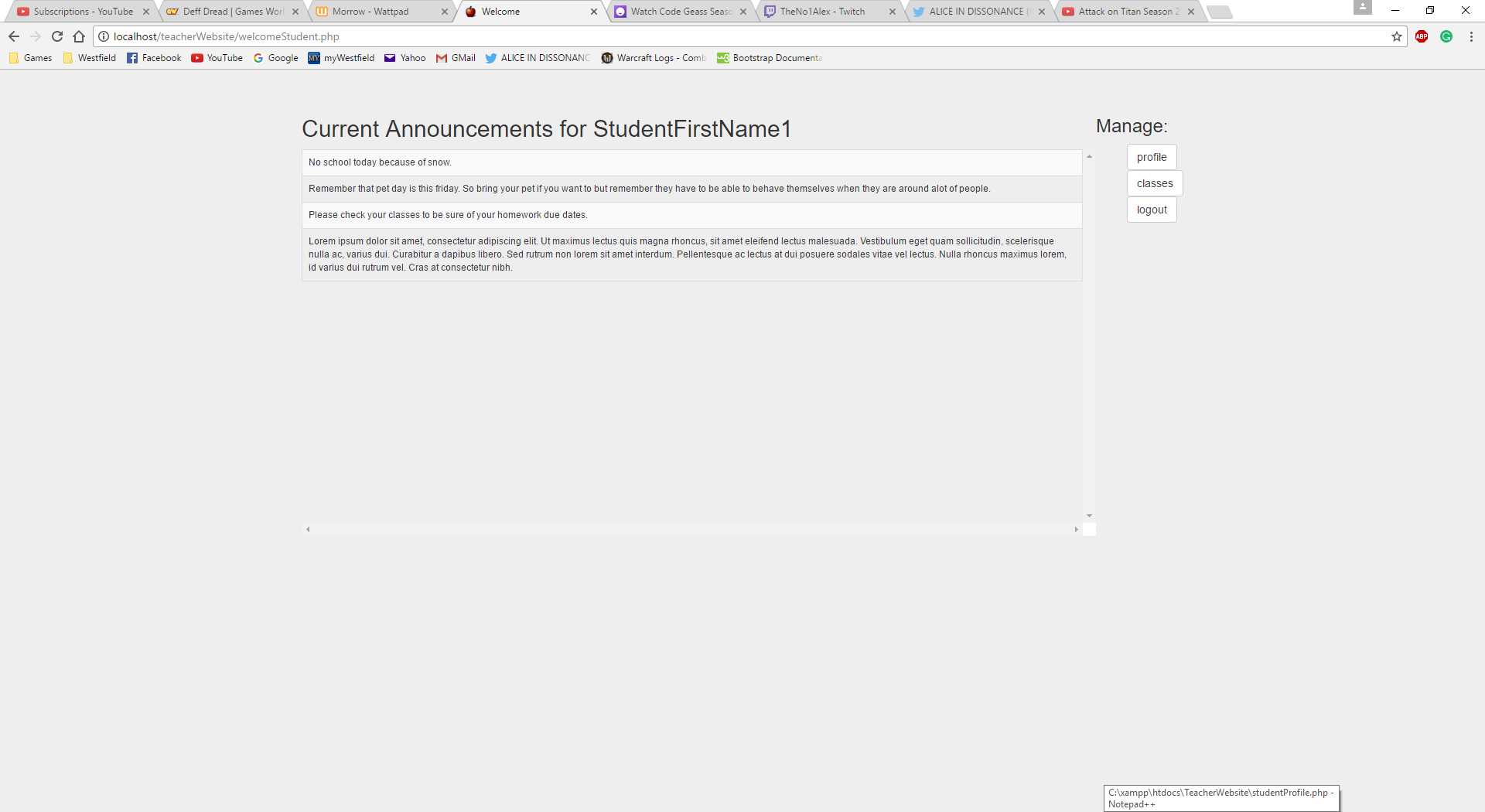
## Implementation

### login.php



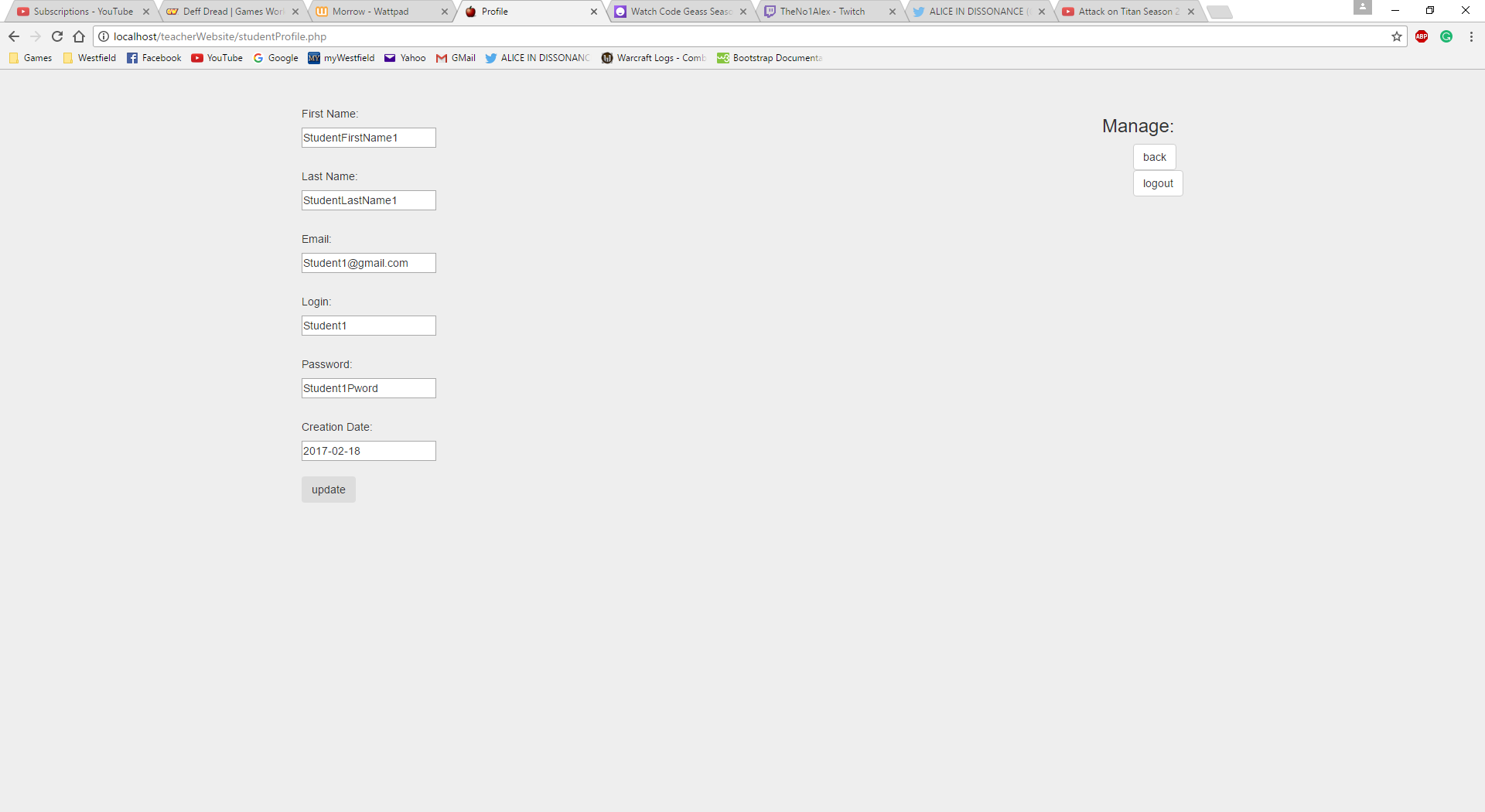
This login page is used by both teachers and students. The website determines whether it’s a teacher or a student based on login credentials.

### studentWelcome.php



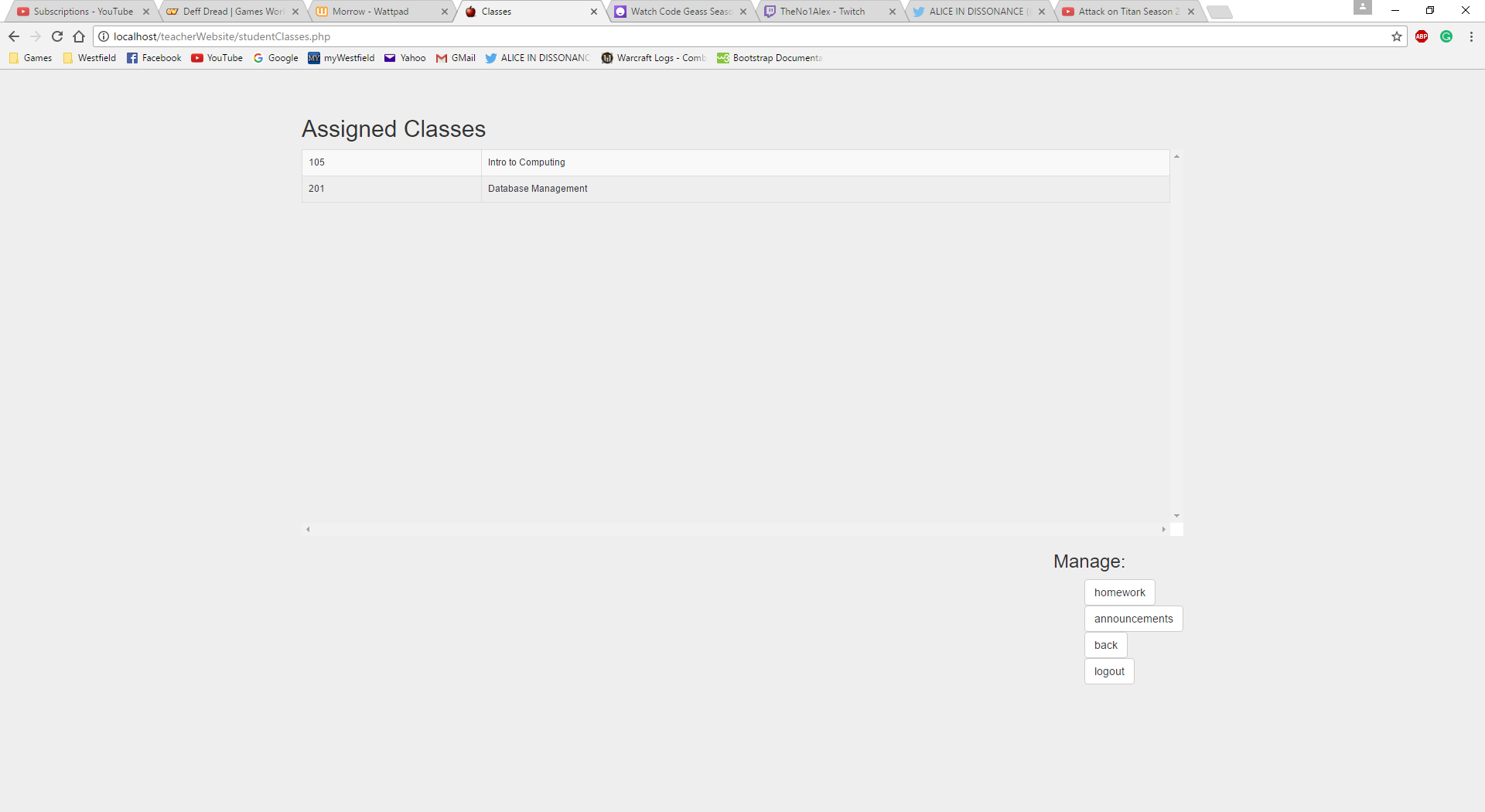
This is the welcome page for students. On this page the student can see all of the general announcements that are currently active. From this page the student can manage their profile, their classes, or logout.

### studentProfile.php



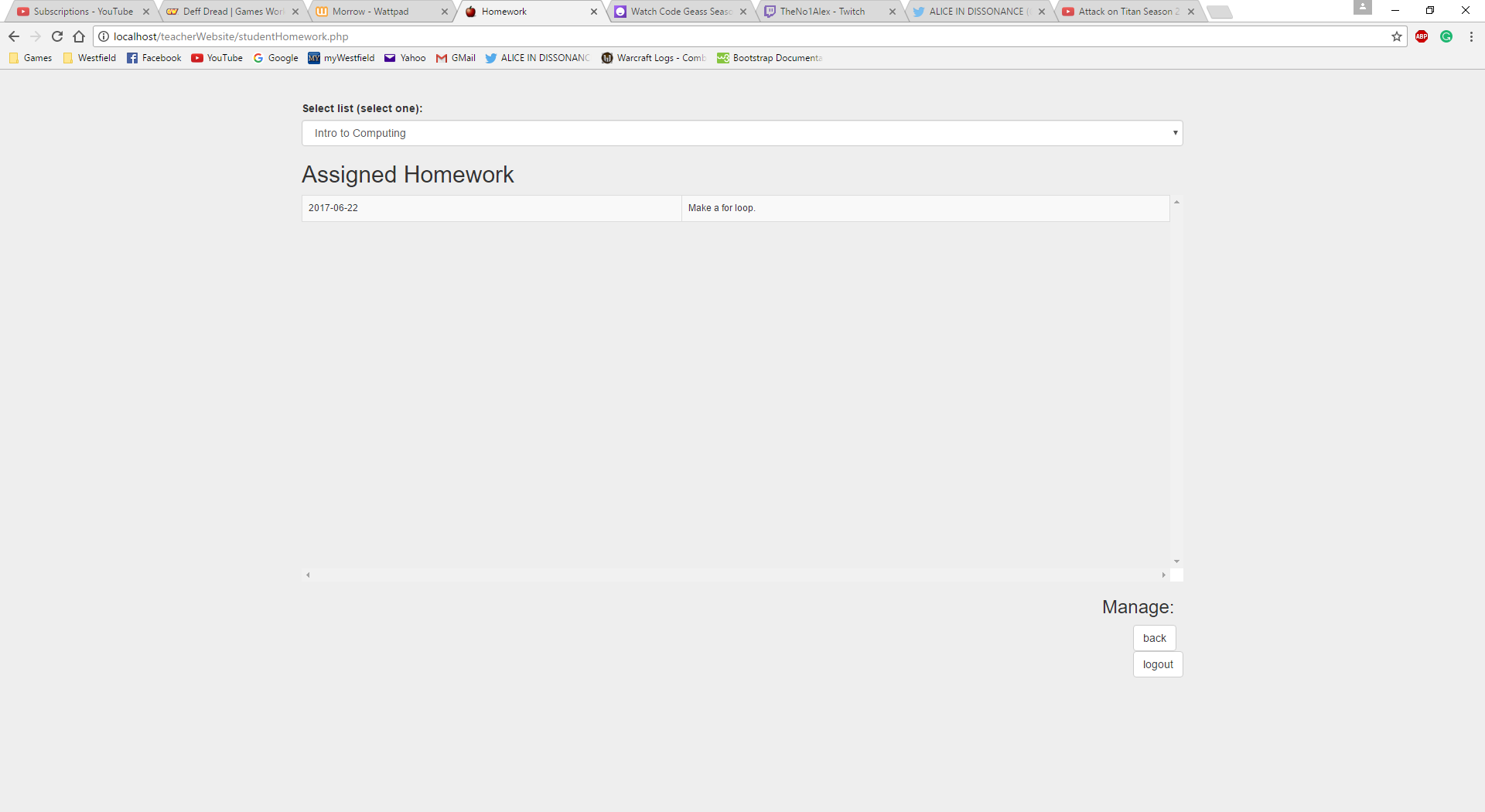
This page allows students to update some but not all of their information.

### studentClasses.php



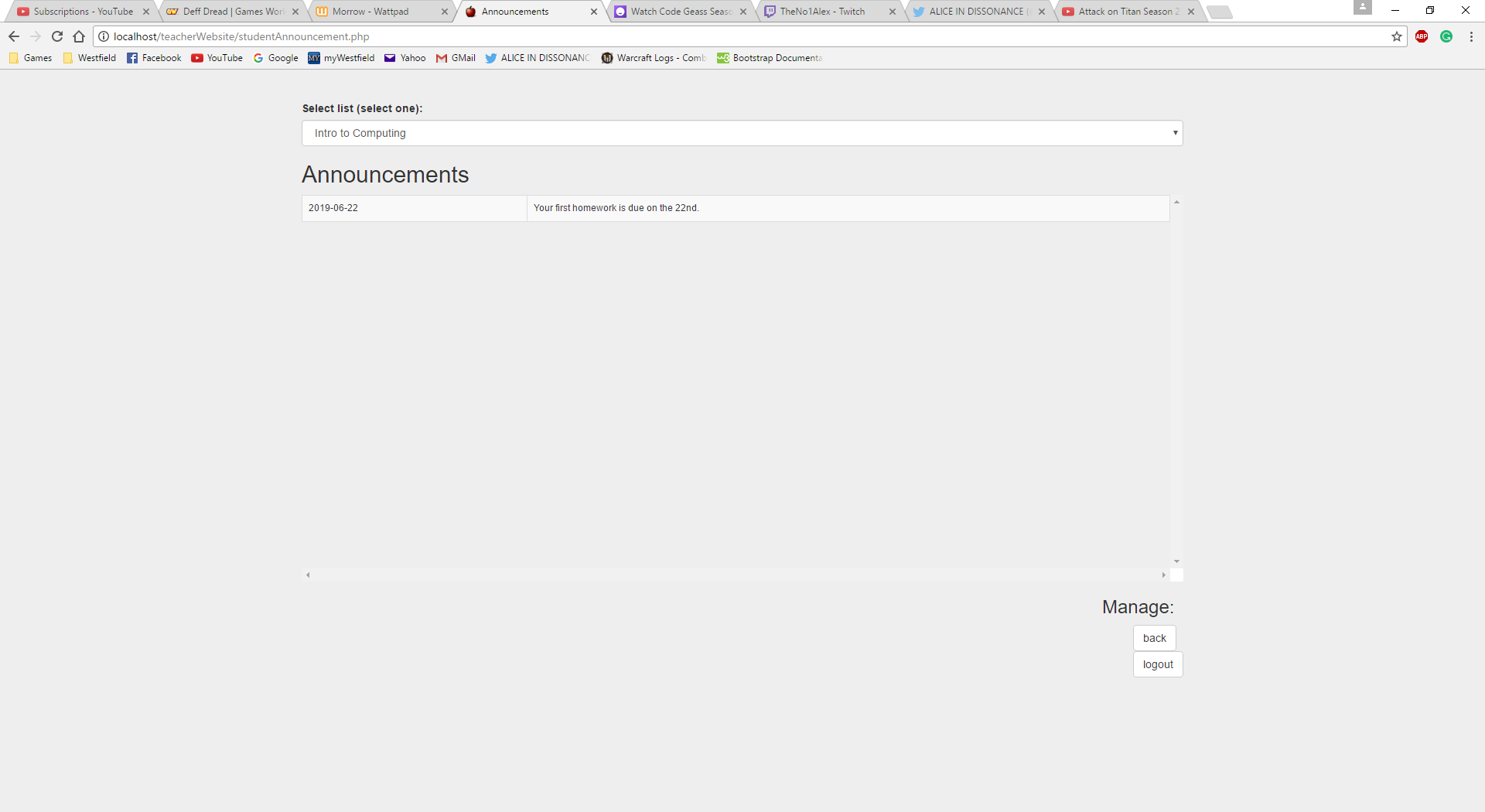
This page is for students to view their assigned classes and look at homework and class specific announcements.

### studentHomework.php



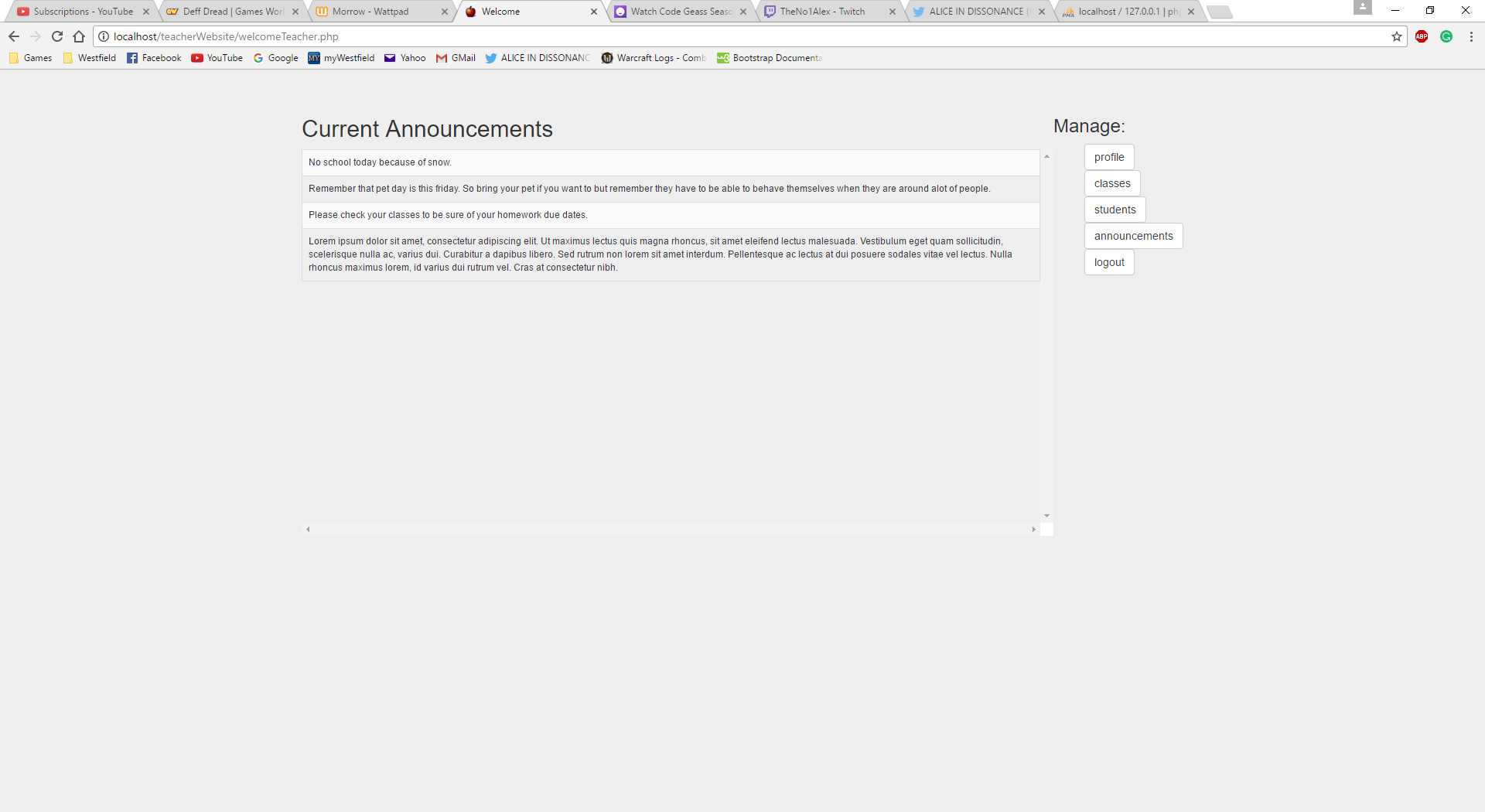
This page is for students to view assigned homework for each class.

### studentAnnouncements.php



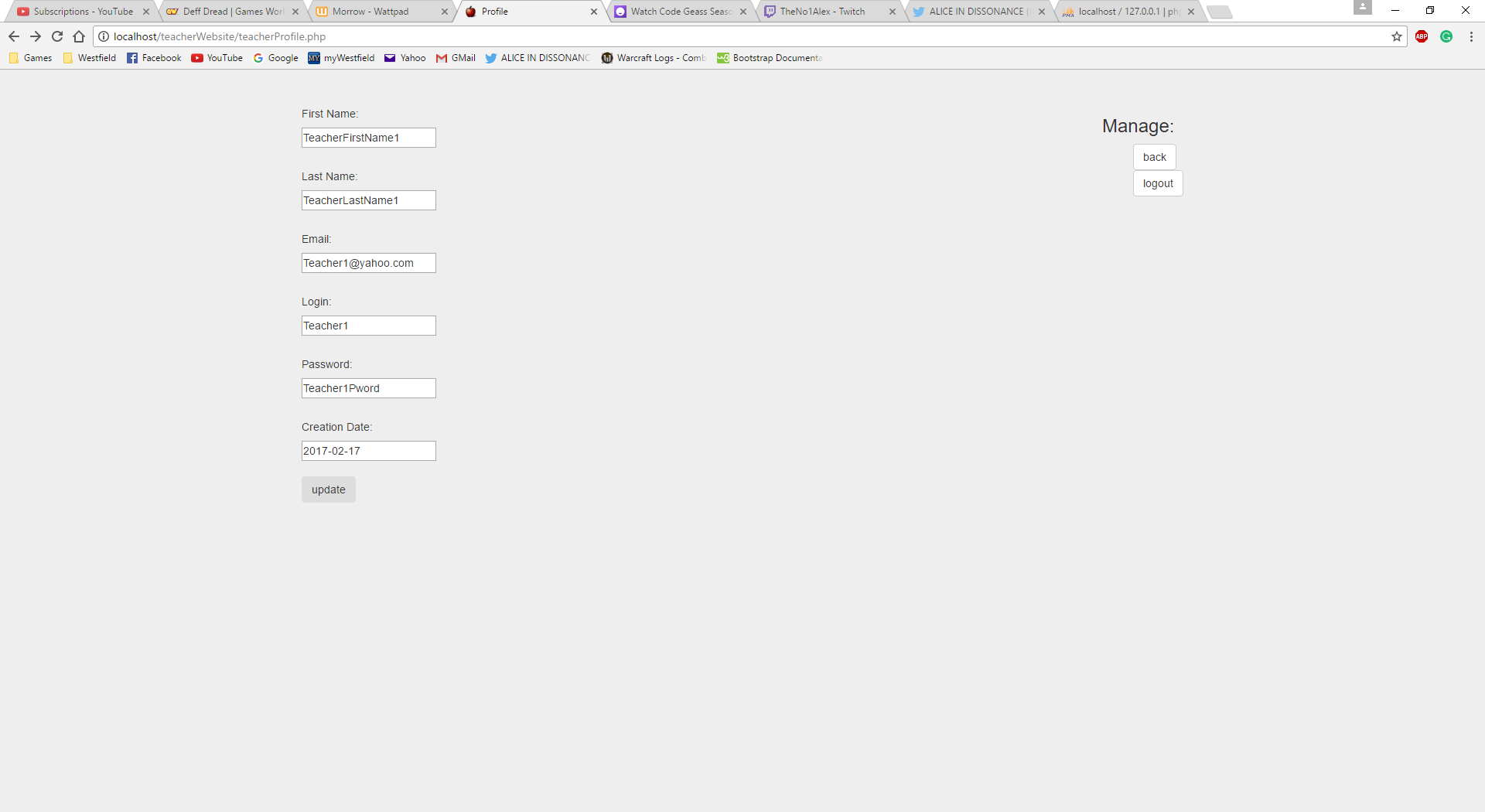
This page is for students to view announcements for each class.

### teacherWelcome.php



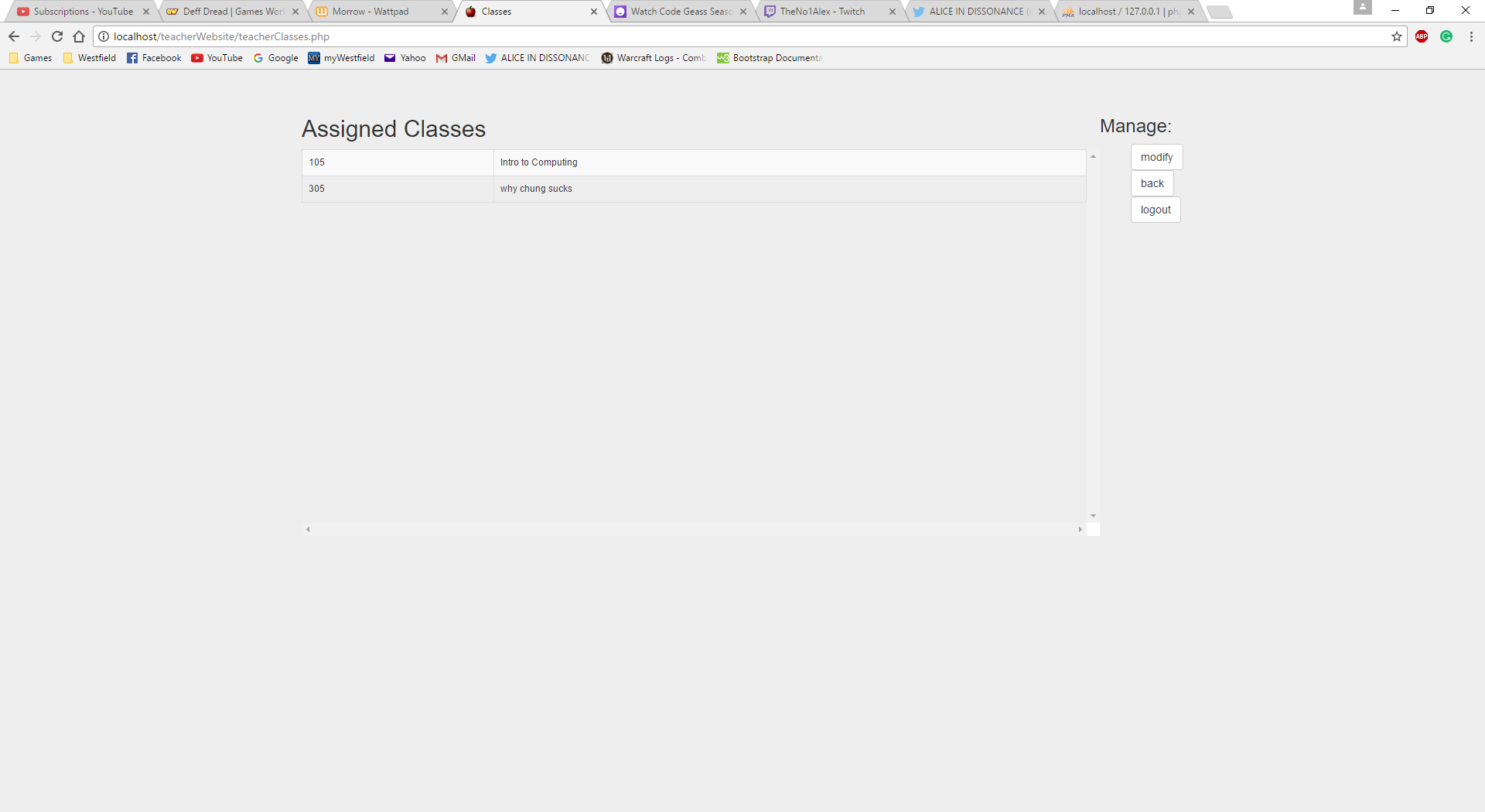
This is the welcome page for teachers. On this page the teacher can update their profile, modify classes, modify students, view announcements, and logout.

### teacherProfile.php



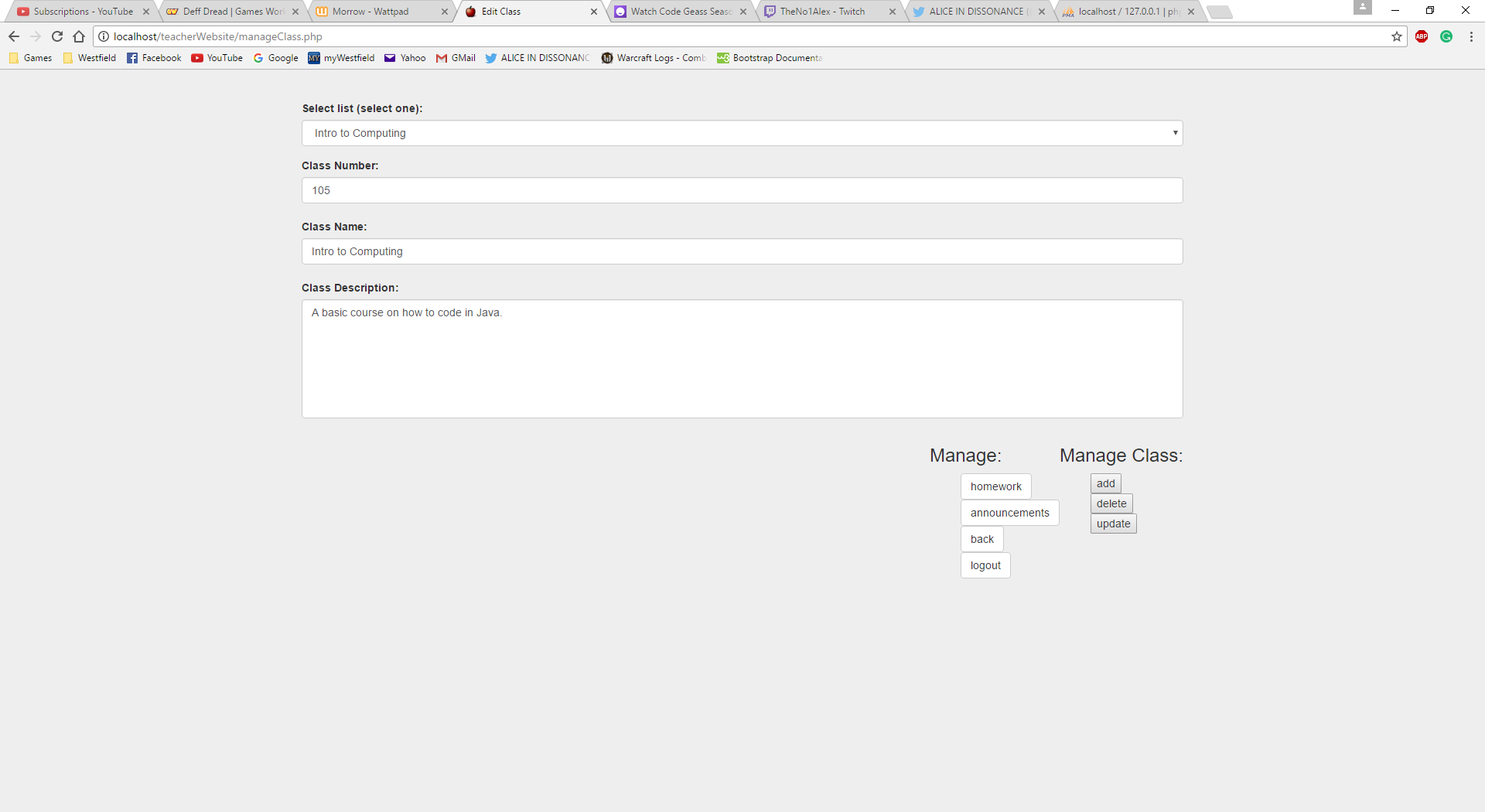
On This page teachers can update their own information, with fewer restrictions than students.

### teacherClasses.php



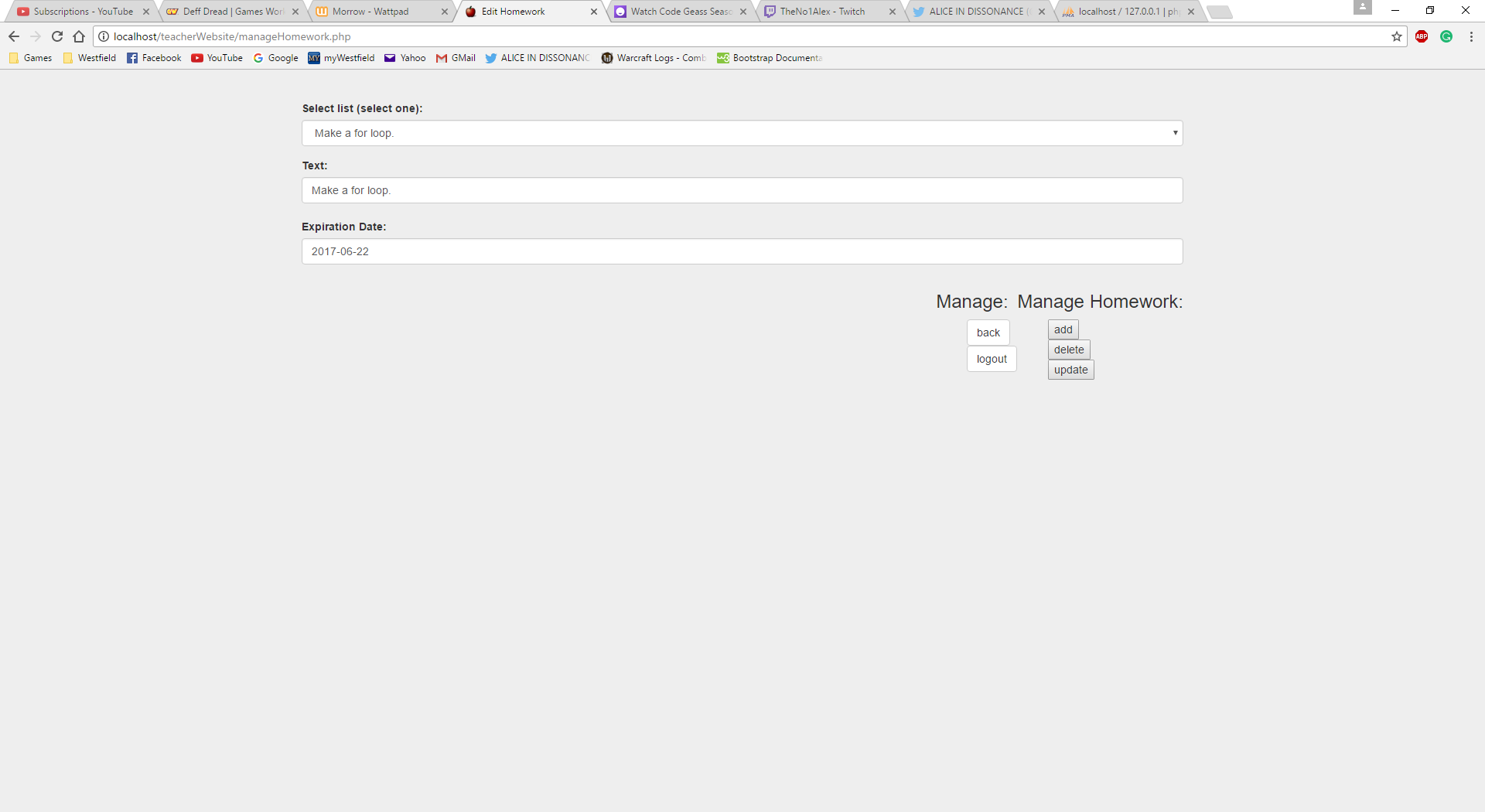
This page allows teachers to view the classes they teach.

### manageTeacherClass.php



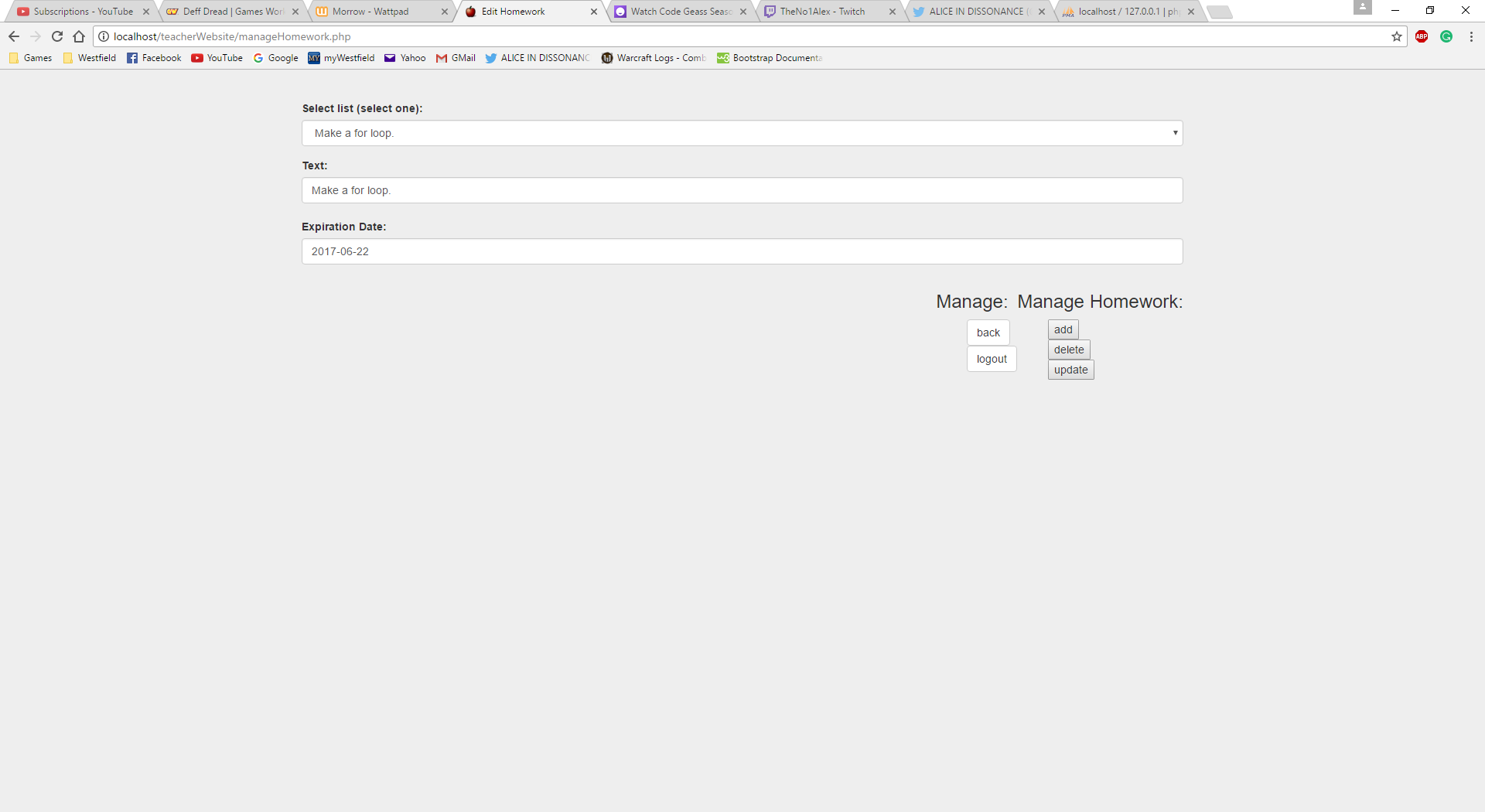
This page allows teachers to alter existing classes, add new classes or delete existing classes. It also allows them to modify homework and class specific announcements.

### manageTeacherHomework.php



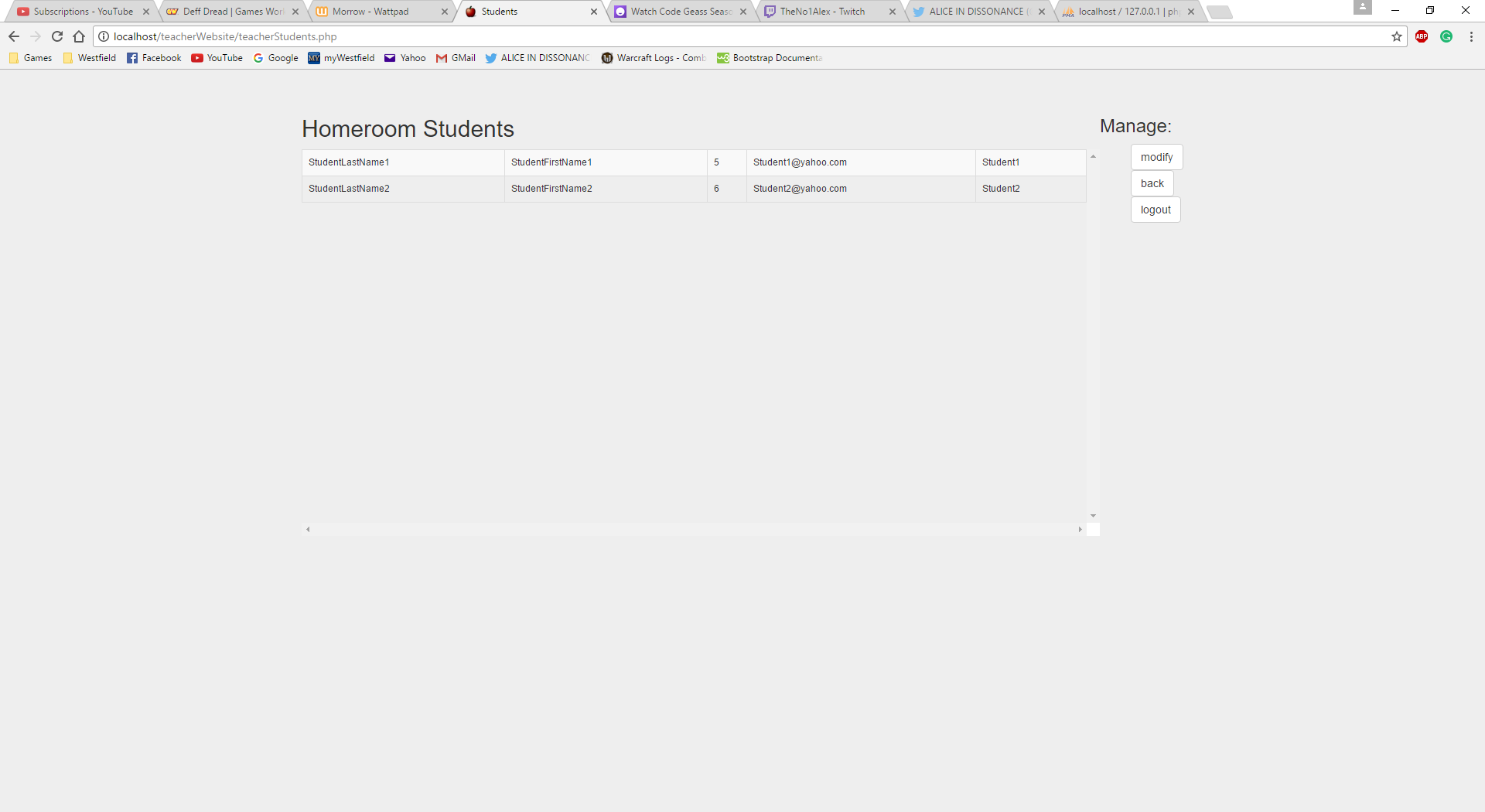
This page allows teachers to alter existing homework, add new homework or delete existing homework.

### manageClassAnnouncement.php



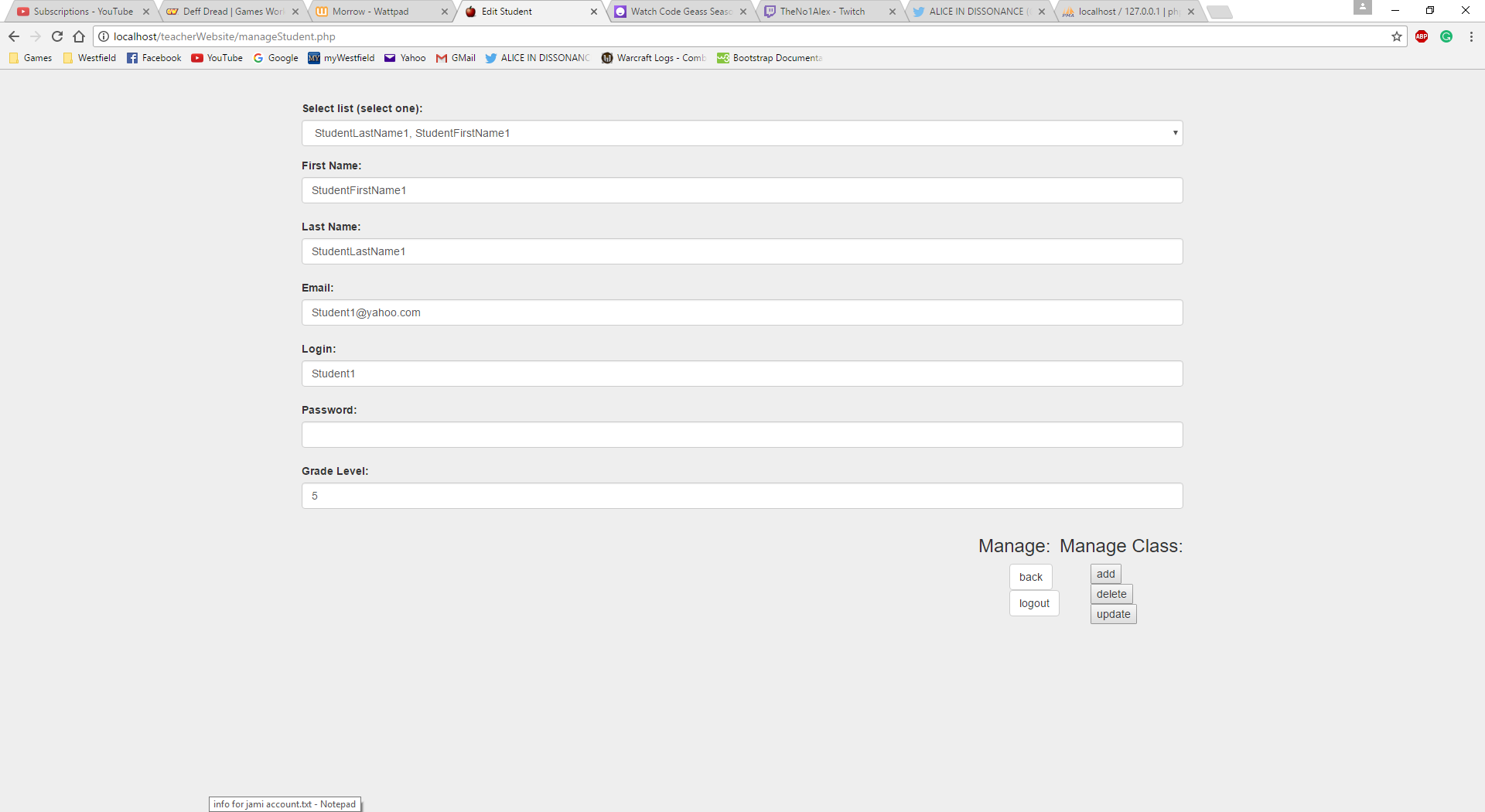
This page allows teachers to alter existing announcements, add new announcements or delete existing announcements.

### teacherStudents.php



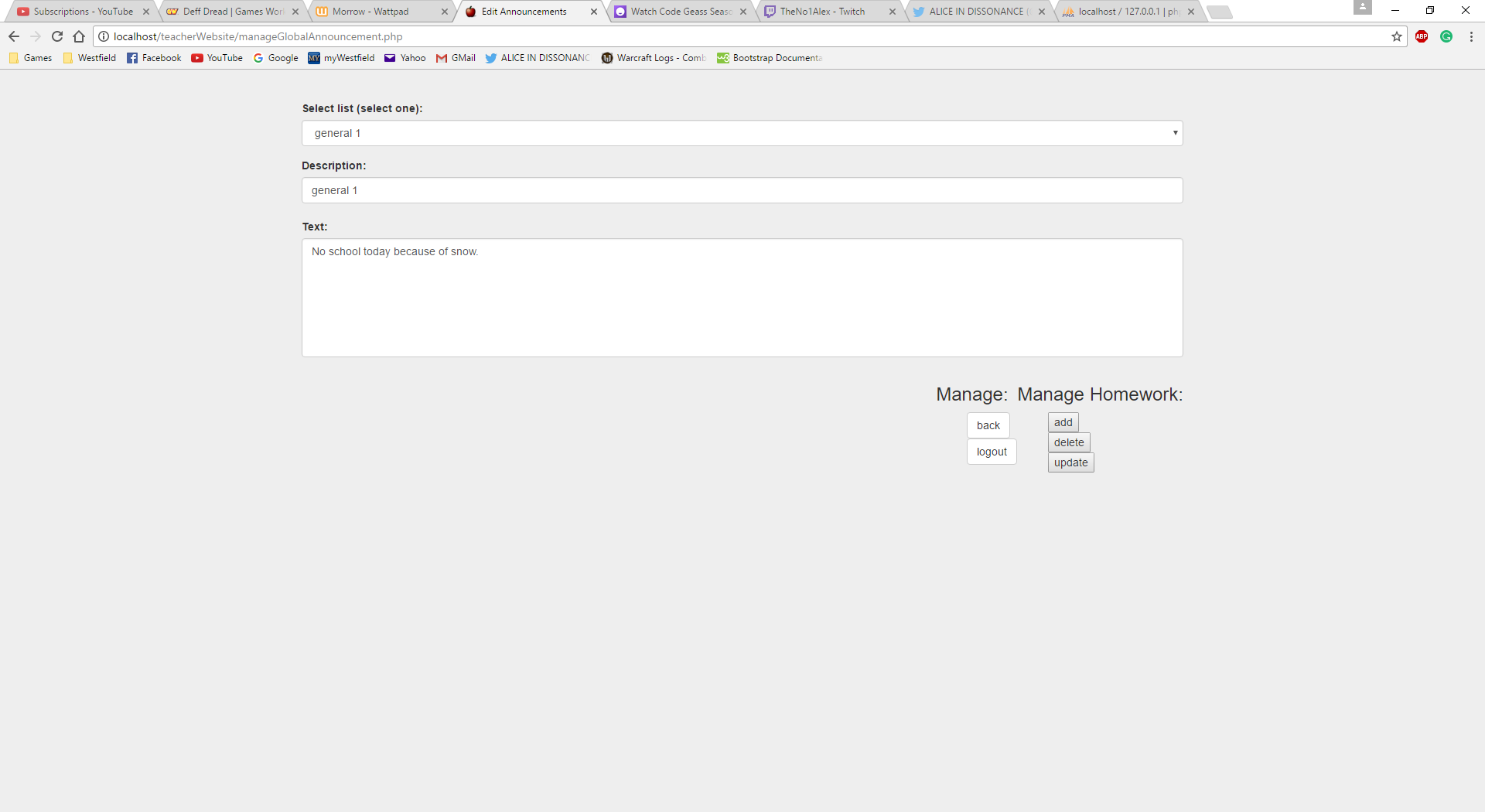
This page allows teachers to view the students in their homeroom.

### manageTeacherStudent.php



This page allows teachers to alter existing students, add new students or delete existing students.

### manageGlobalAnnouncement.php



# Database

## Design

The database will consist of the following tables:

* Person
* Teacher
* Class
* Student
* Homework
* Announcement
* TeacherClassMap
* StudentClassMap

The Person table contains data about all people in our system. Teachers and students are both considered people and have a PersonId in their tables to link the teachers and students to their person information.

A “general” announcement, as opposed to a class specific announcement, can be found by searching the Announcement table and specifying that the ClassId field is NULL.

The TeacherClassMap table is used to be able to keep track of which classes a teacher is assigned to. In some future version of the website this map table would also allow us to have multiple teachers teaching a single class.

## Implementation

All tables have primary and foreign keys as appropriate.

If you go to my GitHub project (<https://github.com/wguyott3843/TeacherWebsite>) and look in the “sql scripts” folder you will find four files: “database creation script.txt”, “deletion script.txt”, “population script.txt”, and “table creation script.txt”.

### database creation script.txt

“database creation script.txt” contains the SQL that I ran to create the TeacherWebsite database. The database uses the utf8mb4 character set and utf8mb4\_unicode\_ci collation.

The utf8mb4 character set uses a maximum of four bytes per character instead of the three bytes per character that the utf8 character set uses and the utf8mb4 character set is a superset of utf8.

The utf8mb4\_unicode\_ci collation is case insensitive and so compares upper and lower cases of the same character as equal.

#### Script Contents

# utf8mb4 is a more complete implementation of utf8

CREATE DATABASE TeacherWebsite

CHARACTER SET utf8mb4

COLLATE utf8mb4\_unicode\_ci;

### deletion script.txt

“deletion script.txt” contains the SQL to drop all of the tables in the database.

What is important about this script is that it drops the tables in the order needed to avoid getting errors when you attempt to drop a table that contains foreign key references to other tables.

This script is useful when I want to wipe the database and start fresh. I run this script then the table creation script followed by the table population script.

#### Script Contents

# delete the last table i created first and the first table i created last

# basically delete the tables in the opposite order you created them

USE TeacherWebsite;

DROP TABLE StudentClassMap;

USE TeacherWebsite;

DROP TABLE TeacherClassMap;

USE TeacherWebsite;

DROP TABLE Announcement;

USE TeacherWebsite;

DROP TABLE Homework;

USE TeacherWebsite;

DROP TABLE Student;

USE TeacherWebsite;

DROP TABLE Class;

USE TeacherWebsite;

DROP TABLE Teacher;

USE TeacherWebsite;

DROP TABLE Person;

### population script.txt

“population script.txt” contains the SQL to insert test data into the database for testing the website implementation.

#### Script Contents

# add test person data

USE TeacherWebsite;

INSERT INTO Person (FirstName,LastName,Email,Login,Password,CreateDate,DeleteDate)

VALUES ('TeacherFirstName1','TeacherLastName1','Teacher1@yahoo.com','Teacher1','Teacher1Pword','2017-02-17',NULL);

INSERT INTO Person (FirstName,LastName,Email,Login,Password,CreateDate,DeleteDate)

VALUES ('TeacherFirstName2','TeacherLastName2','Teacher2@yahoo.com','Teacher2','Teacher2Pword','2017-02-17',NULL);

INSERT INTO Person (FirstName,LastName,Email,Login,Password,CreateDate,DeleteDate)

VALUES ('StudentFirstName1','StudentLastName1','Student1@yahoo.com','Student1','Student1Pword','2017-02-18',NULL);

INSERT INTO Person (FirstName,LastName,Email,Login,Password,CreateDate,DeleteDate)

VALUES ('StudentFirstName2','StudentLastName2','Student2@yahoo.com','Student2','Student2Pword','2017-02-18',NULL);

INSERT INTO Person (FirstName,LastName,Email,Login,Password,CreateDate,DeleteDate)

VALUES ('StudentFirstName3','StudentLastName3','Student3@yahoo.com','Student3','Student3Pword','2017-02-18',NULL);

INSERT INTO Person (FirstName,LastName,Email,Login,Password,CreateDate,DeleteDate)

VALUES ('StudentFirstName4','StudentLastName4','Student4@yahoo.com','Student4','Student4Pword','2017-02-18',NULL);

#add test teacher data

USE TeacherWebsite;

INSERT INTO Teacher (PersonId)

SELECT Id FROM Person WHERE LastName = 'TeacherLastName1';

INSERT INTO Teacher (PersonId)

SELECT Id FROM Person WHERE LastName = 'TeacherLastName2';

#add test class data

USE TeacherWebsite;

INSERT INTO Class (Number, Name, Description, CreateDate, DeleteDate)

VALUES ('101', 'Intro to Computing', 'A basic course on how to code in Java.', '2017-02-17',NULL);

INSERT INTO Class (Number, Name, Description, CreateDate, DeleteDate)

VALUES ('102', 'Web Programming', 'A course on how to build a website.', '2017-02-17',NULL);

INSERT INTO Class (Number, Name, Description, CreateDate, DeleteDate)

VALUES ('201', 'Database Management', 'An advanced course on how to make a database using mysql.', '2017-02-17',NULL);

INSERT INTO Class (Number, Name, Description, CreateDate, DeleteDate)

VALUES ('202', 'Computing I', 'An introduction to the basics of programming computers.', '2017-02-17',NULL);

#add test student data

USE TeacherWebsite;

INSERT INTO Student (PersonId, GradeLevel, HomeroomTeacherId)

SELECT (SELECT Id FROM Person WHERE LastName = 'StudentLastName1' AND FirstName = 'StudentFirstName1'), 5, (SELECT Id FROM Teacher WHERE PersonId = (SELECT Id FROM Person WHERE FirstName = 'TeacherFirstName1' AND LastName = 'TeacherLastName1'));

INSERT INTO Student (PersonId, GradeLevel, HomeroomTeacherId)

SELECT (SELECT Id FROM Person WHERE LastName = 'StudentLastName2' AND FirstName = 'StudentFirstName2'), 6, (SELECT Id FROM Teacher WHERE PersonId = (SELECT Id FROM Person WHERE FirstName = 'TeacherFirstName1' AND LastName = 'TeacherLastName1'));

INSERT INTO Student (PersonId, GradeLevel, HomeroomTeacherId)

SELECT (SELECT Id FROM Person WHERE LastName = 'StudentLastName3' AND FirstName = 'StudentFirstName3'), 7, (SELECT Id FROM Teacher WHERE PersonId = (SELECT Id FROM Person WHERE FirstName = 'TeacherFirstName2' AND LastName = 'TeacherLastName2'));

INSERT INTO Student (PersonId, GradeLevel, HomeroomTeacherId)

SELECT (SELECT Id FROM Person WHERE LastName = 'StudentLastName4' AND FirstName = 'StudentFirstName4'), 8, (SELECT Id FROM Teacher WHERE PersonId = (SELECT Id FROM Person WHERE FirstName = 'TeacherFirstName2' AND LastName = 'TeacherLastName2'));

# add test homework data

USE TeacherWebsite;

INSERT INTO Homework(ClassId, Text, ExpirationDate, CreateDate, DeleteDate)

SELECT Id,'Make a for loop.', '2017-06-22', '2017-02-18', NULL FROM Class WHERE Number = '101';

INSERT INTO Homework(ClassId, Text, ExpirationDate, CreateDate, DeleteDate)

SELECT Id,'Make the homepage for a website of a company that sells cars.', '2017-06-25', '2017-02-18', NULL FROM Class WHERE Number = '102';

INSERT INTO Homework(ClassId, Text, ExpirationDate, CreateDate, DeleteDate)

SELECT Id, 'Write the sql to create 3 separate tables with atleast 5 items in each.', '2017-06-30', '2017-02-18', NULL FROM Class WHERE Number = '201';

INSERT INTO Homework(ClassId, Text, ExpirationDate, CreateDate, DeleteDate)

SELECT Id, 'Add primary keys to the tables you created in the last assignment', '2017-07-05', '2017-02-18', NULL FROM Class WHERE Number = '201';

# add test announcements data

USE TeacherWebsite;

INSERT INTO Announcement(ClassId, Description, Text, ExpirationDate, CreateDate, DeleteDate)

SELECT Id,'intro to computing 1','Your first homework is due on the 22nd.', '2019-06-22', '2017-02-18', NULL FROM Class WHERE Number = '101';

INSERT INTO Announcement(ClassId, Description, Text, ExpirationDate, CreateDate, DeleteDate)

SELECT Id,'webprogramming 1','Your first homework assignment is due on the 25th.','2019-06-25', '2017-02-18', NULL FROM Class WHERE Number = '102';

INSERT INTO Announcement(ClassId, Description, Text, ExpirationDate, CreateDate, DeleteDate)

SELECT Id,'database 1','Your first homework assignment is due on the 30th.', '2019-06-30', '2017-02-18', NULL FROM Class WHERE Number = '102';

# this is a general announcement so we left ClassId as NULL

INSERT INTO Announcement(ClassId, Description, Text, ExpirationDate, CreateDate, DeleteDate)

VALUES (NULL,'general 1','No school today because of snow.', '2019-06-30','2017-03-14', NULL);

INSERT INTO Announcement(ClassId, Description, Text, ExpirationDate, CreateDate, DeleteDate)

VALUES (NULL,'general 2','Please check your classes to be sure of your homework due dates.', '2019-06-30','2017-02-18', NULL);

INSERT INTO Announcement(ClassId, Description, Text, ExpirationDate, CreateDate, DeleteDate)

VALUES (NULL,'general 3','Remember that pet day is this friday. So bring your pet if you want to but remember they have to be able to behave themselves when they are around alot of people.', '2019-06-30','2017-03-01', NULL);

INSERT INTO Announcement(ClassId, Description, Text, ExpirationDate, CreateDate, DeleteDate)

VALUES (NULL,'general 4','Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut maximus lectus quis magna rhoncus, sit amet eleifend lectus malesuada. Vestibulum eget quam sollicitudin, scelerisque nulla ac, varius dui. Curabitur a dapibus libero. Sed rutrum non lorem sit amet interdum. Pellentesque ac lectus at dui posuere sodales vitae vel lectus. Nulla rhoncus maximus lorem, id varius dui rutrum vel. Cras at consectetur nibh.', '2019-06-30','2017-02-18', NULL);

# add test teacherclassmap data

USE TeacherWebsite;

INSERT INTO TeacherClassMap (TeacherId, ClassId, CreateDate, DeleteDate)

SELECT Id, (SELECT Id FROM Class WHERE Number = '101'),'2017-02-18', NULL FROM Teacher WHERE PersonId = (SELECT Id FROM Person WHERE LastName = 'TeacherLastName1');

INSERT INTO TeacherClassMap (TeacherId, ClassId, CreateDate, DeleteDate)

SELECT Id, (SELECT Id FROM Class WHERE Number = '102'),'2017-02-18', NULL FROM Teacher WHERE PersonId = (SELECT Id FROM Person WHERE LastName = 'TeacherLastName2');

INSERT INTO TeacherClassMap (TeacherId, ClassId, CreateDate, DeleteDate)

SELECT Id, (SELECT Id FROM Class WHERE Number = '201'),'2017-02-18', NULL FROM Teacher WHERE PersonId = (SELECT Id FROM Person WHERE LastName = 'TeacherLastName1');

INSERT INTO TeacherClassMap (TeacherId, ClassId, CreateDate, DeleteDate)

SELECT Id, (SELECT Id FROM Class WHERE Number = '202'),'2017-02-18', NULL FROM Teacher WHERE PersonId = (SELECT Id FROM Person WHERE LastName = 'TeacherLastName2');

# add test studentclassmap data

USE TeacherWebsite;

INSERT INTO StudentClassMap (StudentId, ClassId, CreateDate, DeleteDate)

SELECT Id, (SELECT Id FROM Class WHERE Number = '101'),'2017-02-18', NULL FROM Student WHERE PersonId = (SELECT Id FROM Person WHERE LastName = 'StudentLastName1');

INSERT INTO StudentClassMap (StudentId, ClassId, CreateDate, DeleteDate)

SELECT Id, (SELECT Id FROM Class WHERE Number = '102'),'2017-02-18', NULL FROM Student WHERE PersonId = (SELECT Id FROM Person WHERE LastName = 'StudentLastName2');

INSERT INTO StudentClassMap (StudentId, ClassId, CreateDate, DeleteDate)

SELECT Id, (SELECT Id FROM Class WHERE Number = '201'),'2017-02-18', NULL FROM Student WHERE PersonId = (SELECT Id FROM Person WHERE LastName = 'StudentLastName1');

INSERT INTO StudentClassMap (StudentId, ClassId, CreateDate, DeleteDate)

SELECT Id, (SELECT Id FROM Class WHERE Number = '202'),'2017-02-18', NULL FROM Student WHERE PersonId = (SELECT Id FROM Person WHERE LastName = 'StudentLastName2');

INSERT INTO StudentClassMap (StudentId, ClassId, CreateDate, DeleteDate)

SELECT Id, (SELECT Id FROM Class WHERE Number = '202'),'2017-02-18', NULL FROM Student WHERE PersonId = (SELECT Id FROM Person WHERE LastName = 'StudentLastName3');

INSERT INTO StudentClassMap (StudentId, ClassId, CreateDate, DeleteDate)

SELECT Id, (SELECT Id FROM Class WHERE Number = '202'),'2017-02-18', NULL FROM Student WHERE PersonId = (SELECT Id FROM Person WHERE LastName = 'StudentLastName4');

### table creation script.txt

“table creation script.txt” contains the SQL to create the tables in the database.

#### Table Creation Script Contents

# create the User table

USE TeacherWebsite;

CREATE TABLE Person (

Id INT NOT NULL AUTO\_INCREMENT,

FirstName VARCHAR(20) NOT NULL,

LastName VARCHAR(30) NOT NULL,

Email VARCHAR(60) NOT NULL,

Login VARCHAR(25) NOT NULL,

Password VARCHAR(25) NOT NULL,

CreateDate DATE NOT NULL,

DeleteDate DATE NULL,

PRIMARY KEY (Id),

UNIQUE KEY (FirstName, LastName)

);

# create the Teacher table

USE TeacherWebsite;

CREATE TABLE Teacher (

Id INT NOT NULL AUTO\_INCREMENT,

PersonId INT NOT NULL,

PRIMARY KEY (Id),

CONSTRAINT fk\_teacher\_person\_id

FOREIGN KEY (PersonId)

REFERENCES Person (Id)

ON DELETE RESTRICT

ON UPDATE RESTRICT

);

# create the Class table

USE TeacherWebsite;

CREATE TABLE Class(

Id INT NOT NULL AUTO\_INCREMENT,

Number INT NOT NULL,

Name VARCHAR(40) NOT NULL,

Description VARCHAR(500) NOT NULL,

CreateDate DATE NOT NULL,

DeleteDate DATE NULL,

PRIMARY KEY (Id),

UNIQUE (Number)

);

# create the Student table

USE TeacherWebsite;

CREATE TABLE Student (

Id INT NOT NULL AUTO\_INCREMENT,

PersonId INT NOT NULL UNIQUE,

GradeLevel INT NOT NULL,

HomeroomTeacherId INT NOT NULL,

PRIMARY KEY (Id),

CONSTRAINT fk\_student\_person\_id

FOREIGN KEY (PersonId)

REFERENCES Person (Id)

ON DELETE RESTRICT

ON UPDATE RESTRICT,

CONSTRAINT fk\_student\_teacher\_id

FOREIGN KEY (HomeroomTeacherId)

REFERENCES Teacher (Id)

ON DELETE RESTRICT

ON UPDATE RESTRICT

);

# create the Homework table

USE TeacherWebsite;

CREATE TABLE Homework(

Id INT NOT NULL AUTO\_INCREMENT,

ClassId INT NOT NULL,

Text VARCHAR(50) NOT NULL,

ExpirationDate DATE NOT NULL,

CreateDate DATE NOT NULL,

DeleteDate DATE NULL,

PRIMARY KEY (Id),

CONSTRAINT fk\_homework\_class\_id

FOREIGN KEY (ClassId)

REFERENCES Class (Id)

ON DELETE RESTRICT

ON UPDATE RESTRICT

);

# create the Announcement table

USE TeacherWebsite;

CREATE TABLE Announcement(

Id INT NOT NULL AUTO\_INCREMENT,

ClassId INT NULL,

Description VARCHAR(20) NOT NULL,

Text VARCHAR(500) NOT NULL,

ExpirationDate DATE NOT NULL,

CreateDate DATE NOT NULL,

DeleteDate DATE NULL,

PRIMARY KEY (Id),

CONSTRAINT fk\_announcement\_class\_id

FOREIGN KEY (ClassId)

REFERENCES Class (Id)

ON DELETE RESTRICT

ON UPDATE RESTRICT

);

# create the TeacherClassMap table

USE TeacherWebsite;

CREATE TABLE TeacherClassMap(

TeacherId INT NOT NULL,

ClassId INT NOT NULL,

CreateDate DATE NOT NULL,

DeleteDate DATE NULL,

CONSTRAINT fk\_teacherclassmap\_teacher\_id

FOREIGN KEY (TeacherId)

REFERENCES Teacher (Id)

ON DELETE RESTRICT

ON UPDATE RESTRICT,

CONSTRAINT fk\_teacherclassmap\_class\_id

FOREIGN KEY (ClassId)

REFERENCES Class (Id)

ON DELETE RESTRICT

ON UPDATE RESTRICT

);

# create the StudentClassMap table

USE TeacherWebsite;

CREATE TABLE StudentClassMap(

StudentId INT NOT NULL,

ClassId INT NOT NULL,

CreateDate DATE NOT NULL,

DeleteDate DATE NULL,

CONSTRAINT fk\_studentclassmap\_student\_id

FOREIGN KEY (StudentId)

REFERENCES Student (Id)

ON DELETE RESTRICT

ON UPDATE RESTRICT,

CONSTRAINT fk\_studentclassmap\_class\_id

FOREIGN KEY (ClassId)

REFERENCES Class (Id)

ON DELETE RESTRICT

ON UPDATE RESTRICT

);

# Improvements for Future Versions

Here is the list of known deficiencies in the current website:

1. There is a bug where if two people logged on in the same computer and in the same browser the session gets messed up.
2. The user interface for adding and updating classes isn’t the best it could be.
3. There is quite a bit of white space and the website doesn’t look very interesting.
4. Currently there isn’t a way to add a student to a class(any newly created students just get assigned as a member of that teachers homeroom)
5. Deleting a student then trying to add a student with the same first and last name will result in a failure to add the student because their name will still be in the database.

Here is the list of potential enhancements for the next version:

1. Spice up the website with more images and colors.
2. Make better use of space.
3. Implement a way to let teachers assign students to classes.