**PROJECT**

**4th Increment Report**

**on**

**“CLASS SCHEDULER”**

**By**

**Sumanth Koushik Kalli**

**Alekhya Boyapati**

**Savya Pathuri**

**Importing Existing Services:**

The existing services in this increment are implemented using HTML language, JavaScript, cascading style sheets (CSS) and PHP in making a good user interface. Also, we have tried to implement the class scheduling using a genetic algorithm approach. We have used the \_\_\_ Database in our project to store details about the course’s involved in a particular semester such as the Course ID and Course Name, Professor’s name and his research work, timings allowed according to the university policy. In using the genetic algorithm, we found that chromosome is most important in bringing out the output and in our case we found that characterizing the professor’s based on their research helped us in choosing on which set of day/days he/she can take a class. Using this type of approach we could also eliminate the possibility of class overlapping.

**Detail Design of Services:**

In this web application, we have created separate services for the manager and professor to login and see the schedules respectively. The manager will have many services such as adding a new course, Adding a professor and assigning a course to professor, adding a pre requisite course, limiting the timings for a schedule and Generating the schedule.

**Adding a New Course:** In this service, the manager will be able to add new courses to the Class Scheduler for each semester if any new courses come up by entering the name of the course and course ID.

**Adding a Professor:** Whenever a new faculty member comes in, the manager will be able to him to Class Scheduler so that he gets his classes scheduled. Also, a professor’s research work is also taken into consideration because that allows us to build an efficient algorithm for scheduling as per the university policy of class scheduling. While adding a new professor, the manager has to give the email address of the professor and his login credentials can be seen both by the manager and the professor initially and later on the password can be modified by the professor.

**Assigning Course to Professor:** At the beginning, all the courses have to be assigned to their respective professor by selecting them from the drop down menu.

**Timings:** From the University Class scheduling policy, we have divided the entire week into two batches as Mon/Wed/Fri and Tue/Thurs. All the timings mentioned in the Classroom scheduling policy have been added to the class scheduler for scheduling. We have also taken care so that the schedule generated will not interfere with the rules and regulations of the University policy.

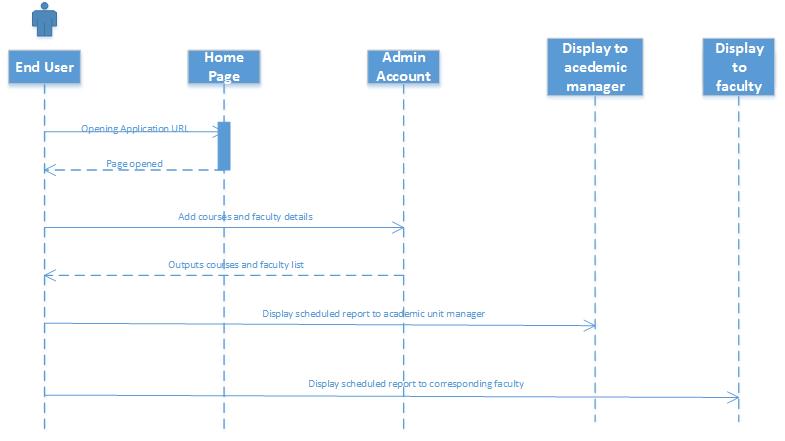
**Check Schedule between particular time:** The manager can check the schedule between two timings by selecting the two time’s from the drop down menu.

**Generate and Check Schedule:** After feeding all the data and course information, the manager can generate the schedule for the semester. Once the schedule is generated, the professor’s can check their schedule individually online by logging into their accounts.

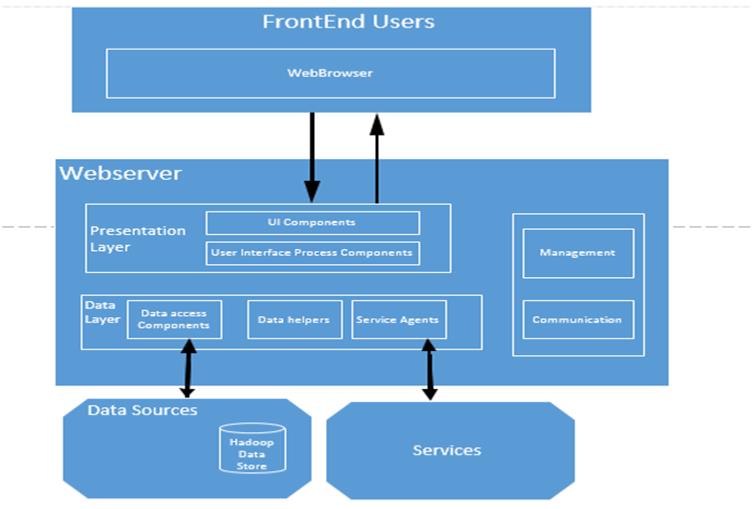
We have used the MySQL database in this project and “phpmyadmin” tool to manage the tables inside database. We have created different tables for courses, professor’s, courses being assigned, and timings for the schedule.

**Sequence Diagram:**

The below diagram shows the sequence diagram of our application.



**System Architecture Diagram:**



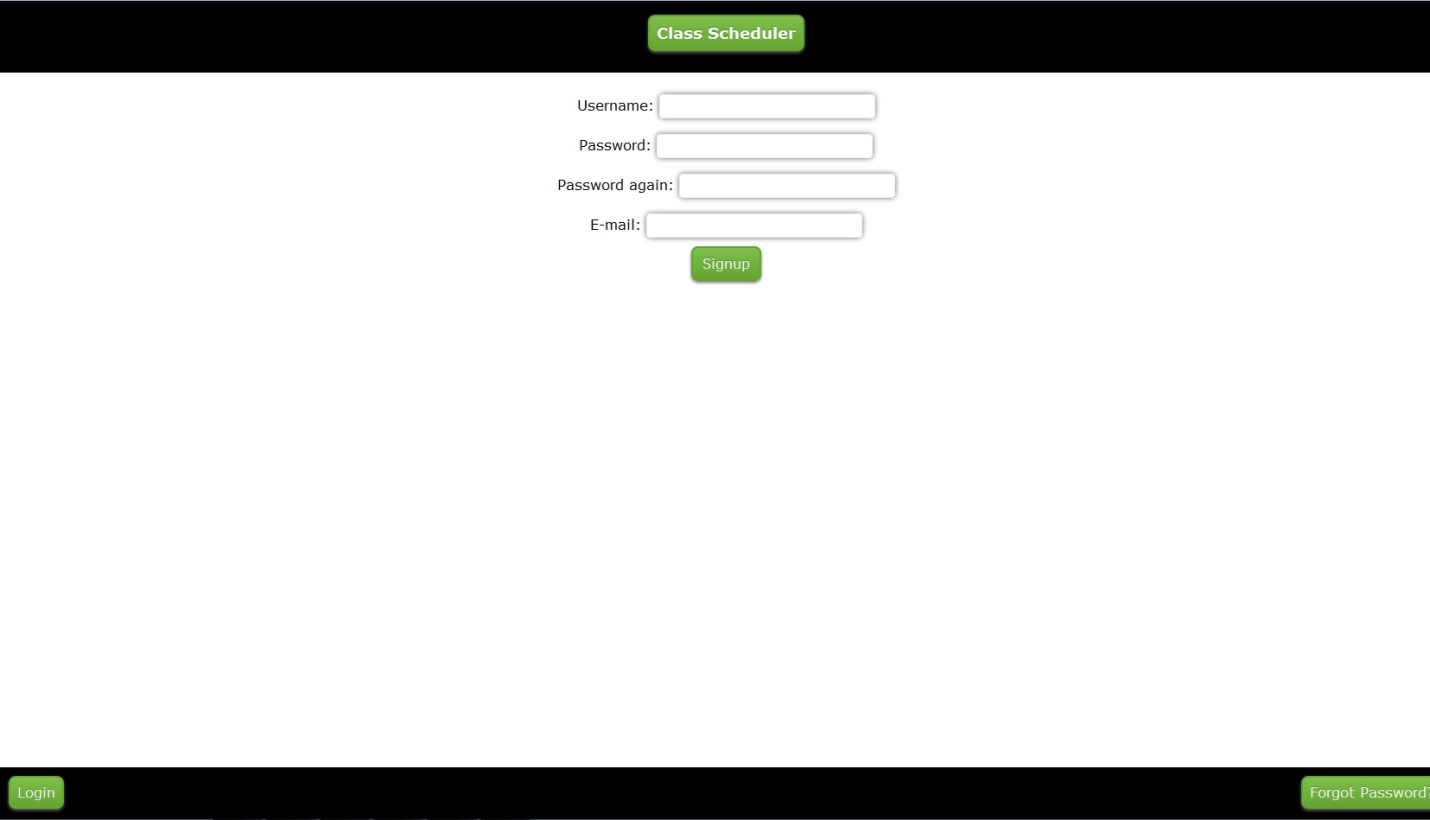
**Implementation**

**Implementation of services:** In this project Class Scheduler, we have implemented a number of different services which helped us in generating the schedule as we planned. Initially, we have created an account for the manager and added all our services such as Adding Courses, Adding professors, assigning courses to professors and timings according the University Classroom Scheduling policy. Later on, we went on developing the scheduling algorithm which could bring out two separate schedules for Mon/Wed/Fri and Tues/Thurs.

**Implementation of user interface:** All the services have been implemented using HTML, CSS, Javascript and PHP. Also, we have made use of MySQL database in this project.

**Report:**

1. In our class scheduler application the manager have to create an account with details like username, password, confirm password along with email address.



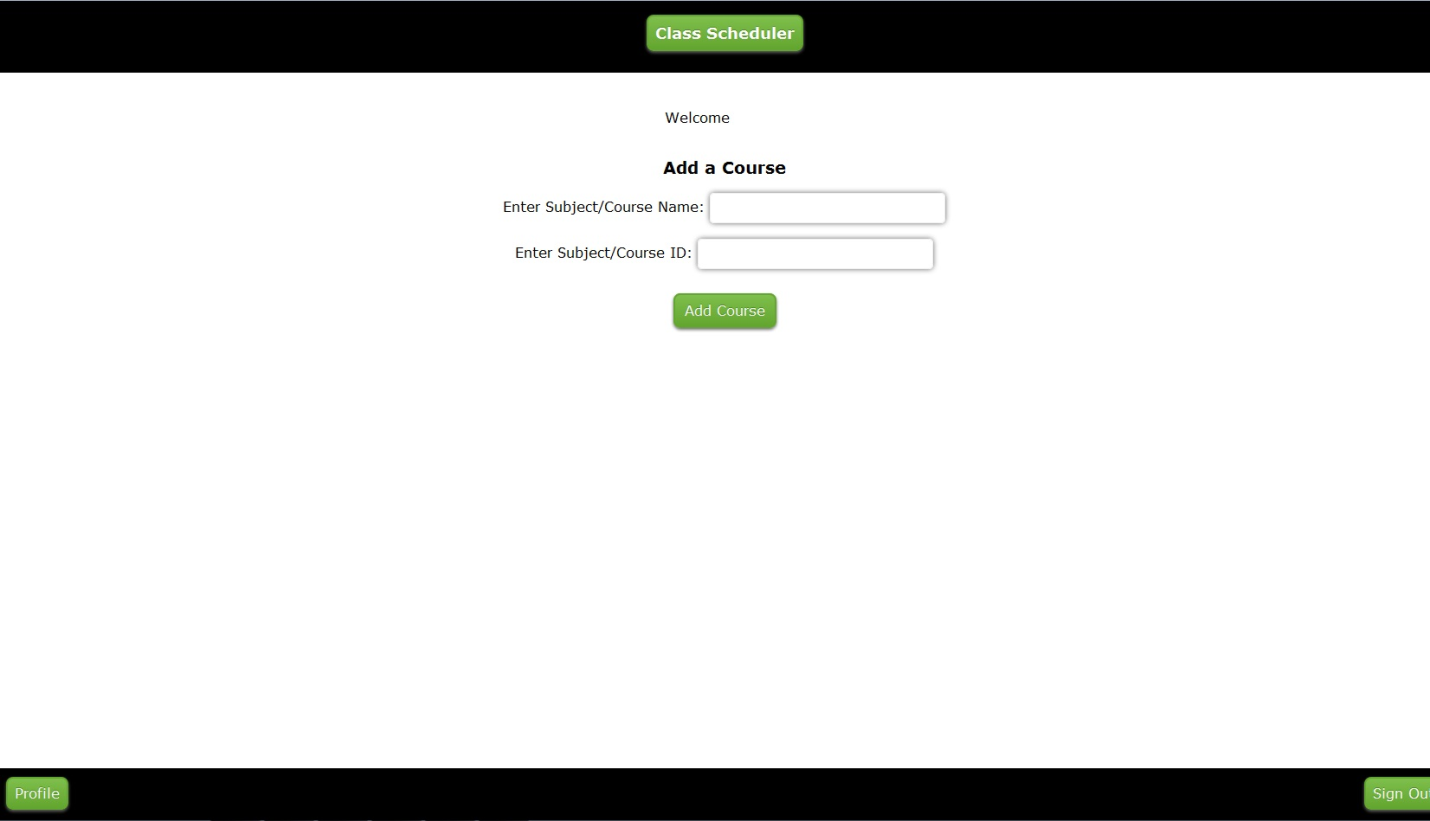
1. Then it displays a message to manager saying “successfully registered”.



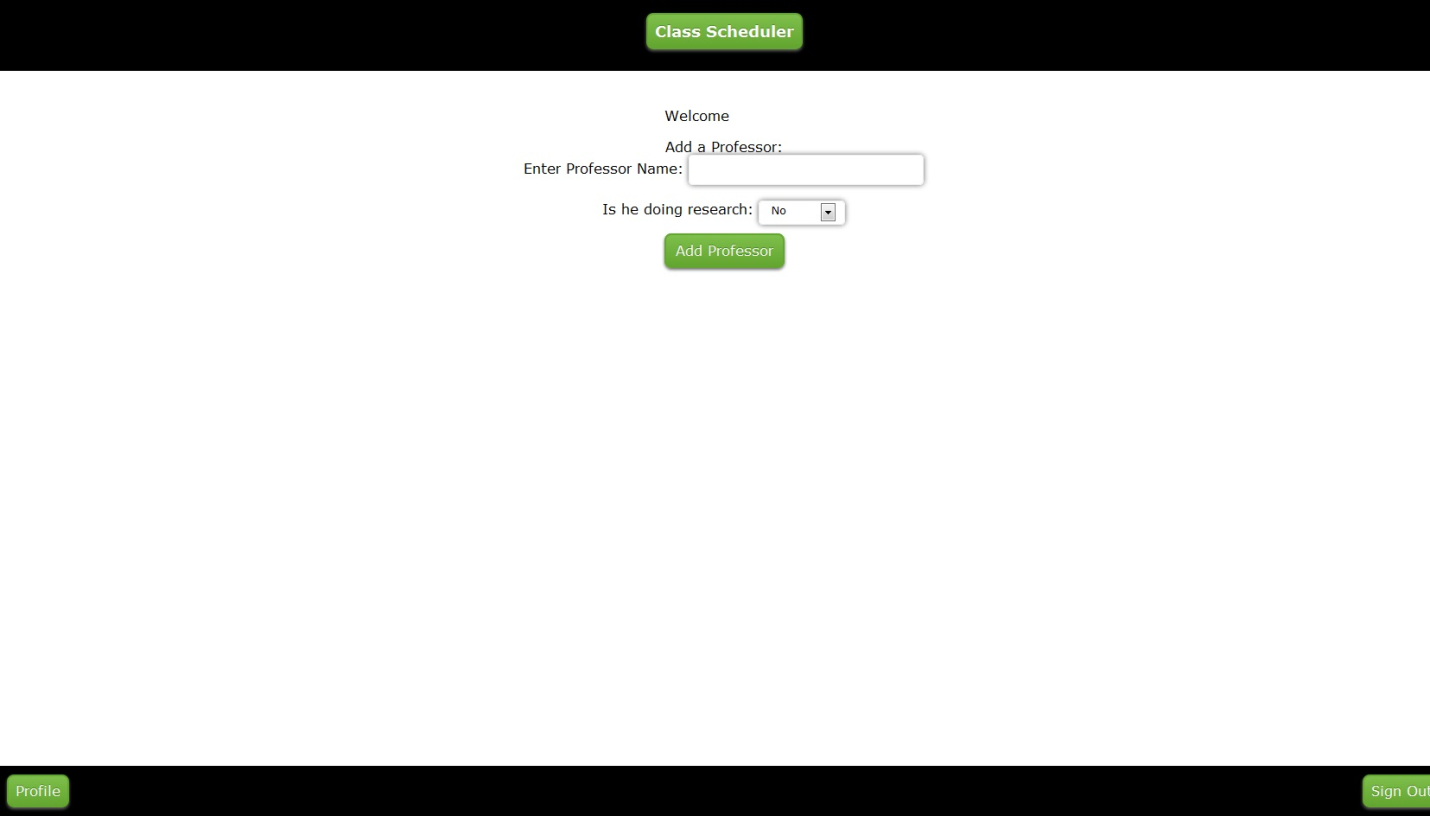
1. The application will now display all the available options like adding course and professor, assigning the particular professor to course, adding pre-requites to the corresponding courses. The manager can also generate the schedule and check it for a particular time interval if needed.



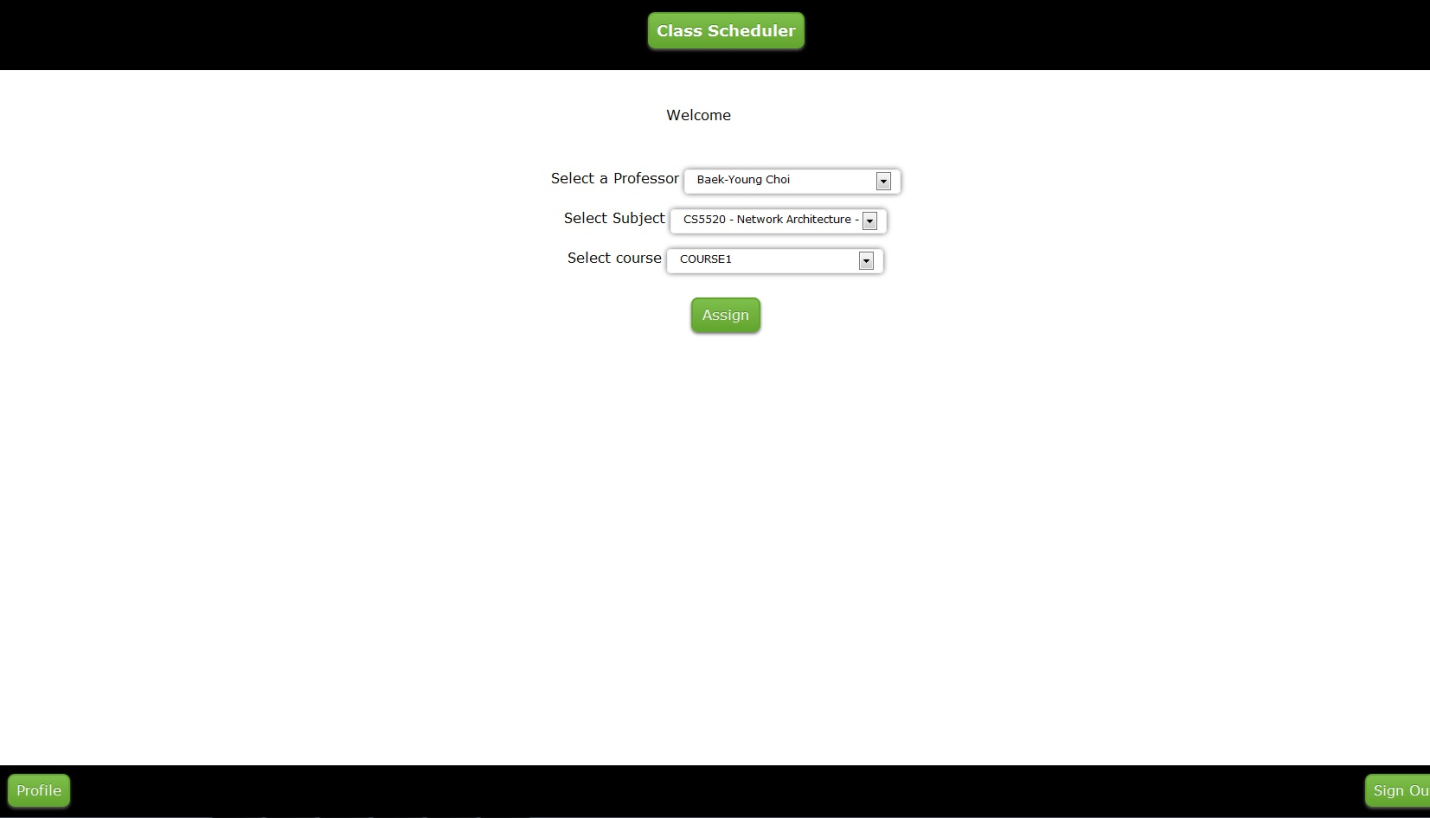
1. First, the manager has to add all the courses planned to be taught in a particular semester by clicking the “Add subject/course” option and it gives us option to enter the course name and course ID. By selecting the add course tab the course will be successfully added to the list.



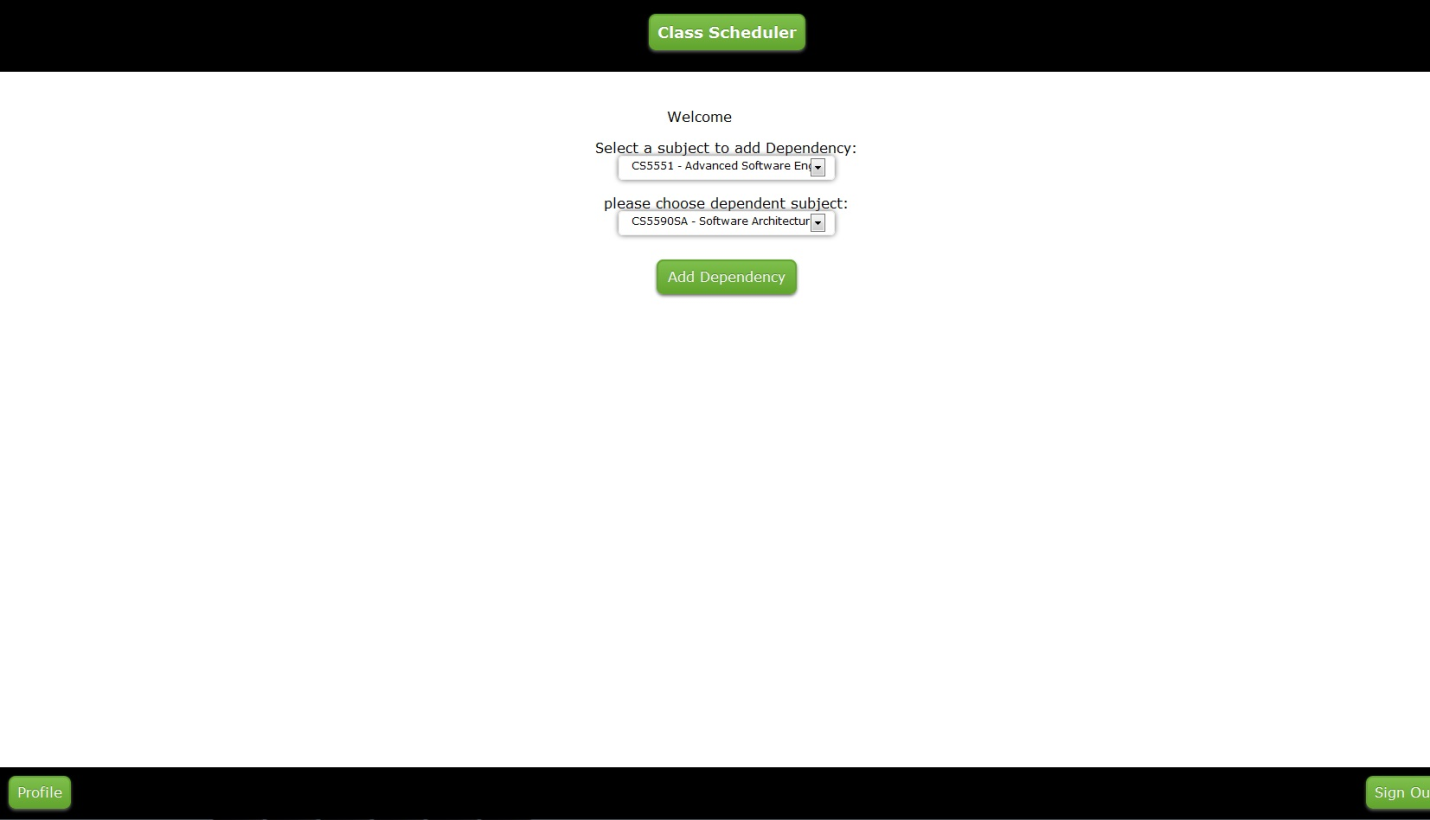
1. For adding the professor to the list, we have to enter his/he information like name and also have to select whether the professor is doing an active research or not.



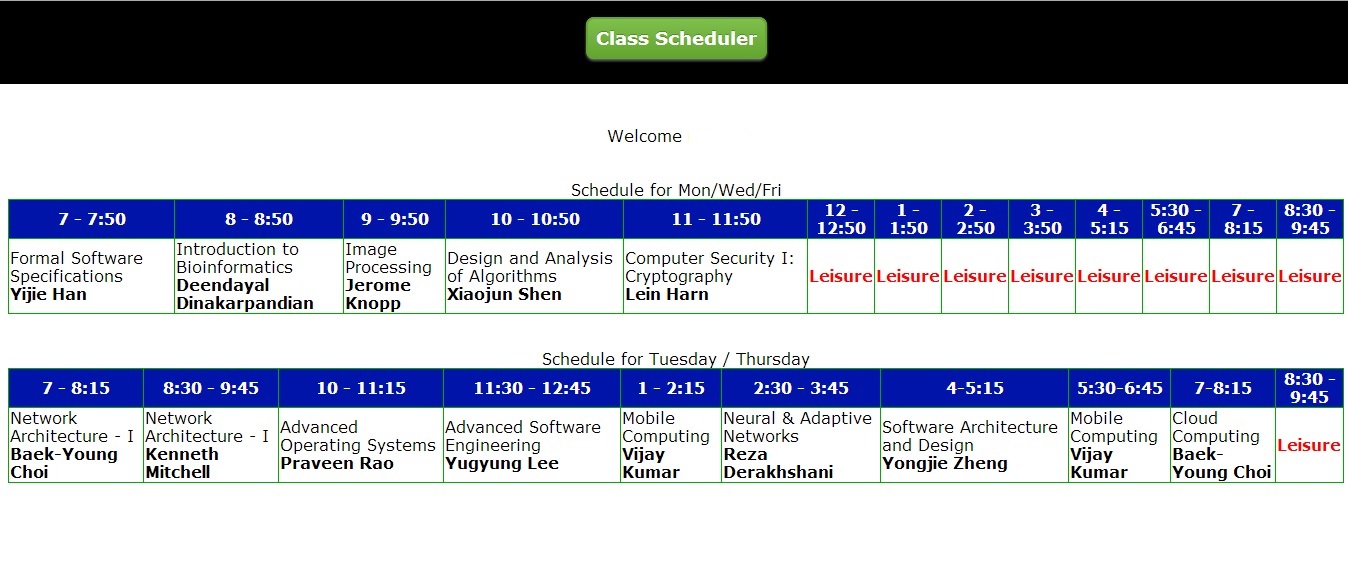
1. We can also assign a particular professor to a corresponding course. In order to that the manager has to enter the professor and the related course he/she teaches.



1. To add the dependency courses the manager has to select the main course and its corresponding dependent course.



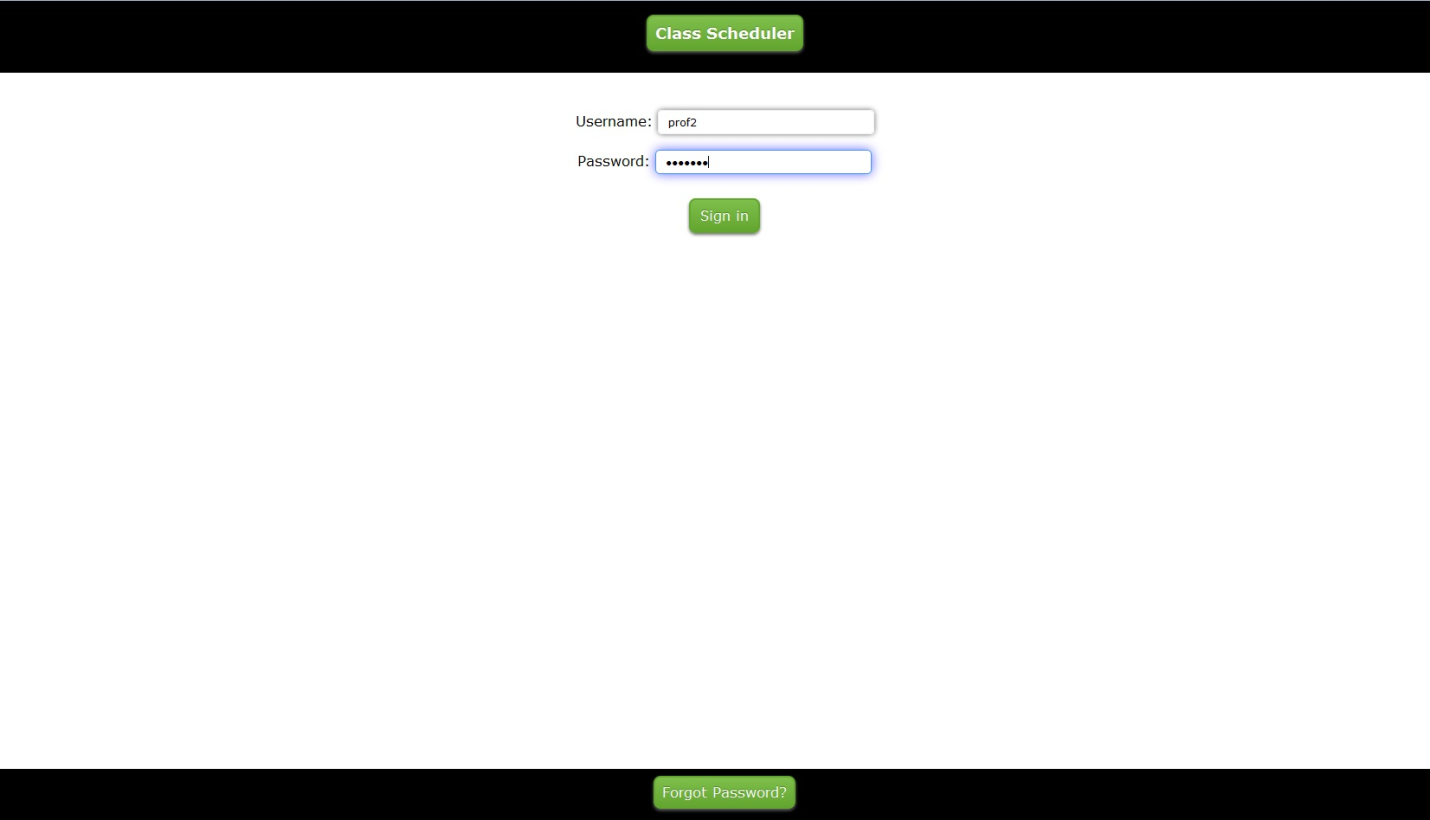
1. After adding all the information and data into the Class Scheduler application, the manager can generate the schedule for that particular semester by clicking Generate & Check Schedule as shown in the below.

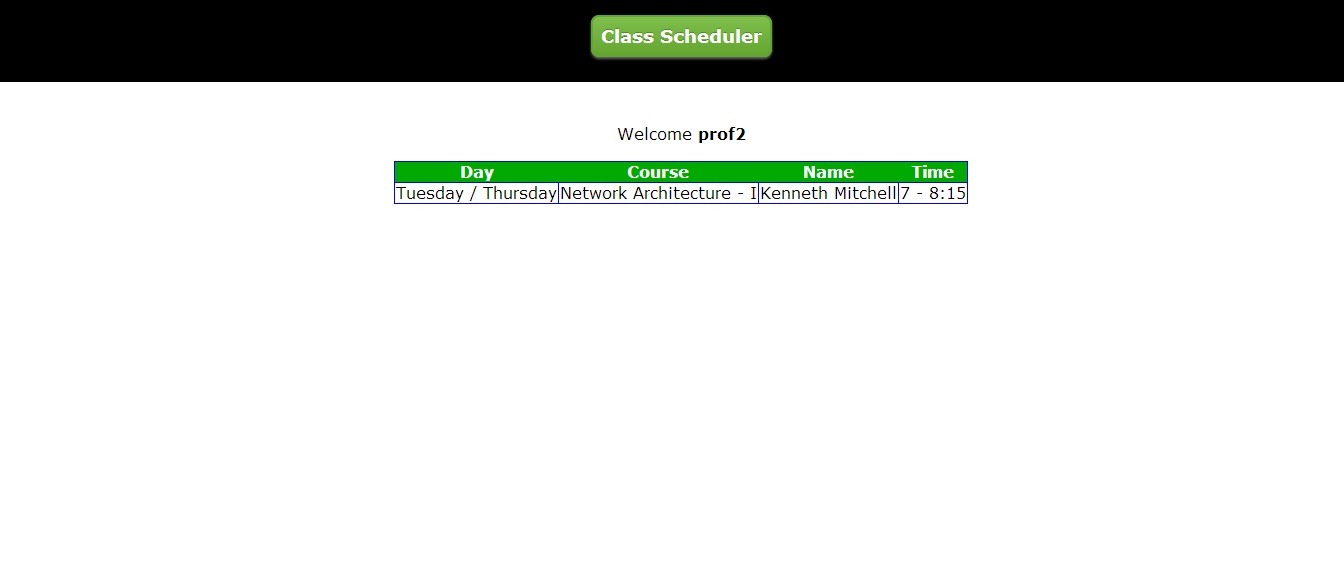


9. The manager also has the option to view the schedule for a particular time interval like 8 AM to 8 PM.



1. Whenever the manager adds a new professor, the manager and professor gets the username and password to login into the class scheduler. The username and password will be sent to the professor automatically when he/she is added on to the class scheduler. The professors can also maintain their own account and can login to view his/her schedule.





**Implementation Status Report**

**Work Completed:**

* **Description:** In the final and fourth increment, we completed the entire project and also extended the algorithm for the entire semester after the third increment.
* **Responsibility:**

**Algorithm:** For developing the algorithm, we have studied many other scheduling procedures in internet and found that the best solution to the scheduling problem can be achieved through the genetic algorithm. After that, all of us played equal role in shaping this algorithm with flow diagram and the implementation in terms of code has been shared by all of us equally.

**Database Installation:** Database installation over the server has been done bydone by

Sumanth Koushik and Alekhya.

**Collection of Data:** The professor’s research details are details are collected by

Savya and organized and analyzed by Alekhya and Sumanth.

**Documentation:** Documentation for the 4th increment has been prepared by Alekhya

Boyapati, Sumanth Koushik Kalli and Savya Paturi.

* **Time Taken:** Individually, we have spent 80 hours each towards this increment.
* **Contributions:**

**Sumanth:** Algorithm building and documentation.

**Alekhya:** Algorithm building and documentation.

**Savya:** Algorithm building and documentation.

**Future Work:**

This project Class Scheduler has larger scope for future work to be done like making dynamic changes to the schedule if there are any issues with the professor’s availability or classroom availability etc. Also, we can extend it in such a way that professor can accept or decline and request changes to their schedule to the manager and the acknowledgement has to reach the manager.

**Issues/Concerns:**

The major concern we had in our last increment was on finding a solution to adding pre-requisite courses and it was solved by linking it with the course that requires as a pre-requisite.

**Deployment:**

**Scrum Do Link:** <https://www.scrumdo.com/projects/project/class-scheduler/iteration/92470>

**Github Link:** <https://github.com/sumanth2109/AdvancedSoftwareEngineering/tree/master/Class%20Scheduler/Increment-4>

**YouTube Link:** https://www.youtube.com/watch?v=5FayR1dwvGY