

**CPSC 597- Project**

**Assessment Report Generation Software**

**Faculty Advisor:**

**Dr. Doina Bein**

**Project Final Report**

**Submitted By:**

**Husamaldeen Naji (890948680)**

**Department of Computer Science**

**California State University, Fullerton**

**Spring 2017**



**Department of Computer Science**

**This project has been satisfactorily demonstrated and is of suitable form.**

**This project report is acceptable in partial completion of the requirements for the Master of Science degree in Computer Science.**

**Assessment Report Generation Software**



**Project Title (type)**

**HUSAMALDEEN NAJI**



**Student Name (type)**

**Dr. DOINA BEIN**



**Advisor's Name (type)**

 **Date:**

**Advisor's signature**

**Dr. ANAND PANANGADAN**



**Reviewers name**

 **Date:**

**Reviewer's signature**

**Dr. Professor NATHAN REED**



**Reviewers name**

 **Date:**

**Reviewer's signature**

Table of Contents

[1. Introduction 4](#_Toc481232307)

[2. Project Objective 4](#_Toc481232308)

[3. Project Development Activities 5](#_Toc481232309)

[4. Software Architecture 5](#_Toc481232310)

[4.1 Hardware and Software Requirements 8](#_Toc481232311)

[4.2 Installation 9](#_Toc481232312)

[5. Project Results 9](#_Toc481232313)

[6. Usage Documentation: 10](#_Toc481232314)

[6.1 Users: 10](#_Toc481232315)

[6.1.1 User login: 10](#_Toc481232316)

[6.1.2 New user registration: 11](#_Toc481232317)

[6.2 Faculty: 12](#_Toc481232318)

[6.2.1 How to add a new class: 13](#_Toc481232319)

[6.2.2 How to add students to new created class: 15](#_Toc481232320)

[6.2.3 Adding students’ grades: 17](#_Toc481232321)

[6.2.4 Adding assessment indicators: 18](#_Toc481232322)

[6.2.5 Displaying assessment report: 21](#_Toc481232323)

[6.2.6 Downloading assessment report: 21](#_Toc481232324)

[6.2.7 Downloading assessment indicators’ files: 21](#_Toc481232325)

[6.3 Admin: 22](#_Toc481232326)

[6.3.1 Displaying assessment report for certain class: 22](#_Toc481232327)

[6.3.2 Downloading assessment report: 22](#_Toc481232328)

[6.3.3 Download assessment indicators’ files: 22](#_Toc481232329)

[7. Conclusion: 23](#_Toc481232330)

[8. Future improvements: 23](#_Toc481232331)

[9. Bibliography 24](#_Toc481232332)

# 1. Introduction

Education plays an extremely important role in intellectual development. In order for the whole educational process to stay healthy and consistent across institutions and states, inspections and evaluations need to be applied, and measurements need to be taken [2].

Assessment, as a solution, is currently one of the more important self-evaluation methodologies of a department, college and university [1]. It is the process of collecting data, which describes the student's performance in a set of relevant courses within a degree program [5][4].

However, this solution exposes some obstacles, which makes it a bit challenging for many educational institutions to gain the expected benefits when applying assessment [2]. Most of the obstacles are there because the process of assessment is manual and it takes a lot of time and effort from the faculty to collect and organize the data [2][3]. As a proposed solution, I am offering a method to automate the whole process of assessment. This is software that takes care of the collecting, organizing, and report generation of assessment data. These processes are all done in one step with no user interaction needed. In this way, it will be easier for faculty to generate assessment reports, allowing more time for other tasks.

# 2. Project Objective

There are two user roles in this project administrator (admin) and faculty. For the faculty role, each semester the instructor needs to create and maintain a database with students in the class, their grades for assignments, a set of assessment indicators, and actual assignments collected or scanned as computer files. These components are used along with these indicators in order to represent benchmarks for what satisfactory, developing, and unsatisfactory work should look like. The department needs the following data for assessing the performance of students in that class:

1- For each assignment, how many students have submitted homework?

2- For each assignment, how many students have performed in each category?

The categories are following the pattern of (Satisfactory – Developing – Unsatisfactory). For each assignment, the software system will produce for the instructor, a report on how many students fall into each category, which assessment indicators are selected, and will also allow the instructor to save the report and select a sample assignment for each category item. When saving, the instructor needs to be able to save and store the file on the instructor’s local machine. As for the admin role, the administrator, after login, can view all classes taught by all instructors. Administrator can also click on each of these classes to view the assessment report for particular classes. The assessment report, viewed by the admin, contains a report on how many students fall into each category, a list of indicators selected for that class, and the actual work files representing each indicator as (Satisfactory – Developing – Unsatisfactory). The admin can simply click on each file to download it to the admin’s machine. This project aims to assist both faculty members and the department through the automation of the assessment reports generation process. Faculty members will be able to use the software to create classes and then add students and their grades, add assessment indicators, and finally generate assessment reports for these classes. Additionally, the department will be able to use this software to view all the classes taught by all instructors, and be able to view and download copies of assessment reports for each class.

# 3. Project Development Activities

The software development process has two major parts. The first part is the client-side development and the second part is the server-side development. In the client-side part, the initial activities aim for preparing a front-end structure for the client using:

* HTML language to create the website structure (several interlinked web pages), displays information for the faculty to input their assessment data: this will be linked to the Department website.
* Formatting style of these webpages will be done using CSS.
* Data collection from the user, data validation, and other interactive communication between the client and the database will be done using JavaScript.

The second part deals with the server side development. The databases will be designed, created, and populated with data. Node.JS (using Express.JS as an MVC framework), will be used to create server-side functionality. The main function of the server is to take the user requests and query the database for the requested data, and then respond back to the user and present the requested data. Node.JS (according to Node.JS Wikipedia page) is a JavaScript runtime environment that is used for server-side programing. As a tool, Node.JS has some impressive advantages that makes it a great tool for building web-based applications. Node.JS is an open source environment, therefore no licenses are needed to use Node.JS, or any of its related extensions (which can be installed and used to add extra functionality). More importantly, Node.JS uses the concept of asynchronous events handling. Therefore, users do not suffer from blocking when they are performing requests or actions. Another important factor is that Node.JS has a strong community base, which allows easier access to the solutions for the problems encountered.

# 4. Software Architecture

This web application has a 3-tier architecture (shown in figure 1). Three computers with Internet connection are needed:

* First computer is the client computer; it is used for accessing the website.
* The second computer is the application server, is hosting the application server.
* The third computer is the data server; it is used to store the databases.

However, for testing purposes, there are technologies to create this environment in a single computer.

Client-side programming languages are HTML, CSS, and JavaScript. HTML language is used to create the structure of webpages of the university website. CSS language is used to add style to these webpages. JavaScript is used to add interactivity to the web pages and it comes in handy when forms containing user data are validated.

With respect to server-side technologies, Node JS are used to create the server and to handle user requests.

For the data management, MySQL is used to store and manage the databases with the data collected from the clients.



[Figure 1 - Three tier architecture].

There are three main modules in the system (shown in figure 2). These modules reside on the server-side and their job is to respond to the user requests, along with providing the mechanism to communicate with the database through the proper queries. Finally, these modules will provide the proper response to the user. Figure 3, shows some possible routes for a user request, starting from the frontend, then to the backend modules, and then all the way to the database.



[Figure 2- System’s main backend modules].



[Figure 3- frontend to backend communication].

## 4.1 Hardware and Software Requirements

***Software Requirements:***

Client-Side Programming : HTML, CSS, JavaScript

Server-Side Programming : Node JS.

Frameworks : Express JS.

Database : MySQL.

Operating System : Windows/ Linux (as a Vagrant Virtual Machine)

***Hardware Requirements:***

Processor : Intel Core i5-4210U CPU @ 1.70GHz 2.40 GHz

RAM : 8 GB

Mouse : Optical Mouse

Screen Width : 14’’ inches, or larger.

## 4.2 Installation

In general, this software is structured to be installed (i.e. hosted) and used by departments in universities and colleges. The department can host both the web application (i.e. the software) and the database on their servers, or any other hosting service that the department is using. On the other hand, users like faculty should be provided with the link (URL) for the application, so they can use it.

However, a faculty member can use this software privately (i.e. on his or her PC or laptop), to make the process of producing assessment reports easier. In this case, the user needs to do the following:

* Install MySQL database, please visit: <https://www.mysql.com/downloads/> for installation instructions, after MySQL installation you need to create an empty database using create database command.
* Install Node.JS, please visit: <https://nodejs.org/en/> for installation instructions.
* Import the database.sql file which contains the SQL instructions to create the database on your machine, to your newly installed MySQL. To do that simply use the following command in your command prompt:

*Mysqldump -u [username] -p [database name] < database.sql*

* Finally, in command prompt simply use the node command to run the application. To do so, in command prompt run the following:

*Node [filename].js*

# 5. Project Results

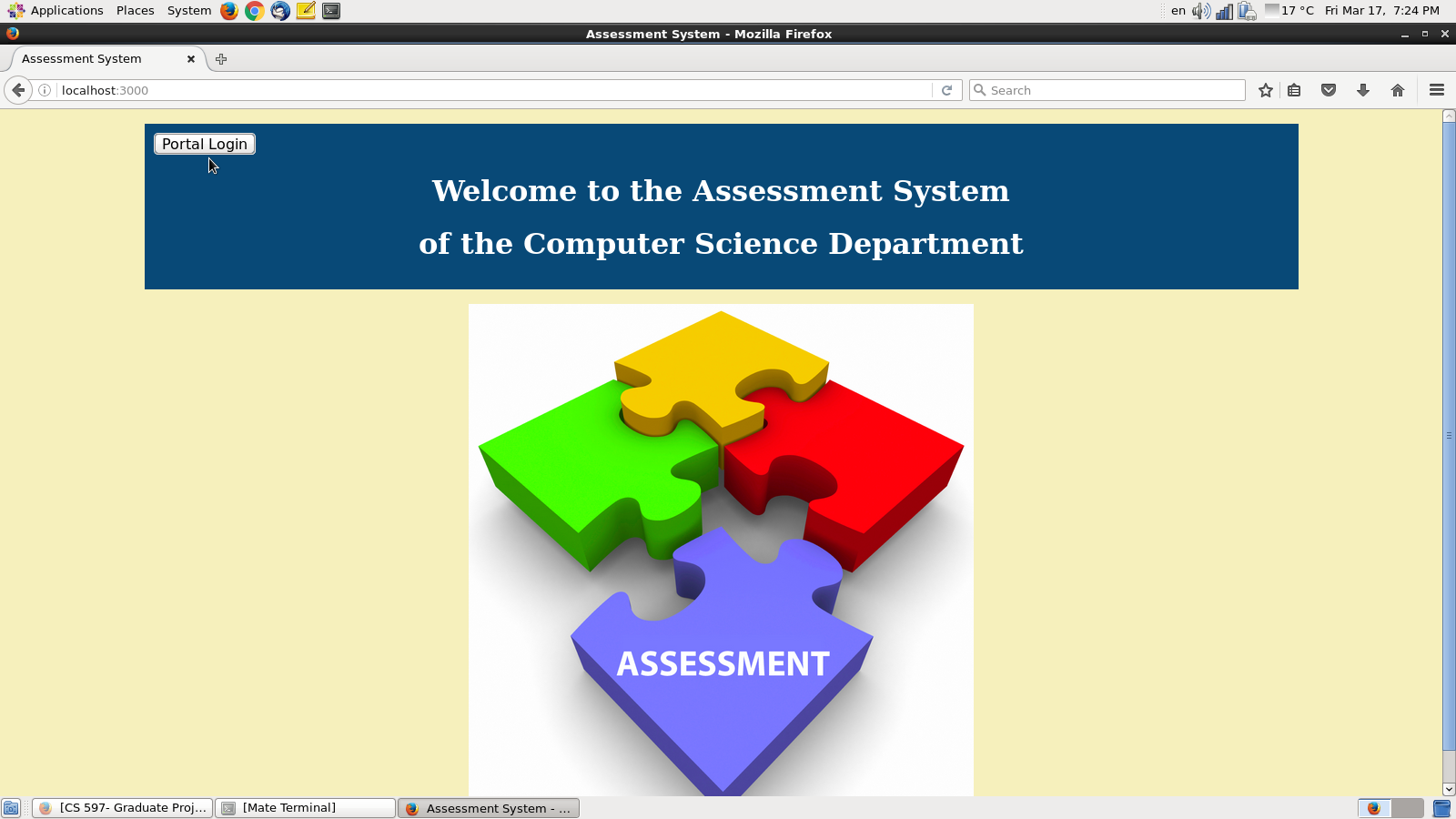
The expected result of this project is the assessment data report from faculty in a Department, which provides information regarding statistics of students’ performance in a specific course subject, assessment indicators used for that subject, and example student work falling into three categories (satisfactory, developing, and unsatisfactory). The reports will be available as PDF files. However, collecting and generating the assessment data needs to be done systematically among all faculty in a Department, thus the software needs to provide a user-friendly interface for data collection and report generation. Thus, an entire website will be created to host the Assessment generation. Furthermore, the website will contain the functionality for faculty and admins to register and login. Faculty will be responsible for uploading and downloading assignments and term projects, the setting of students’ grades and viewing of the data for the entire class, and adding assessment indicators along with exemplary work files for each category.

# 6. Usage Documentation:

## 6.1 Users:

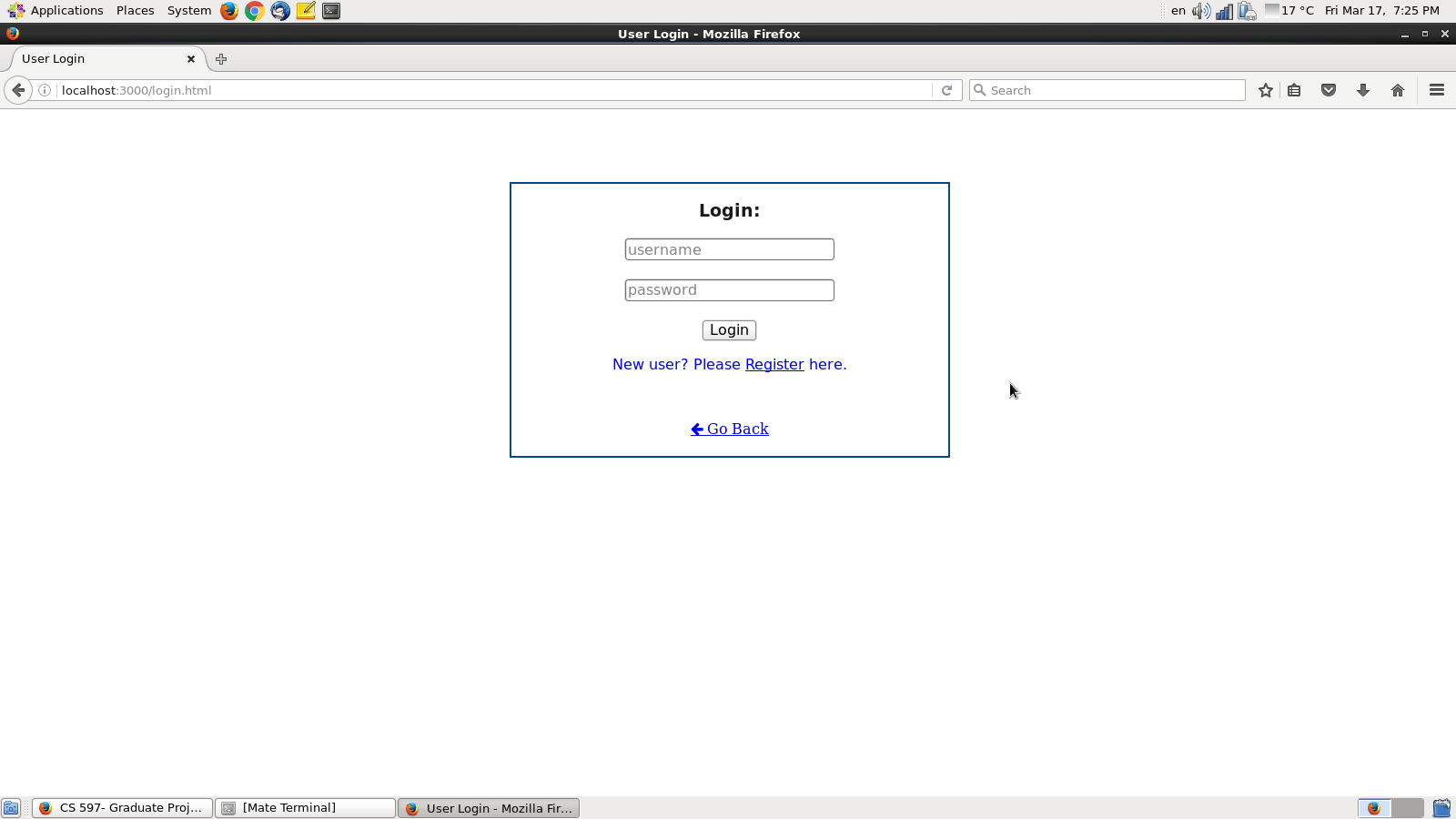
### 6.1.1 User login:

After navigation to the software website, using any web browser of your choice, click on the login button as shown in figure [4]



[Figure 4 - index page].

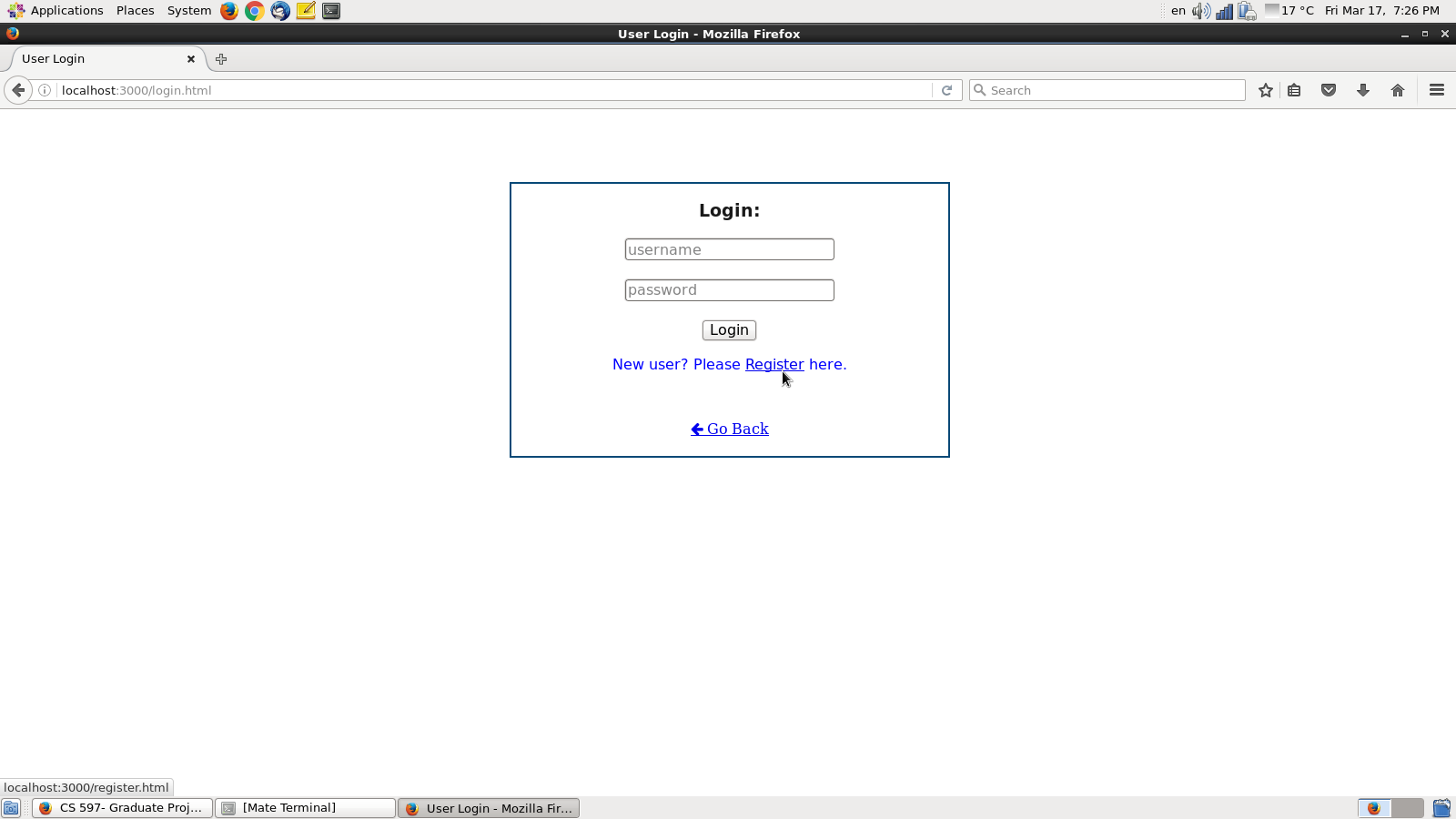
After that, the login page will appear. Enter your username and password. Then click on login button as shown in figure [2]



[ Figure 2 - login page].

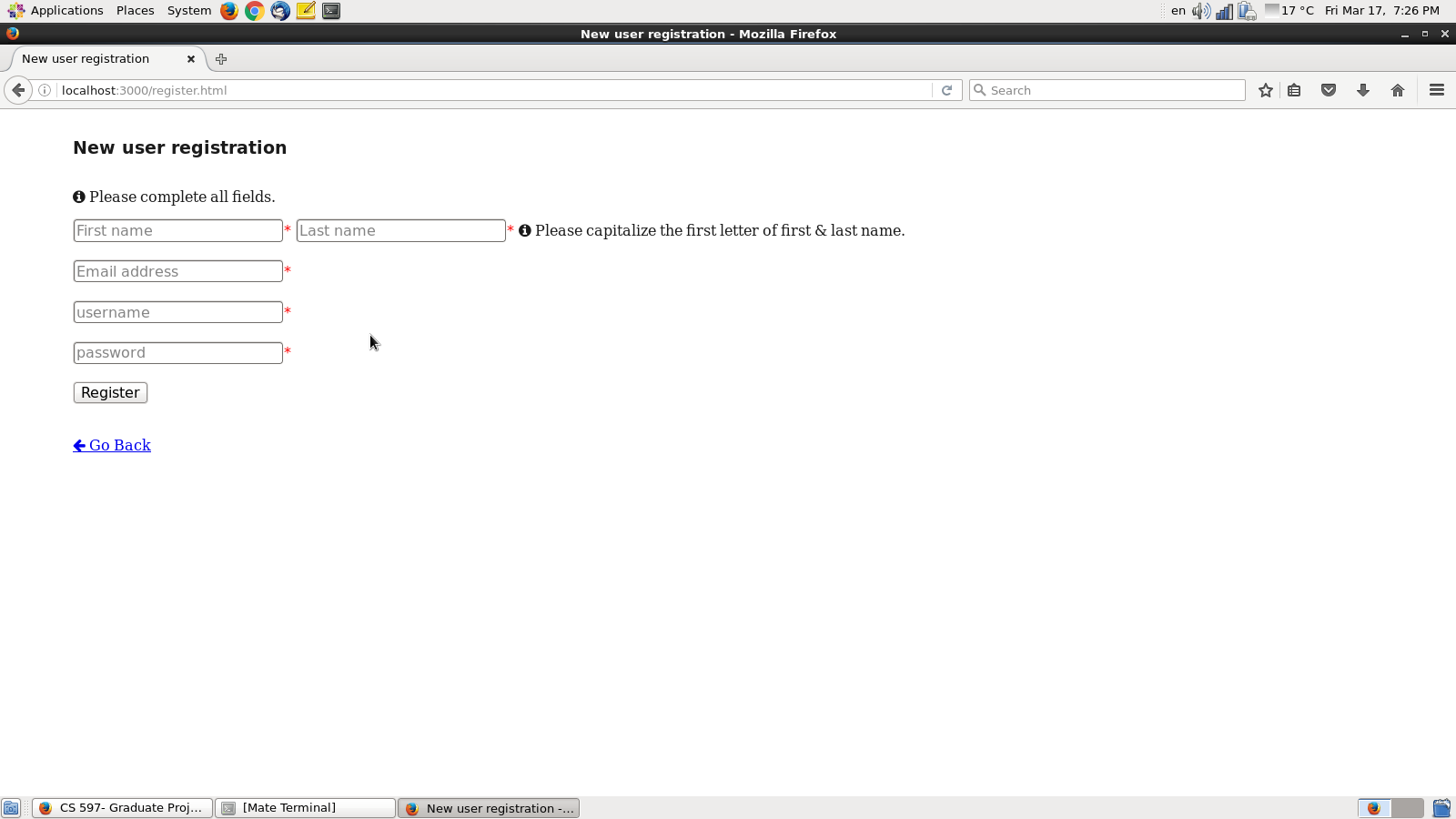
### 6.1.2 New user registration:

In case you are a new user and do not have username and password, you can always register easily. To do so, click on the register link as shown in figure [3]



[Figure 3 - register link].

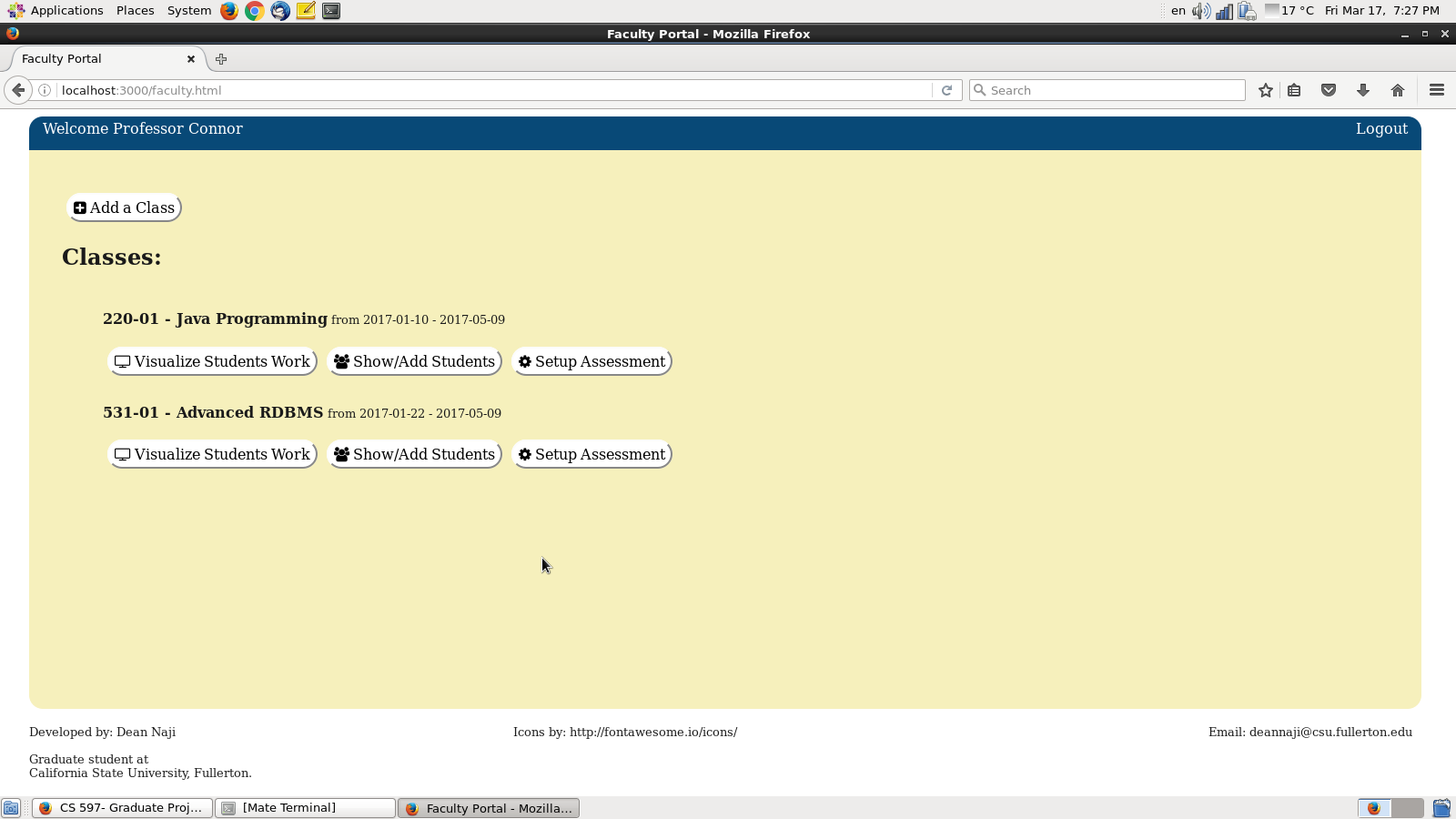
After clicking on the register link, a new user registration form will show up (see figure 4). Please notice that all fields are necessary, and it is notable that they are all marked with red (\*) in order to emphasize necessity. Additionally, notice that an email address is also requested. However, this software doesn’t send, distribute, or present emails. It is only there to provide future possibility to add “forgot username or password?” feature which uses user’s email as lookup parameter to get user’s credentials, as part of future improvements. Figure [4] shows user’s registration page.



[Figure 4 - add registration form page].

## 6.2 Faculty:

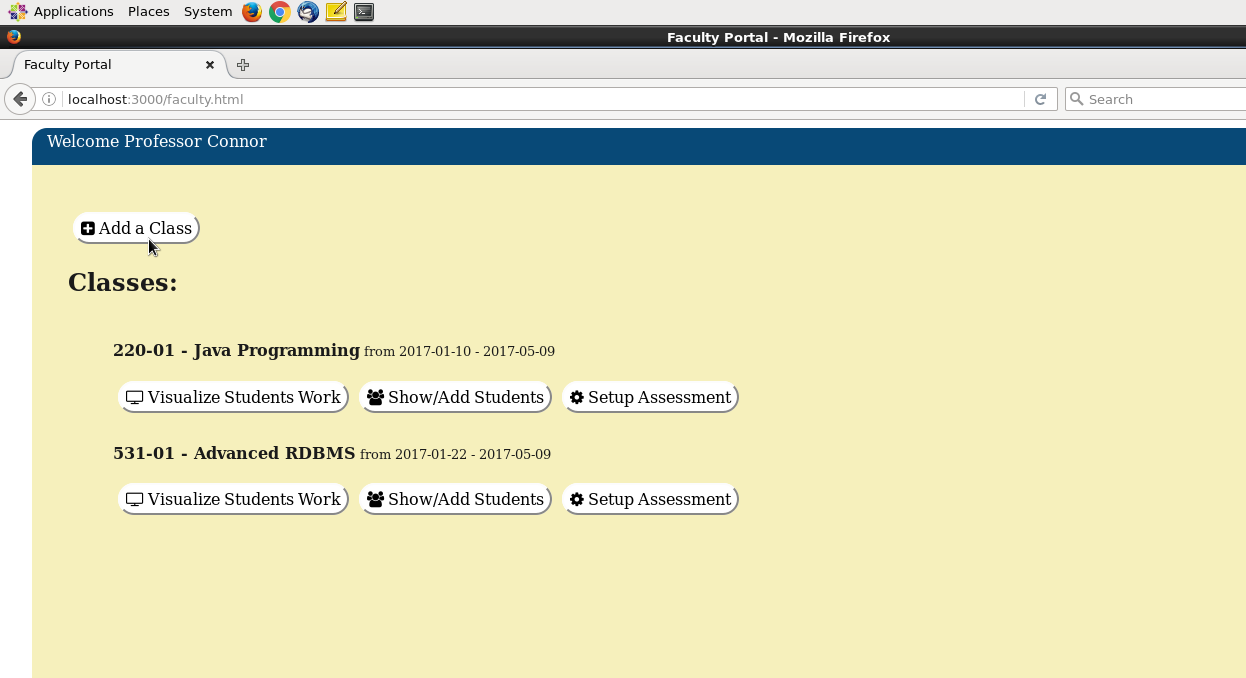
The classes page (shown in figure 5) is the first page appears after a faculty member successfully logs in. It shows all the classes added by that faculty member (in newest first order). It shows class name, number, and starting- ending date. Figure [5] gives a sample of the classes page. It can be seen that figure an example of a classes page with two classes. In the classes page, each class has three buttons below the class name, and these buttons are the visualize students work button for producing and viewing the assessment report for that class, the show/add students button to view and add students and their grades, and finally the setup assessment button which is used to add assessment indicators to that class (refer to section 6.2.4 Adding assessment indicators for more details).



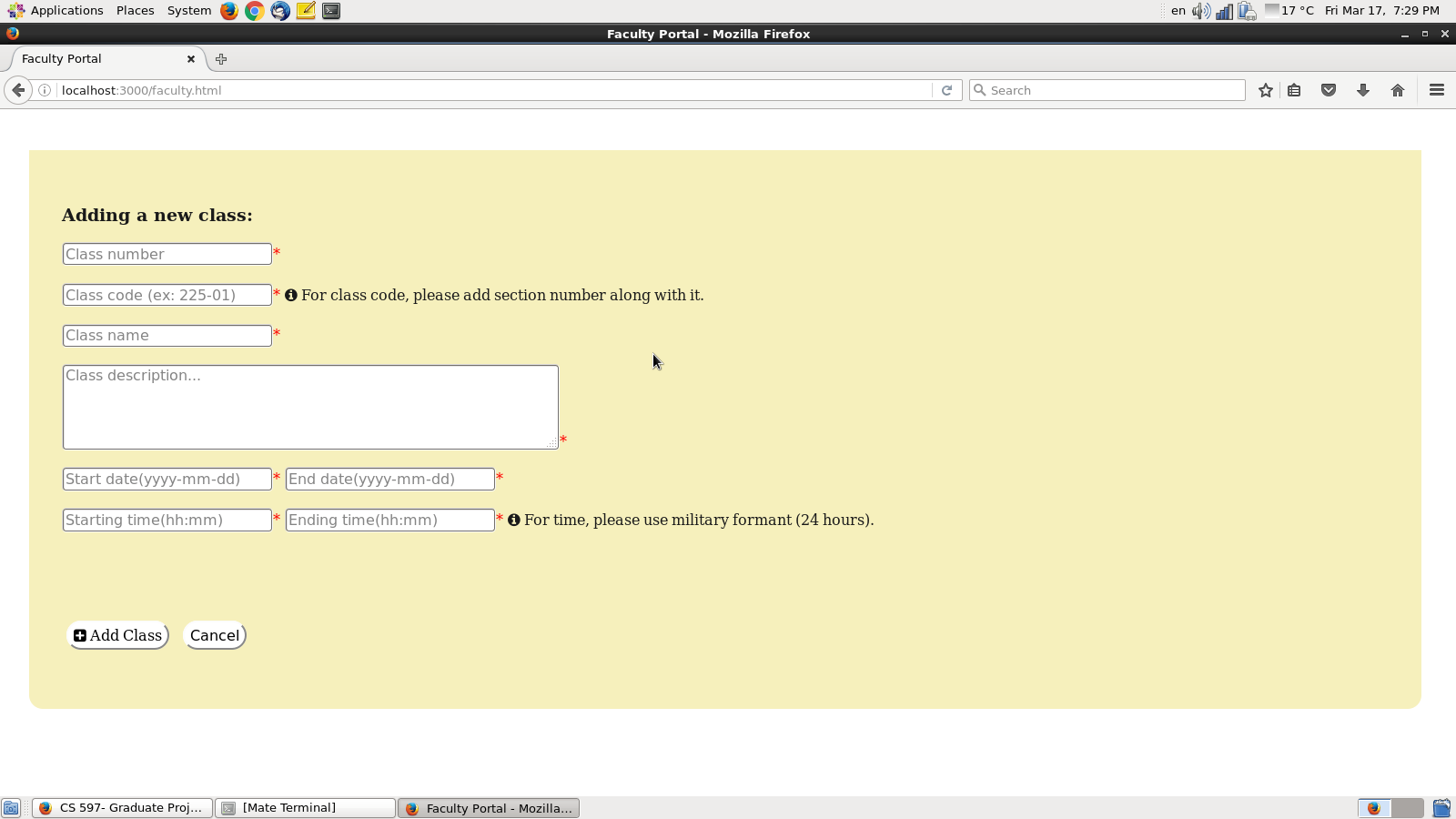
[Figure 5 - faculty classes].

### 6.2.1 How to add a new class:

Starting from the home page (classes page), click the (add class) button and the system will present a form for adding a new class. As is the case with user registration, all fields are necessary and marked with (\*). Users need to enter class related information like class number, class name, description, and date and time. Date should take the form of (yyyy-mm-dd), and time takes the military from (i.e. 24 hours). When finished, click on add button to add the class. Figures [6] and [7] are showing the class adding process. In figure 6, simply click on (Add a class) button, then the adding form will appear (as in figure 7). After that, simply full the form and click on (Add class) button (as you can see in figure 7).



[Figure 6- adding classes]



[Figure 7- adding classes form]

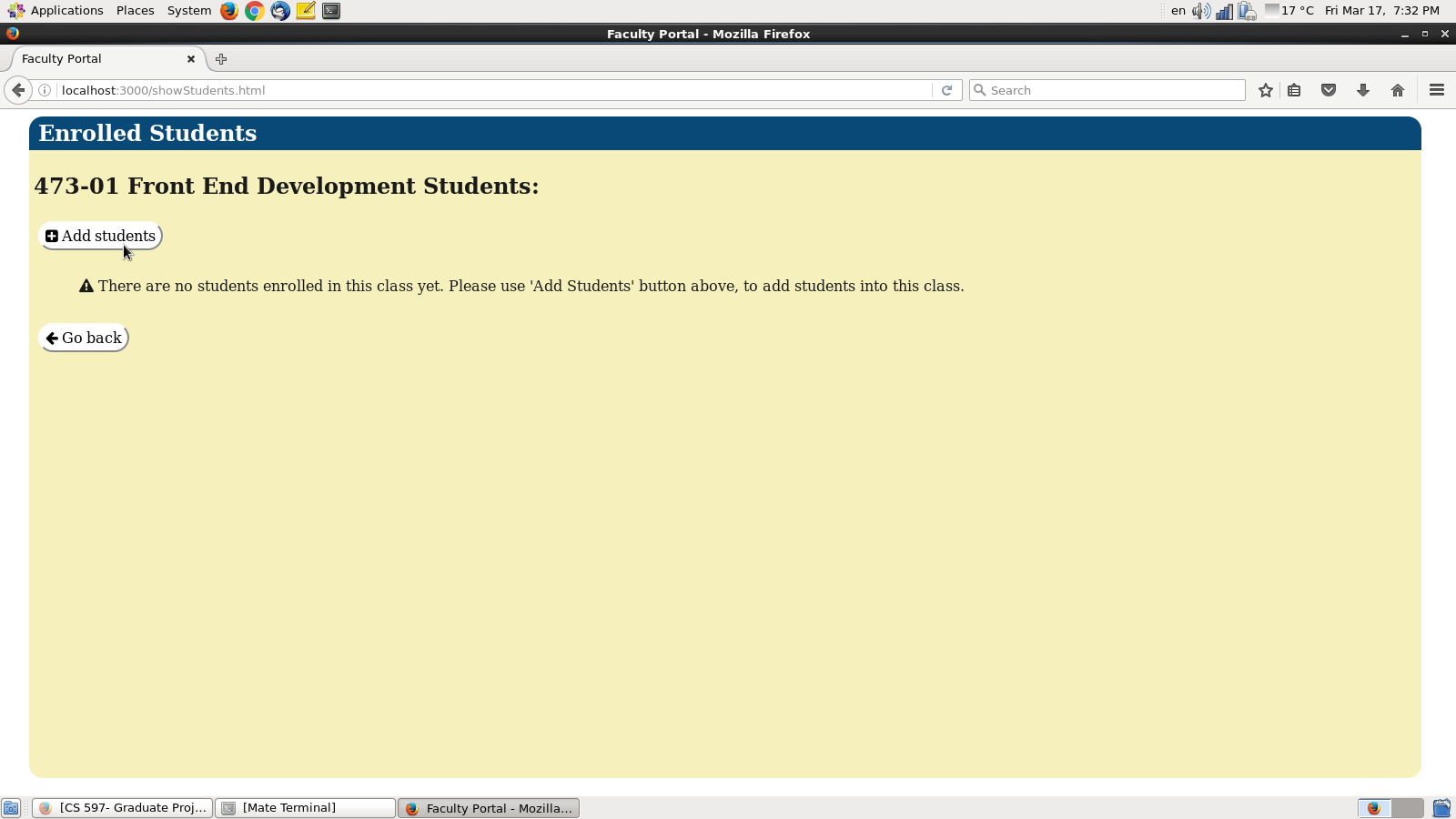
### 6.2.2 How to add students to new created class:

From the classes page (figure 5), click on (show/add students) button. After that, a page similar to figure 8A will appear, then simply click on (add students button) to add students to a newly created class. In the process of adding students, the user needs to have a CSV (comma separated values) file which is formatted in the following manner:

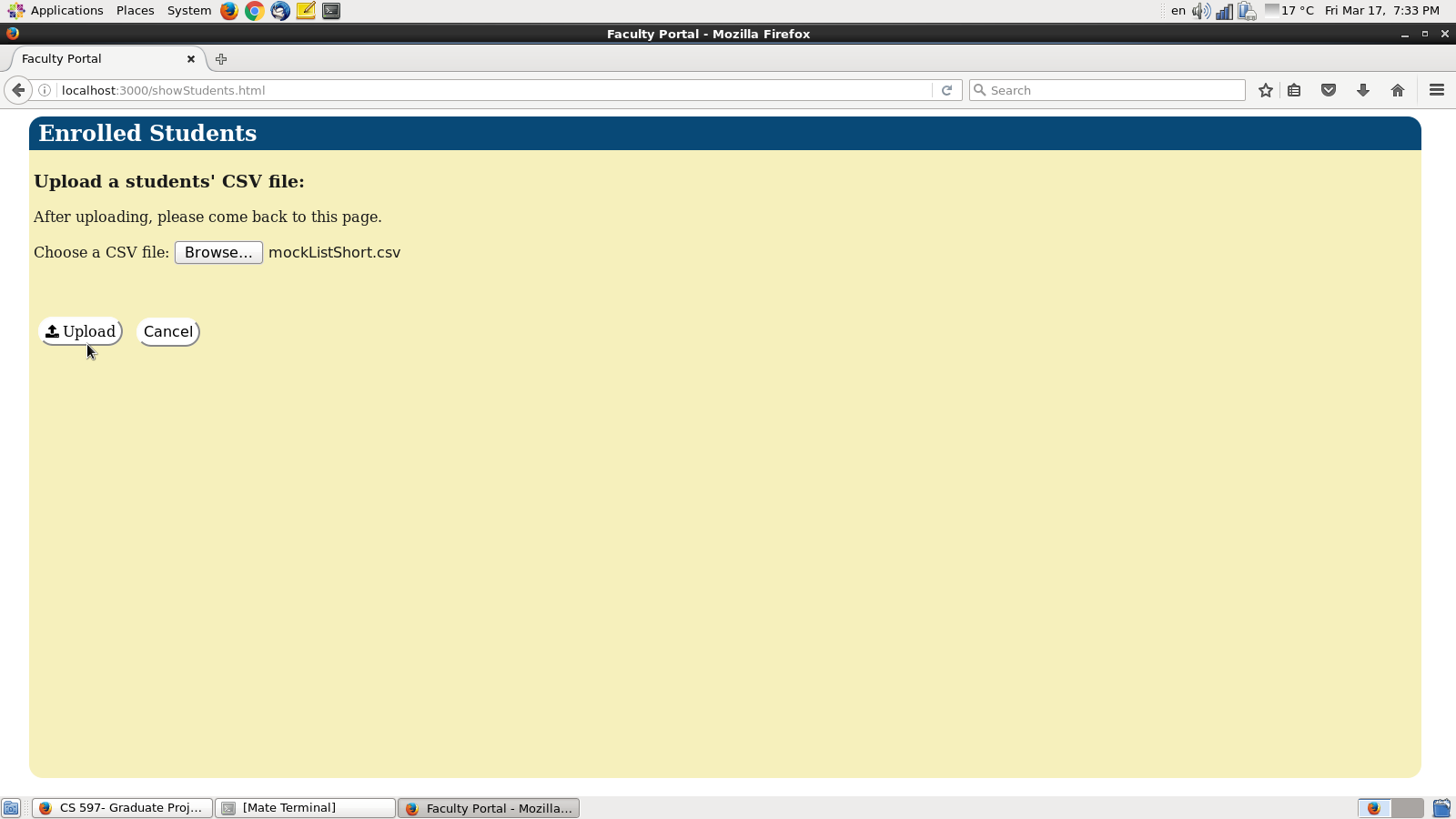
[student id, student first name, student last name, student email, ..., ...]

Note: The CSV file has to have six data fields. However, not all six fields are required to have values in them, only first name, last name, and email are needed, and the rest can be left with trivial input like (“-”, “N/A”, “null”, “0”).

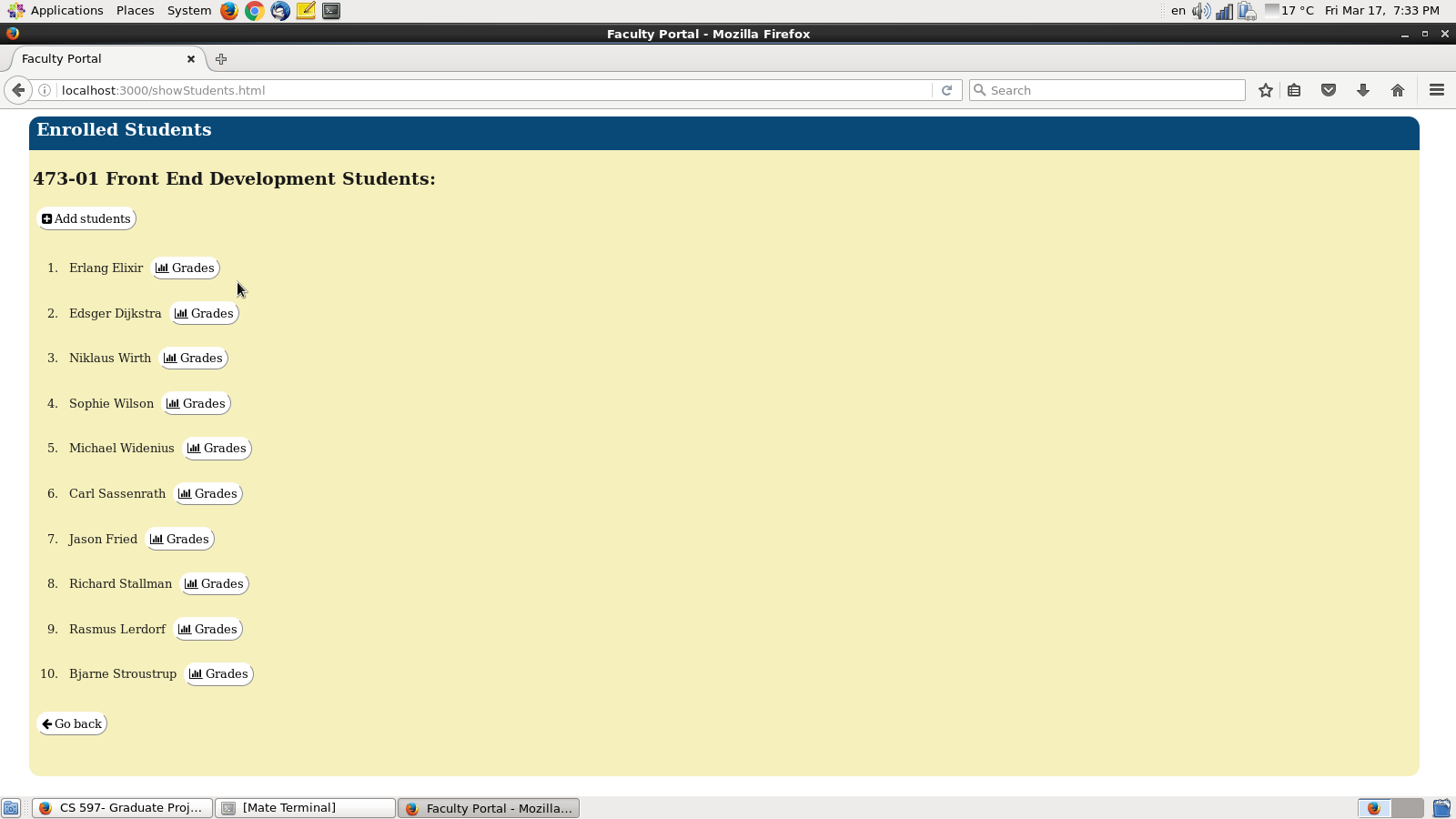
Once the files are selected and successfully uploaded, the new class will be populated with students as shown in figures 8A, 8B, and 8C.



[Figure 8A- adding students]



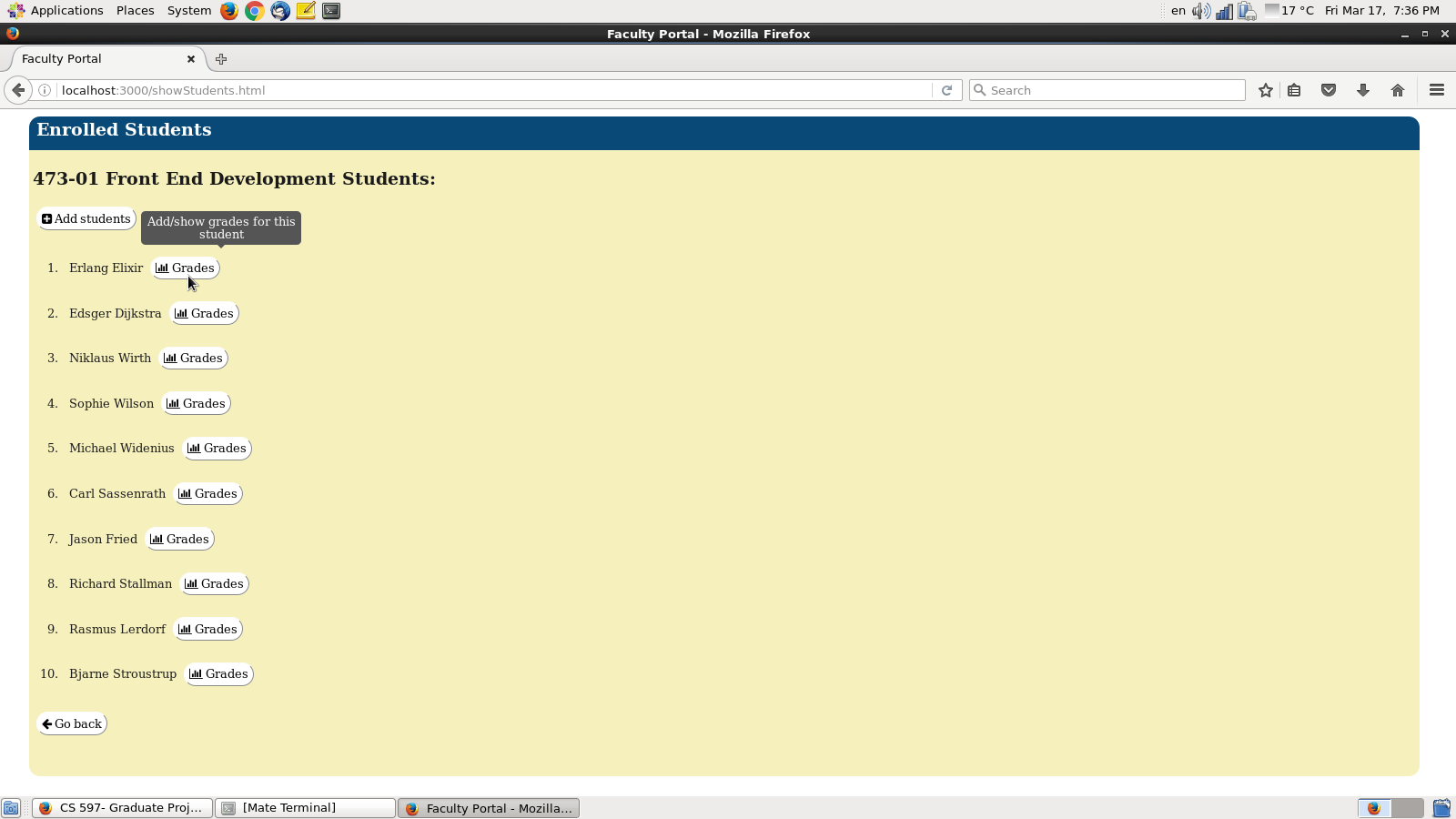
[Figure 8B-Adding students]



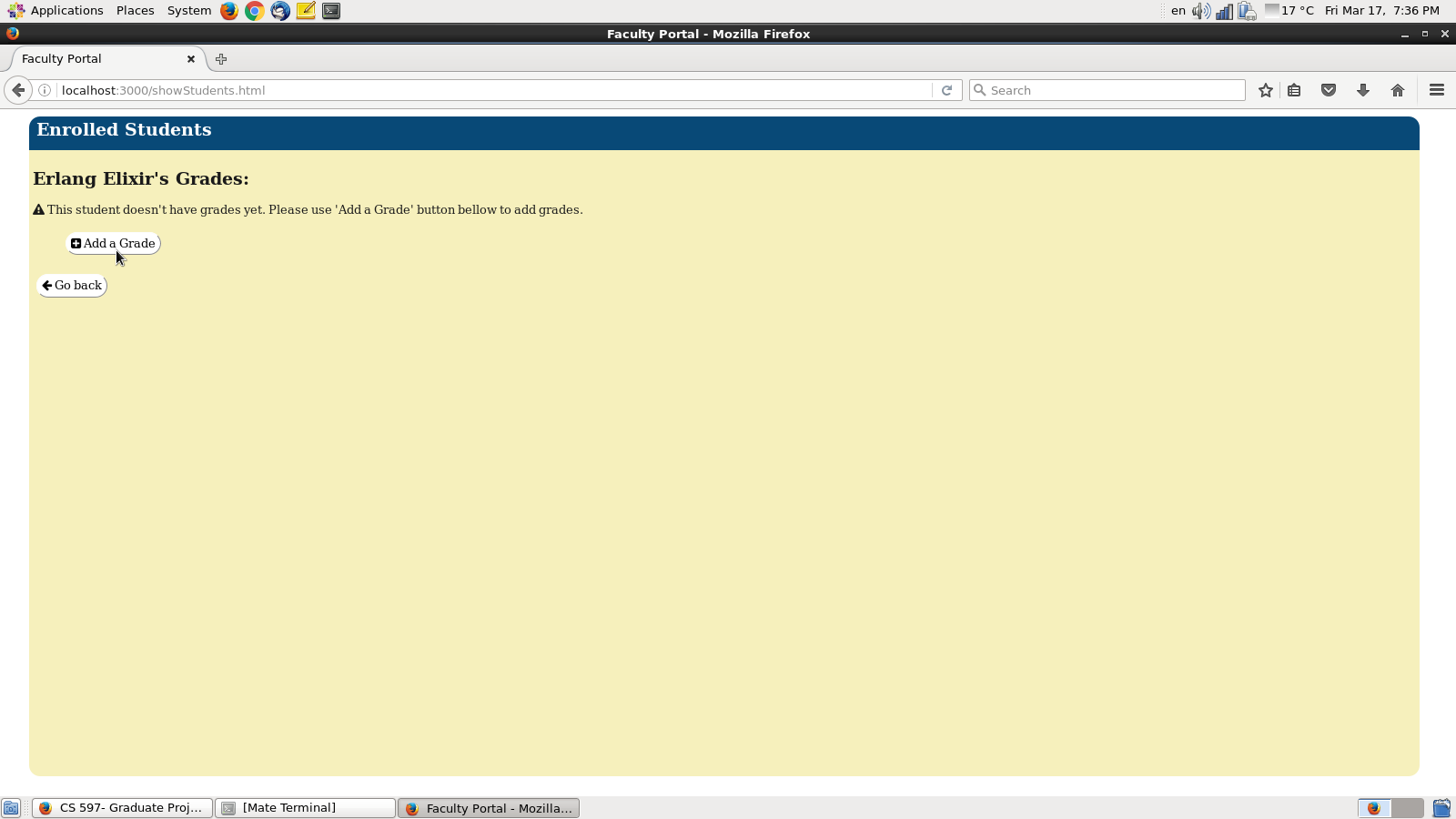
[Figure 8C- new class with students]

### 6.2.3 Adding students’ grades:

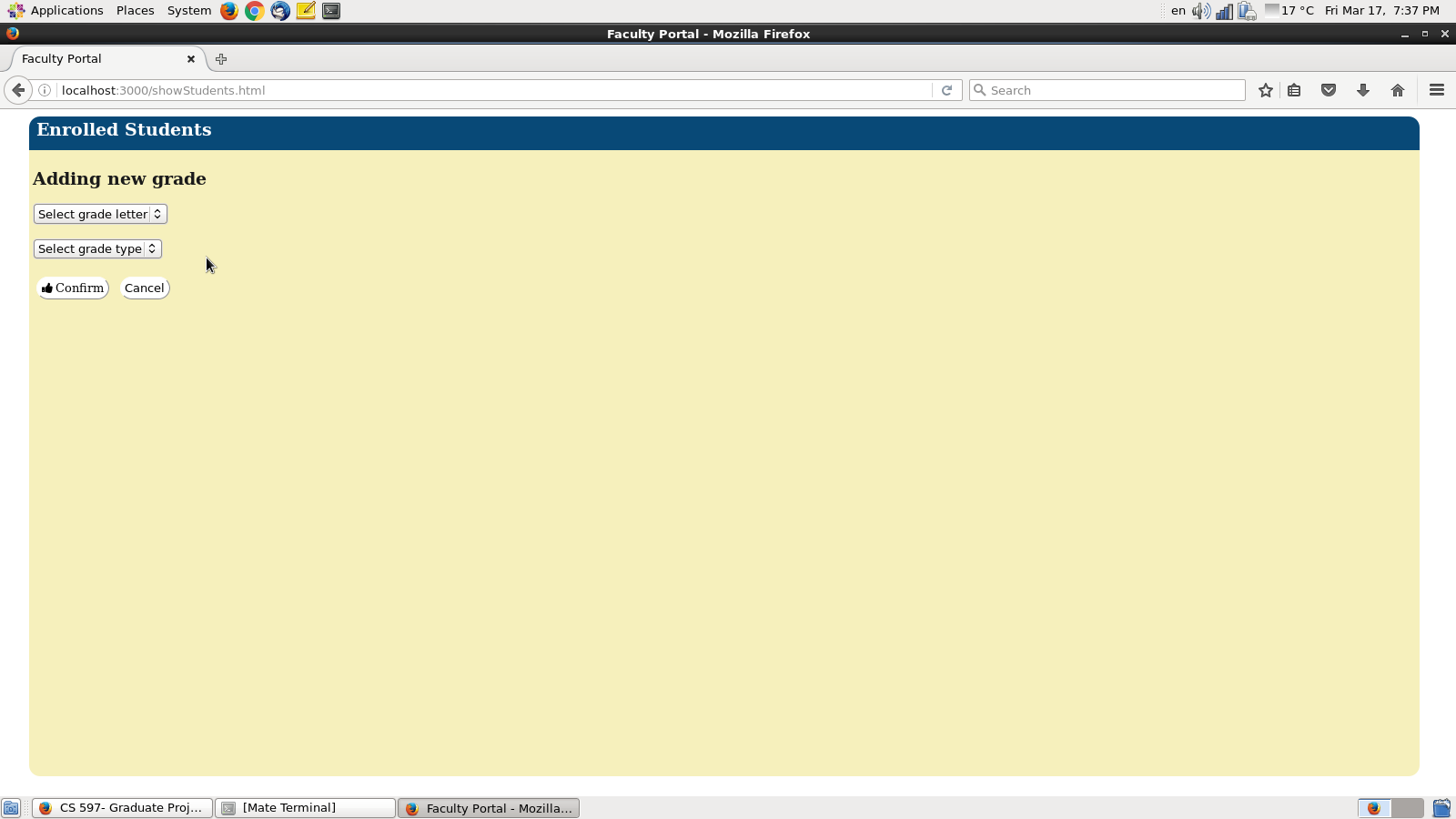
After successfully populating a new class with students, the next step is to add grades to students. Starting from the students list page figure [fig 8C], click on (add grades) button. After that, the adding grades form will appear. Select the grade (A, B, C, D, E, F), then select the type of work, and finally click on (add grade button). The figure below presenting the process:



[Figure 9A]



[Figure 9B]

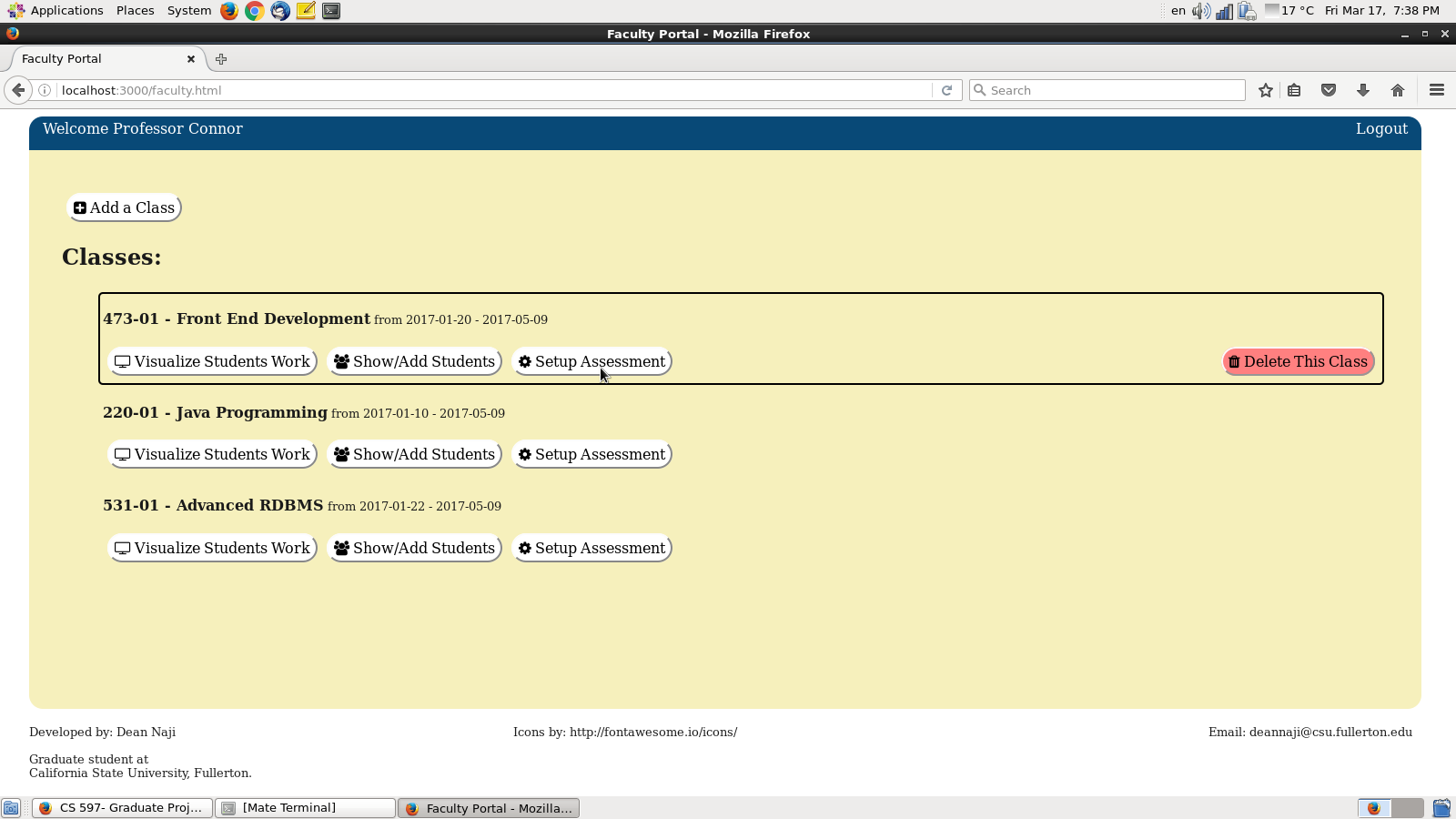


[Figure 9C]

[Figure 9A, B, C Adding Students]

### 6.2.4 Adding assessment indicators:

After finishing the process of adding grades, the next important step (on the road to complete the assessment report) is to add assessment indicators. Assessment indicators are specific benchmarks for measuring students’ performance. For each class, the user needs to add a number of indicators (depending on the assessment requirements of that class), then upload files as students’ work samples. For each indicator, the user needs to upload three types of files: satisfactory, developing, and unsatisfactory. To add an indicator, start from the classes page [figure 5]. For the desired class, click on the (setup assessment) button as shown in figure 10A. Then, after the indicator adding dialog shows up, fill all the fields (indicator name, type of work, and select work files) and then click on the (add indicator) button as shown in figures 10A, 10B, 10C, and 10D. In figure 10B, a user can add “blank” indicators by simply clicking on (Add indicator) button. After that, each indicator needs to be populated with data. To do so, simply click on the (Setup indicator) button for each blank indicator, as shown in figure 10C. As a result of that step, a setup form will show up as in figure 10D, from that form you can use drop down lists to add indicator name and type of work, and finally upload three sample student’s work files that falls into the three previously mentioned categories (satisfactory, developing, and unsatisfactory). Finally, click on the (save changes) button to add that assessment indicator and save its data.



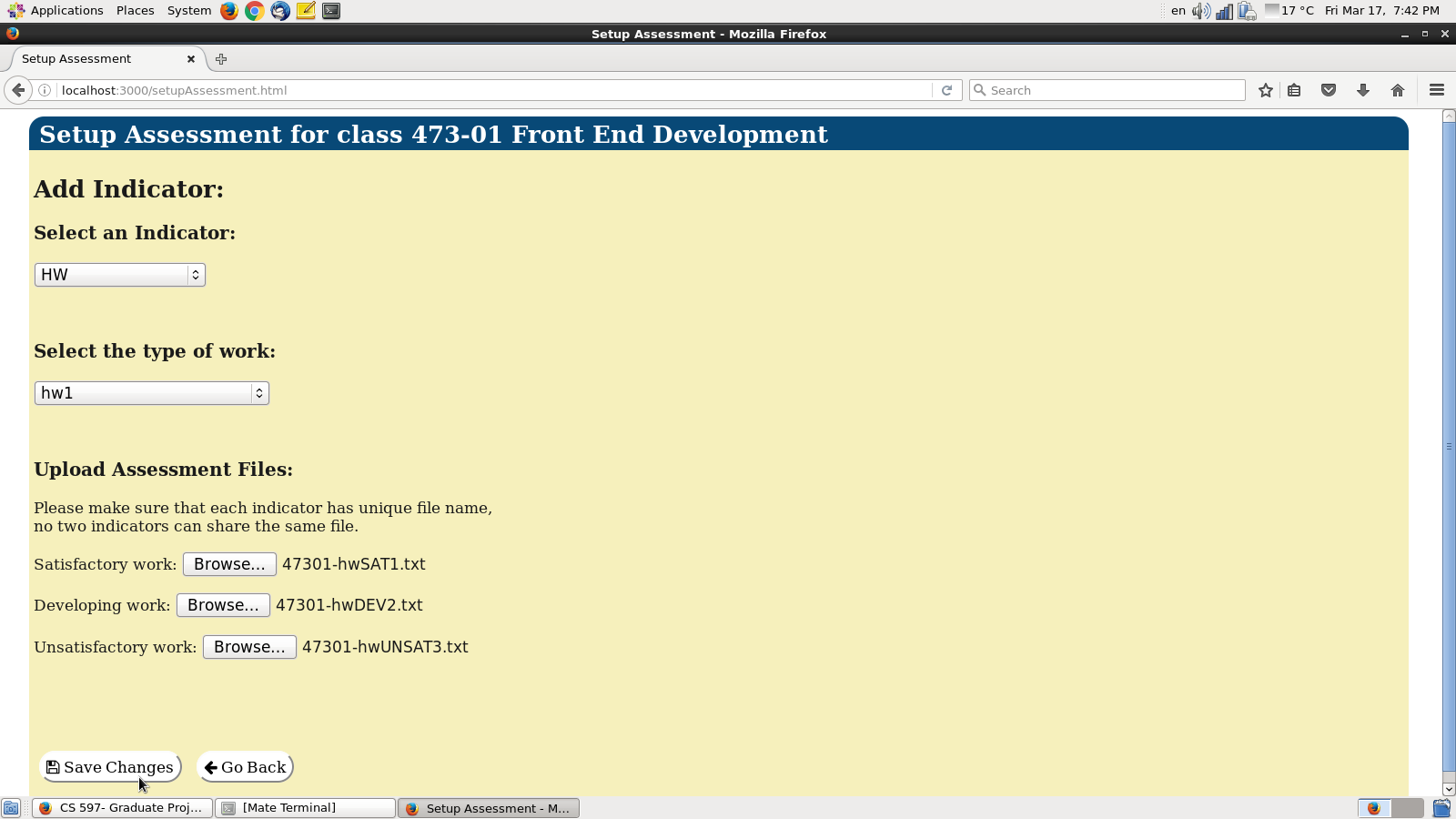
[Figure 10A]



[Figure 10B]



[Figure 10C]



[Figure 10D]

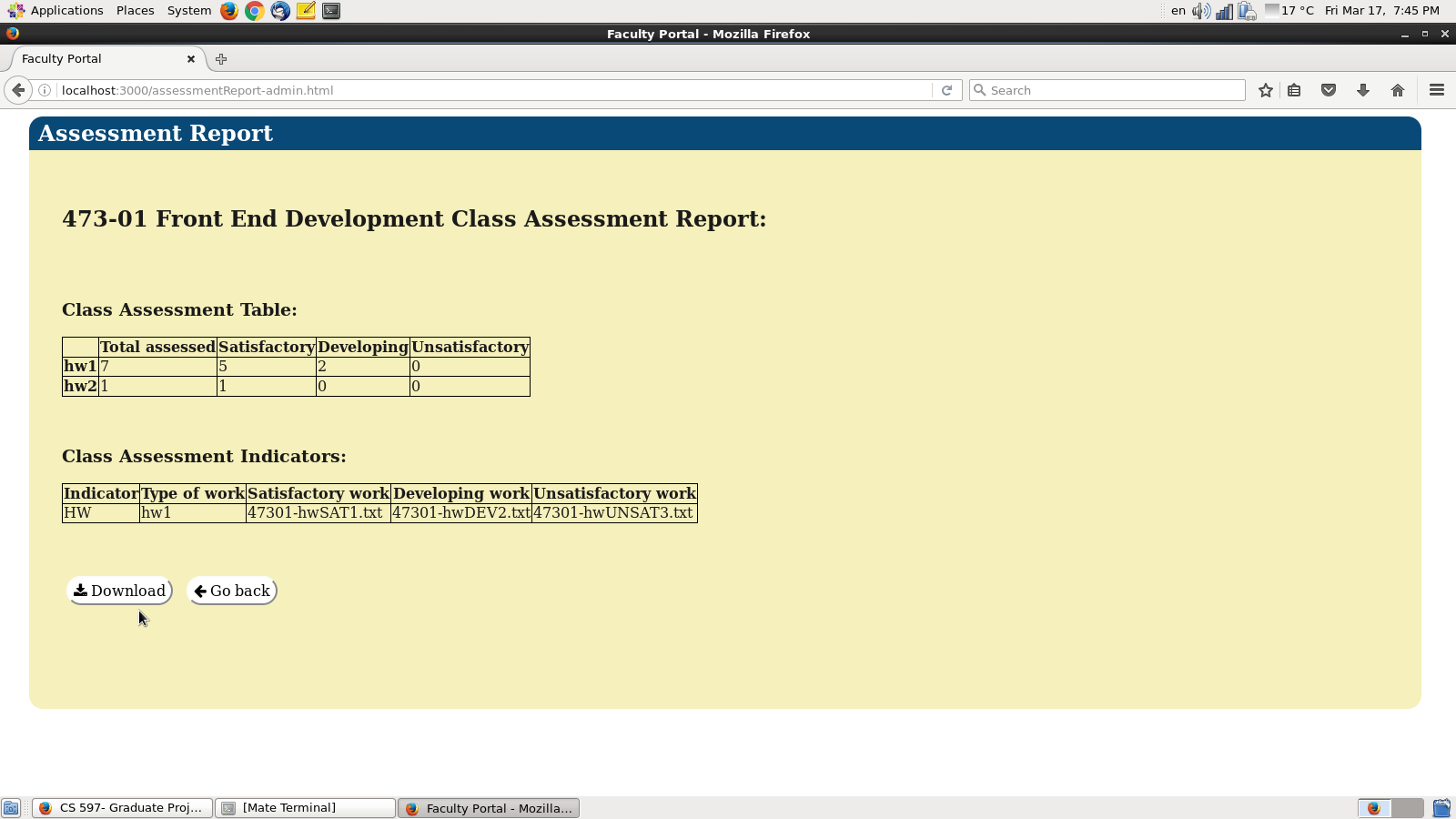
[Figures 10A, 10B, 10C, and 10D adding indicators]

### 6.2.5 Displaying assessment report:

To display the assessment report for a certain class, click on the (assessment report) button, [figure 5]. The system will display the assessment report for the desired class [figure 11]. The assessment report contains two sections: First, a list of all assignments for this specific class along with total students who participated in each assignment, and the number of students that fall into each assessment category (satisfactory, developing, and unsatisfactory). The second section is dedicated to displaying the class indicators along with their uploaded work files. Figure 11 below represent a sample assessment report generated by the system.

### 6.2.6 Downloading assessment report:

In order to download the generated assessment report, the system provides the ability to download the report in PDF format. Simply, click on (Download assessment report) to download the assessment report to the local machine, see figure 11.



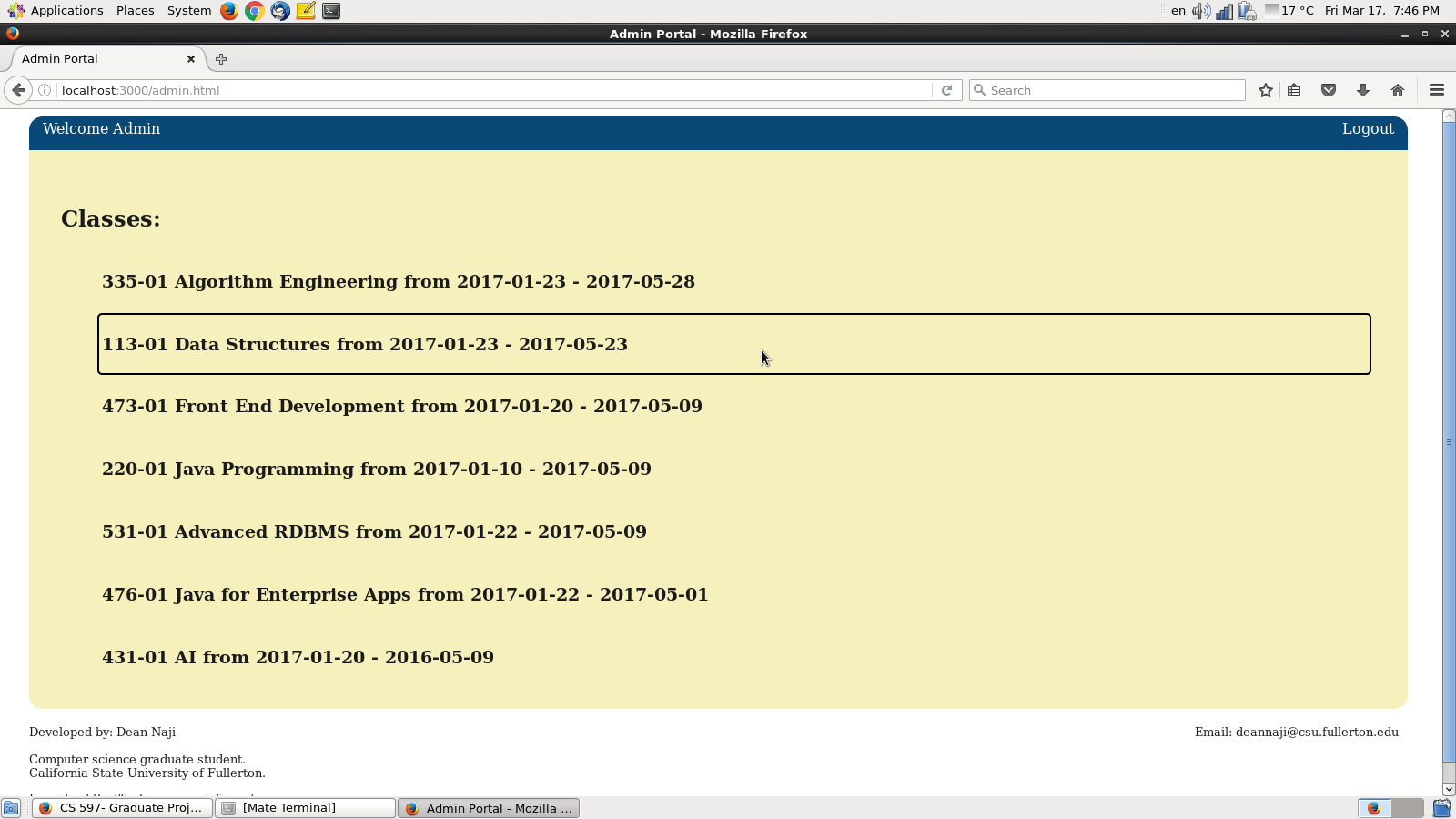
[Figure 11- download assessment report]

### 6.2.7 Downloading assessment indicators’ files:

To do so, starting from classes page and click on the assessment report for the desired class, [figure 11]. Then simply click on the desired indicator work file name from the indicators table [figure 11].

## 6.3 Admin:

When logging in as an admin, the user will be able to look at the broader scope of the department classes and assessment reports. All classes for all faculty members, which are registered users in the system, are presented on the main page, ordered by (newest-first).



[Figure 12- sample admin home page]

### 6.3.1 Displaying assessment report for certain class:

To do so, from the admin home page, just click on the desired class, and the assessment report will be displayed as shown in figure 11.

### 6.3.2 Downloading assessment report:

Please refer to (6.2.6) downloading assessment report for faculty user.

### 6.3.3 Download assessment indicators’ files:

Please refer to (6.2.7) downloading assessment indicators for faculty user.

## 7. Conclusion:

With the aid of computer software, we can automate the task of the assessment report generation. Automation can increase the ease and speed of generating an assessment report. Beyond that, it is very easy to share these assessment reports with the department, as the department can use the (admin) account to easily obtain all assessments reports. Yet another benefit achieved is that the data can be safely stored as an electronic version, rather than as hard copies. If this software is made generally available among all universities and colleges, a large volume of data will be generated. This data can be warehoused or stored in a big data repository. In the future, knowledge discovery algorithms can be applied to that data to extract possible useful knowledge, which might lead to new beneficial discoveries in the field of education.

## 8. Future improvements:

The main purpose behind this project is to provide an automated solution to the problem of conducting assessment reports by faculty that is currently performed manually. This project however is not meant to be completely perfect or to provide an optimum solution, which can be used in every university or college. Therefore, the possibilities for improvement are open. Below is a set of possible improvements that might be considered in the future or depending on future needs:

* Enhancing system’s security aspects, to provide more protection to the data being stored.
* Improving the front-end part of the project, to present more user options, or create more user-friendly design, if the need arises to a newer front-end design.
* Generalizing the project, by adding more configuration options, in order to make the system compatible with all possible cases and work perfectly with any chosen university or college.
* Taking a back-end path, to switch to a new choice of database. For example, a NoSQL database like (MongoDB) can be used instead of MySQL.
* Refactoring the current project from its current state, to adopt microservices architecture.
* Transferring the project from development phase into production phase.

# 9. Bibliography

[1] Mary‐Ann Winkelmes, Transparency in Teaching: Faculty Share Data and Improve Students’ Learning Association of American Colleges & Universities, 02/11/2016,

https://www.aacu.org/publications-research/periodicals/transparency-teaching-faculty-share-data-and-improve-students

[2] Mary‐Ann Winkelmes, Thriving in Academe, NEA and the Professional and Organizational Development Networking Higher Education,

http://www.nea.org/assets/docs/HE/1301Advocate\_PDF%281%29.pdf

[3] Jo Beld, Mary Walczak, Dana Gross, Engaging Faculty in Department-Level Assessment, AAC&U Engaging Departments Institute, July 2009,

http://www.teaglefoundation.org/teagle/media/library/documents/resources/Beld\_departmentlevel-assessment\_1.pdf

[4] Sherry Lee Linkon, How Can Assessment Work for Us, July - August 2005,

http://www.assessment.uconn.edu/docs/resources/ARTICLES\_and\_REPORTS/Sherry\_Lee\_Linkon\_How\_Can\_Assessment\_Work\_for\_Us.pdf

[5] Northern Illinois University, Faculty Development and Instructional Design Center,

Formative and Summative Assessment,

facdev@niu.edu ,<http://facdev.niu.edu> , 815.753.0595