CS551 Advanced Software Engineering

Fourth Increment Report (PG - 7)

Title: Campus Network

Submitted By: Sumanth Gunda (Class ID: 21) (TL)

Goutham Donthu (Class ID: 14)

Kothuri UmaMaheshwarRao(Class ID:28)

Jonnalagadda Rajesh (Class ID: 23)

I. INTRODUCTION

To get an appointment from the Instructor students are using mail to interact with the respective faculty. There is no facility to the students to check their attendance. Instructors do not have the smart facility to assess the student's profile. Faculty are using mails to notify any deadlines to the students. We are designing a web based portal which provides all the above facilities in an efficient way.

II. GOAL OF THE PROJECT

Our project provides a web based application which includes major options like Appointment scheduler, Attendance tracker, Performance tracker, Course tracker. Instructors and Students will no longer interact with the mails. They are provided with an interface which makes the work simple, efficient and saves the time.

III. EXISTING SERVICES

1. **Service Name - Google Maps API**

   Description: - To track the student position before marking Attendance.

   URL: https://developers.google.com/maps/documentation/javascript/tutorial.

2. **Service Name - Google Chart API**

   Description: - To visualize the attendance and performance report of students.

   URL: https://google-developers.appspot.com/chart/interactive/docs/quick_start.

3. **Service Name - Google Drive API**

   Description: - To help create docs inside the web portal.

   URL: https://developers.google.com/drive/web/.

IV. DESIGN

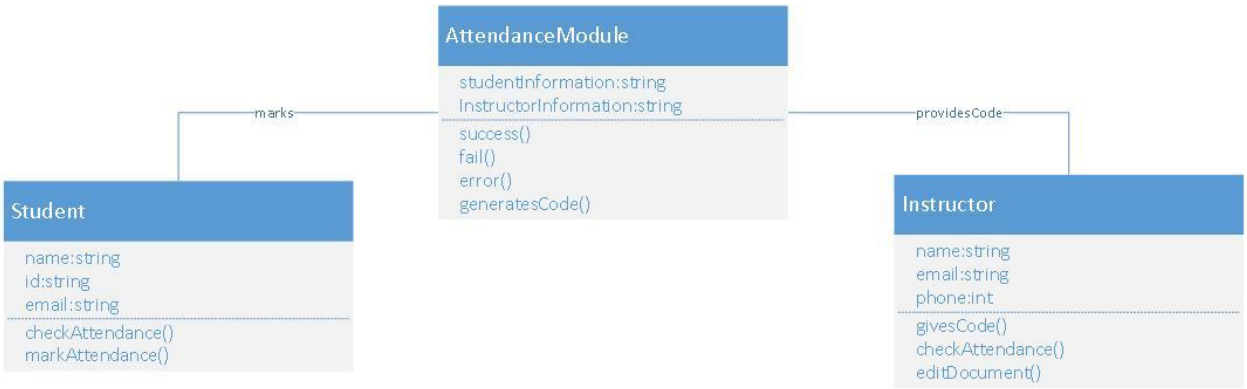
The portal has 3 logins Administrator, Instructor and Student.

The project is divided into 4 modules

1. Requests Module.
2. Course Tracking.
3. Performance Module.
4. Attendance Module.

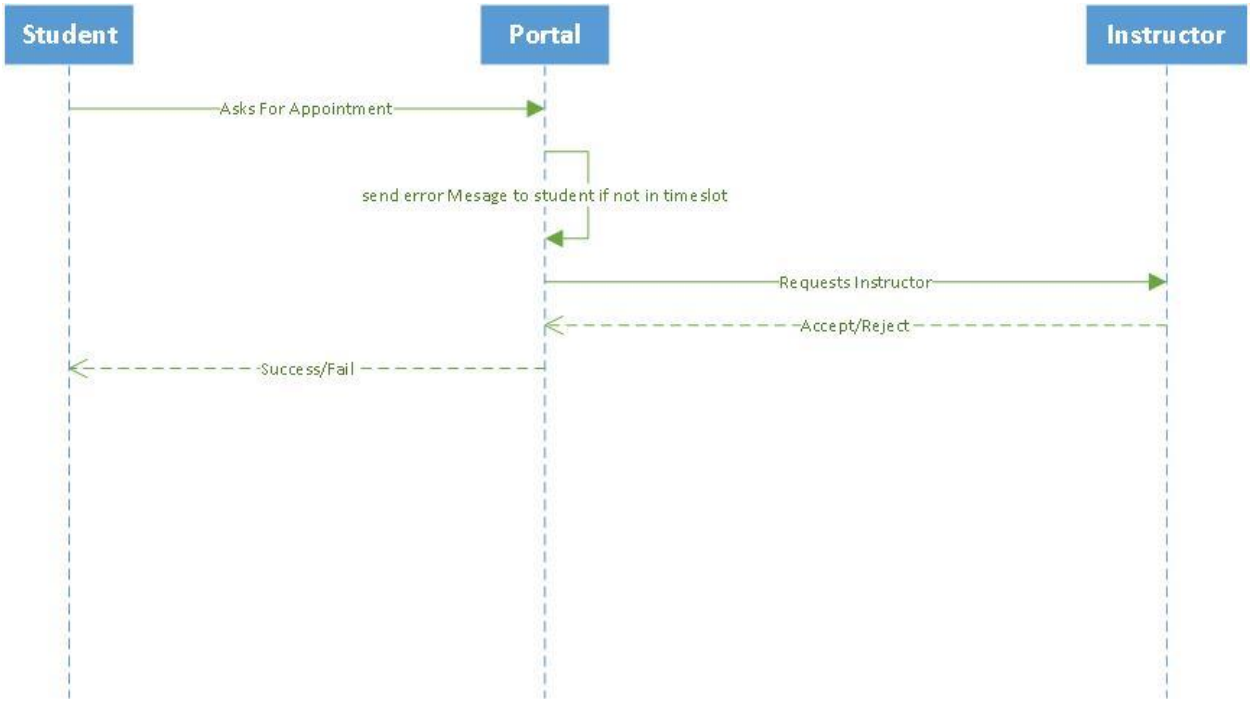
CLASS DIAGRAM:

Request Module

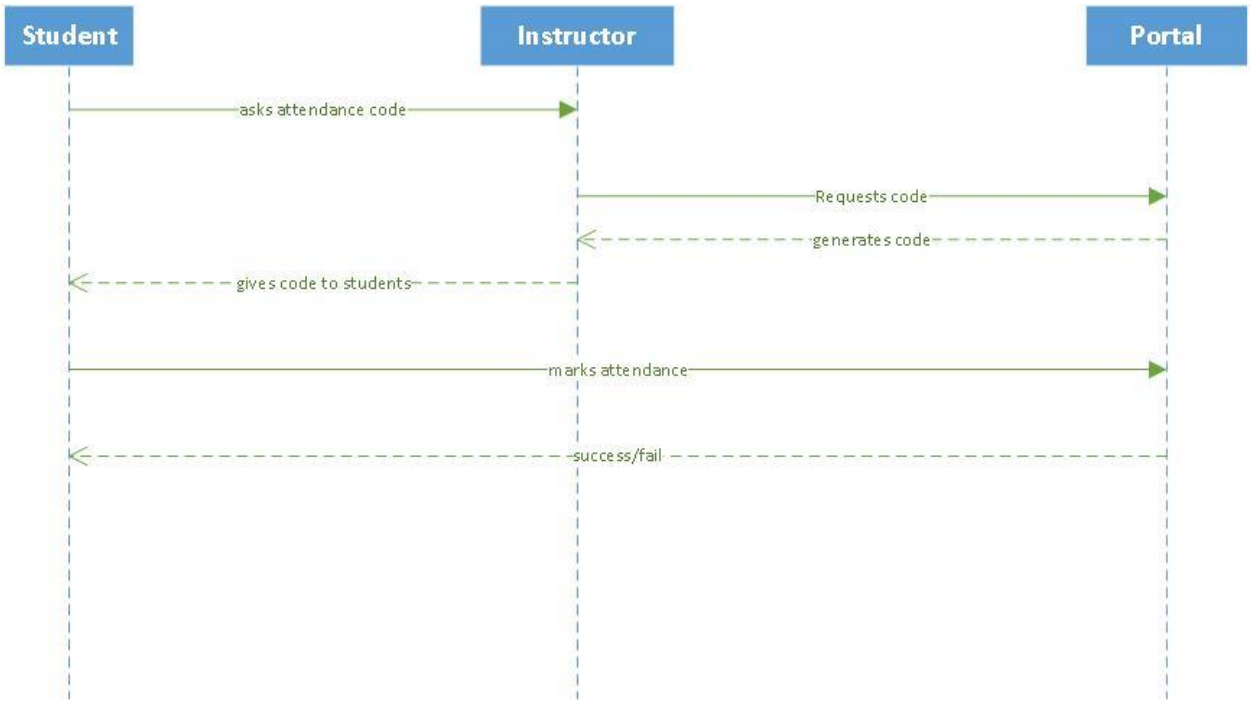


Attendance Module

SEQUENCE DIAGRAM:

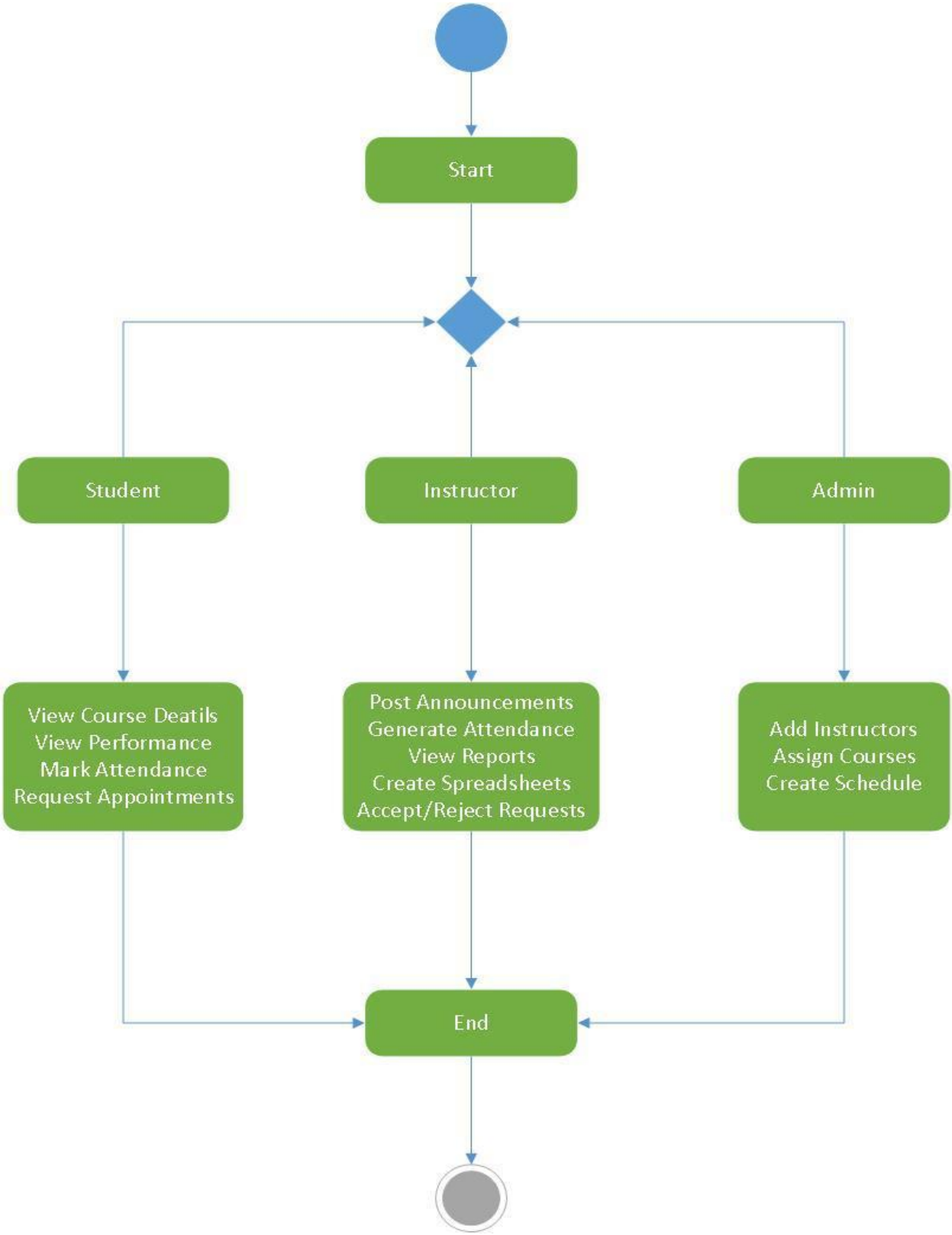


Request Module



Attendance Module

ACTIVITY DIAGRAM:



V. IMPLEMENTATION

The Phase 4 of the Project Implements

1. Home page extension for students
2. Home page extension for Instructor
3. Instructor page in Attendance module
4. Student page in Attendance module
5. Validation of all pages
6. Database design and Executing scripts

Screenshots:

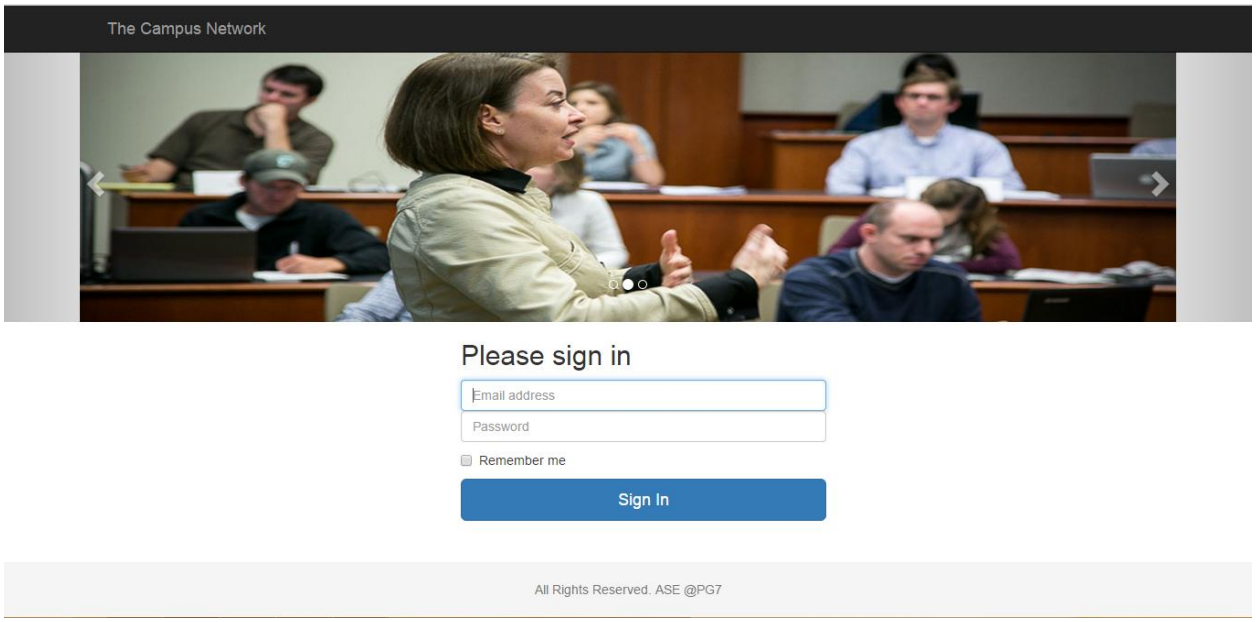


Fig 1: Login Page

This page will allow both students and Instructor to login to the portal.

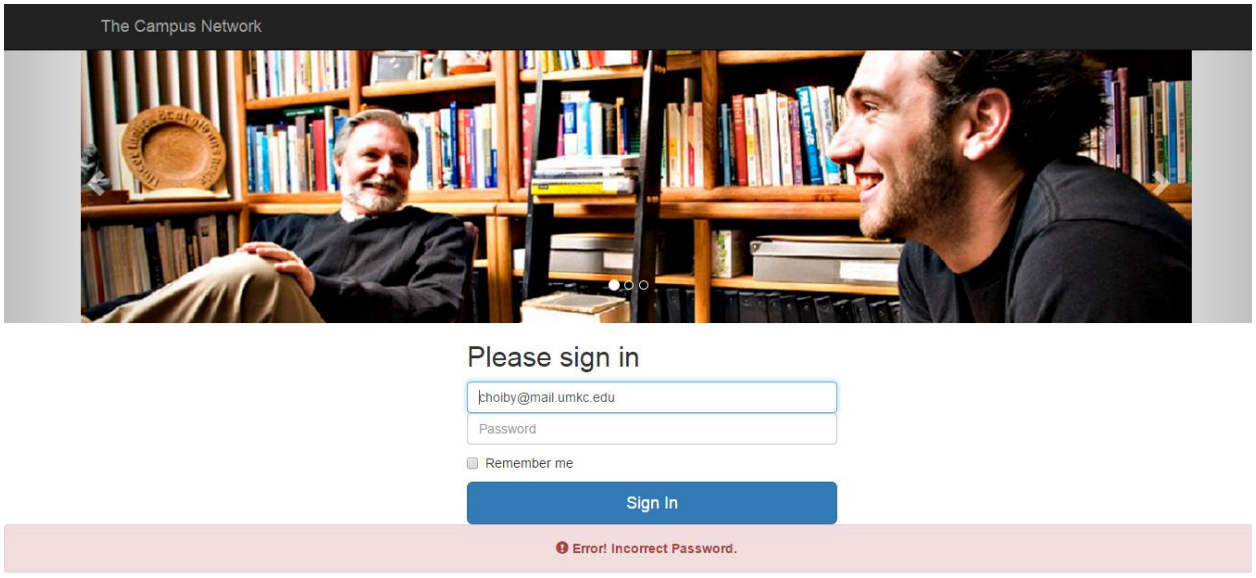


Fig 2: Login failed because of incorrect password

Here Instructor failed to enter correct password so login failed and error message is shown.

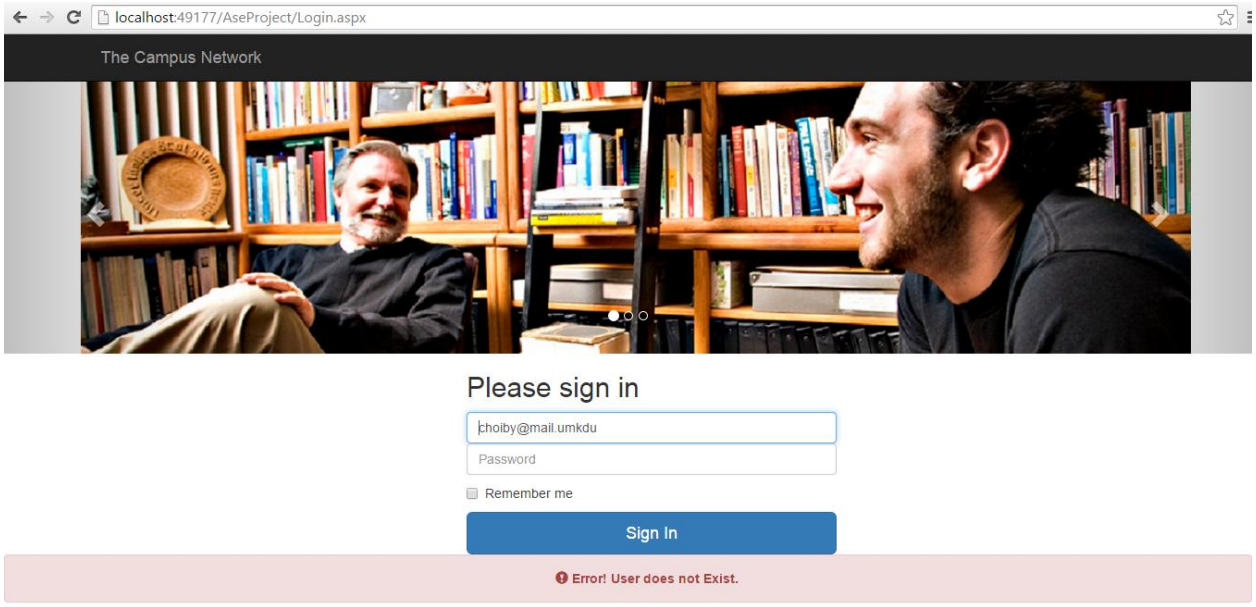


Fig 3: Login Failed because of wrong user name

Here Instructor failed to enter correct ID so login failed and error message is shown

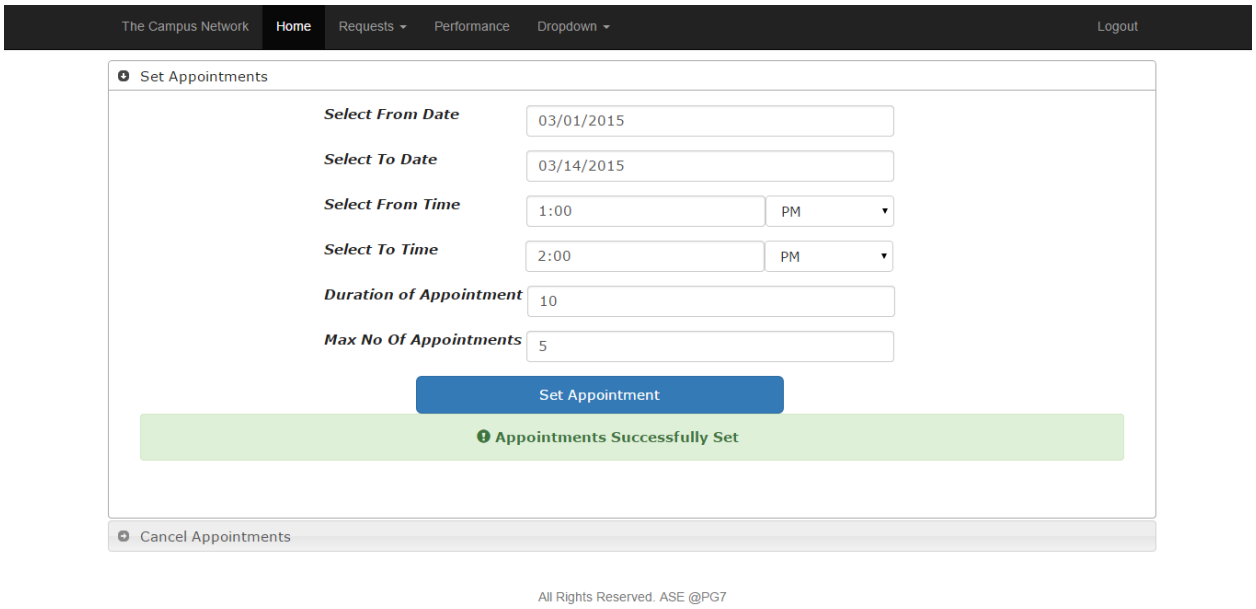


Fig 4: Instructor set's Appointment

Here Instructor can set the time for their appointments. He should select the start date, end date, start time, end time, duration and number of appointments. Based on all the requirements given **appointments are scheduled dynamically** in our project.

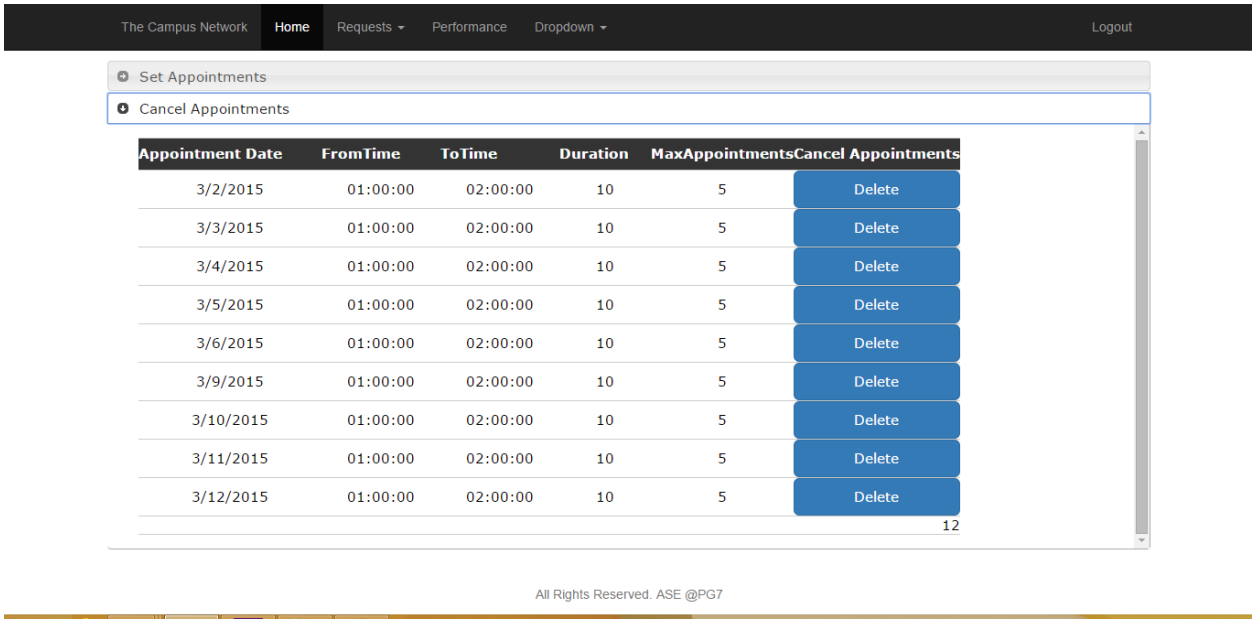


Fig 5: Instructor can delete Appointments for a day

Appointments are shown for all the days to the instructor. The beauty of this feature is if the Instructor is busy for any day **he can delete appointments for only that day** and the appointments on that day will be not be shown to students.

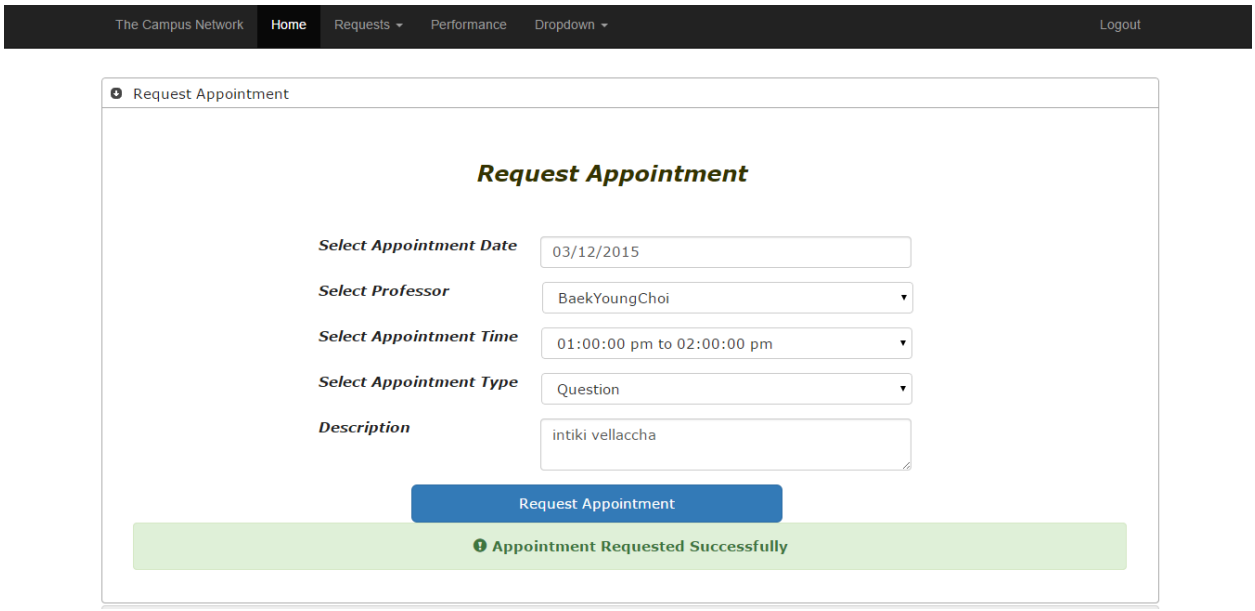


Fig 6: Student Requesting Appointment

Student can request appointment from any of his Instructor. Only the Instructor of the courses registered by the student is displayed to him. This request will be **pushed to the respective Instructor**.

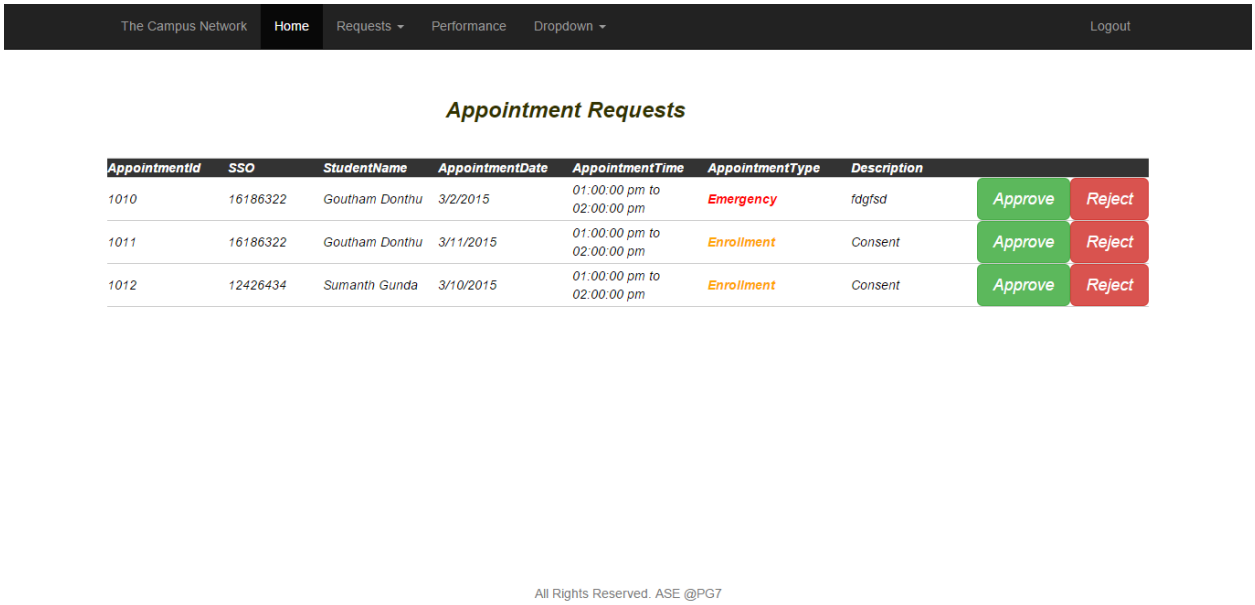


Fig 7: Instructor's view of Appointments

This is the page showing all the appointment requests came to the Instructor by students. Instructor **can approve or reject** the appointment and the result will be shown in student's page.

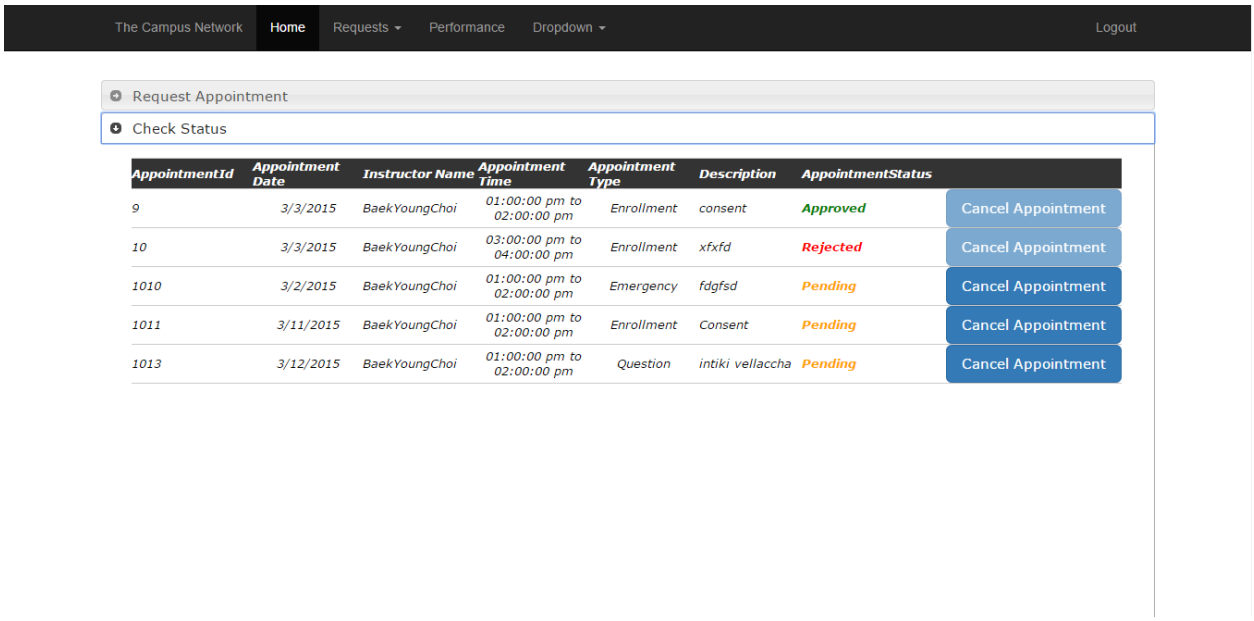


Fig 8: Student's view of his Appointment Status

Student can view status all his Appointments. You can see color variation in cancel appointment button. **Dark blue** cancel appointment button indicates that Instructor did not see his request so that student himself can cancel his request. **Light blue** indicates that Instructor already took decision about appointment.

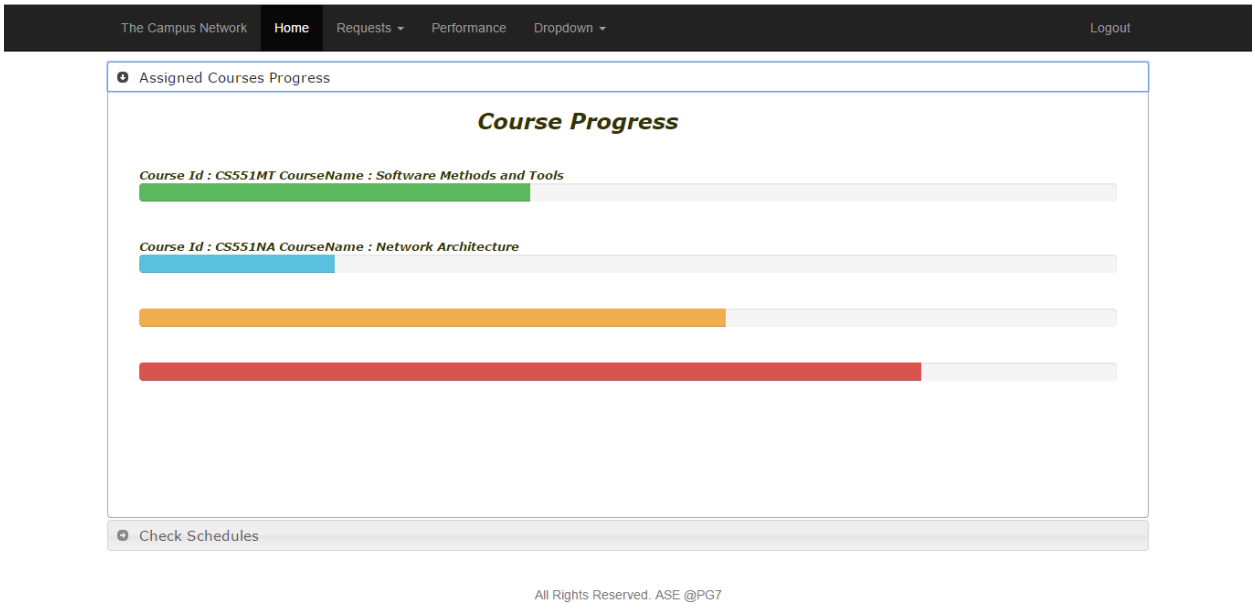


Fig 9: Course Completion Progress

Bar indicates the progress of course completion dynamically

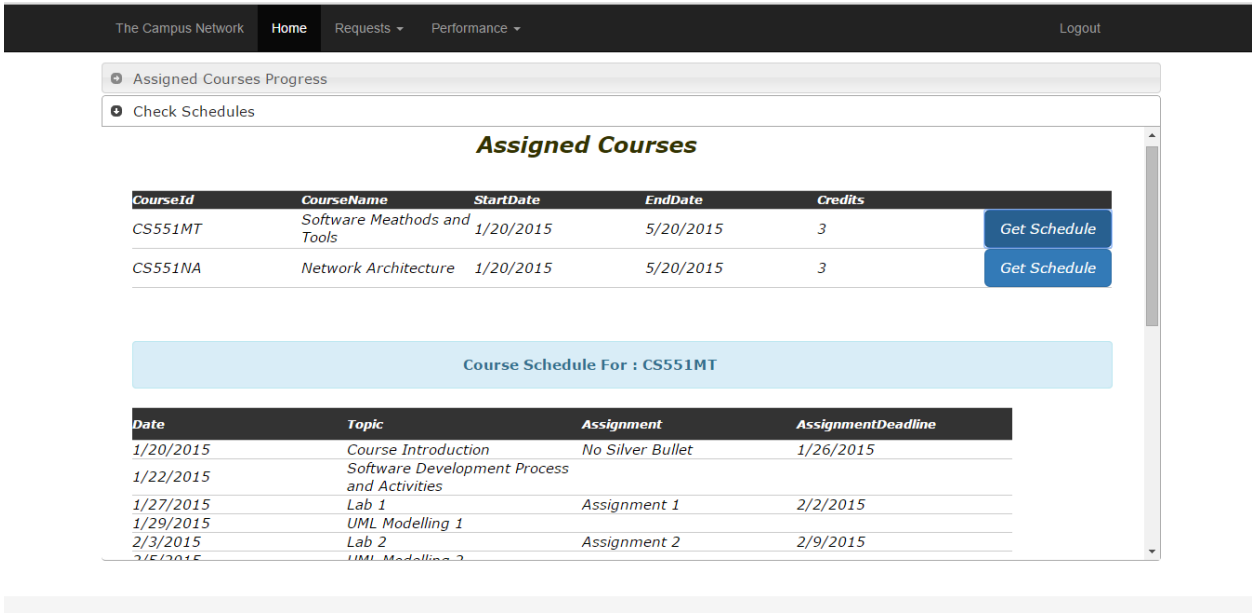


Fig 10: Assigned courses for Instructor

Here Instructor can see what all **courses assigned for him** and details of those courses. When he choose getschedule button the schedule is displayed below as shown in this figure.

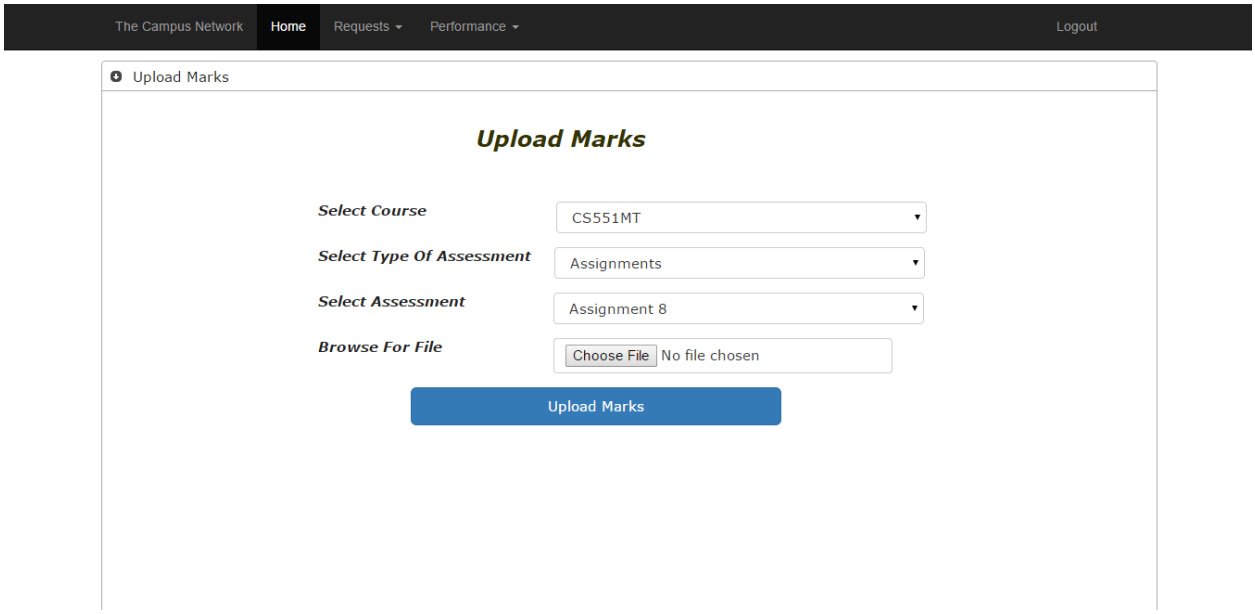


Fig 11: Uploading marks

Here Instructor can upload student marks. Based on the type of assessment selected **assessment options will be changed**. For example in type of assessment Assignments are

selected so in select assessment drop down list only assignments are shown. Instructor should add respective excel sheet for uploading marks.

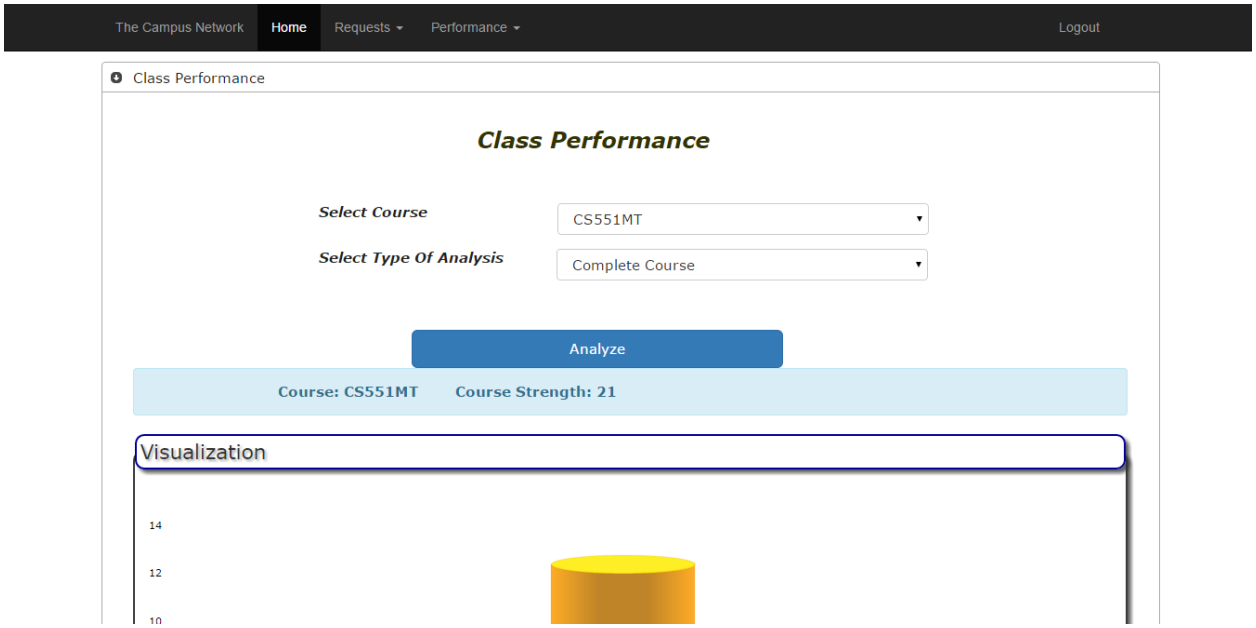


Fig 12: Seeing class performance

Here Instructor can see the performance of class

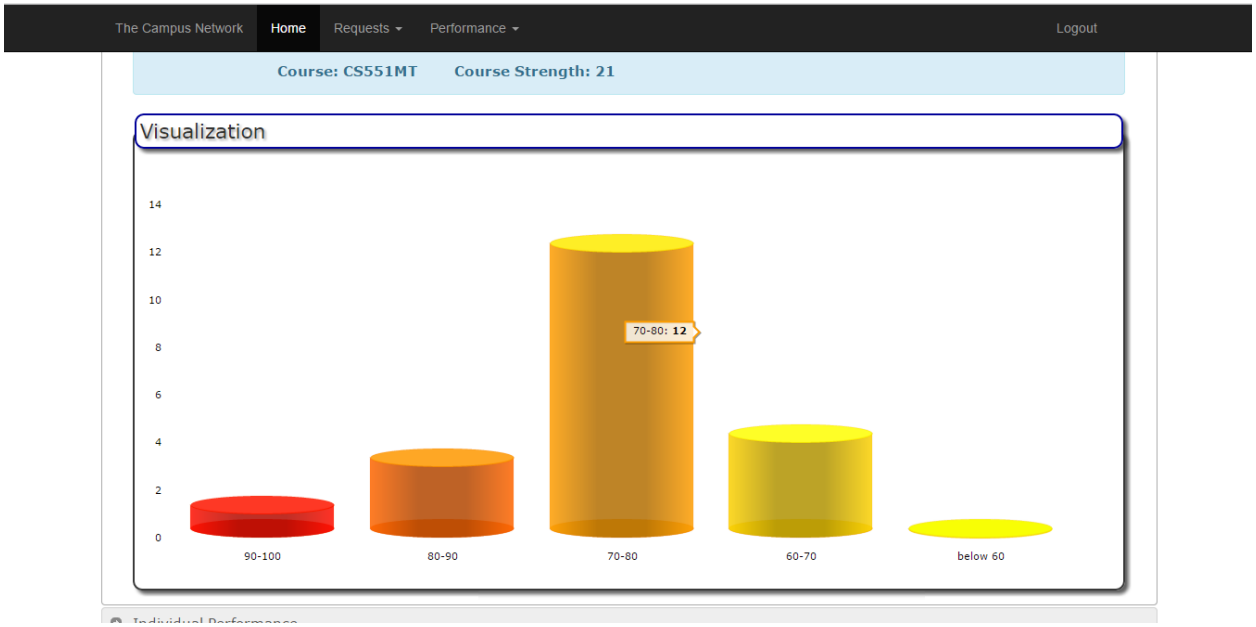


Fig 13: class performance

Class performance for complete course is graphically represented

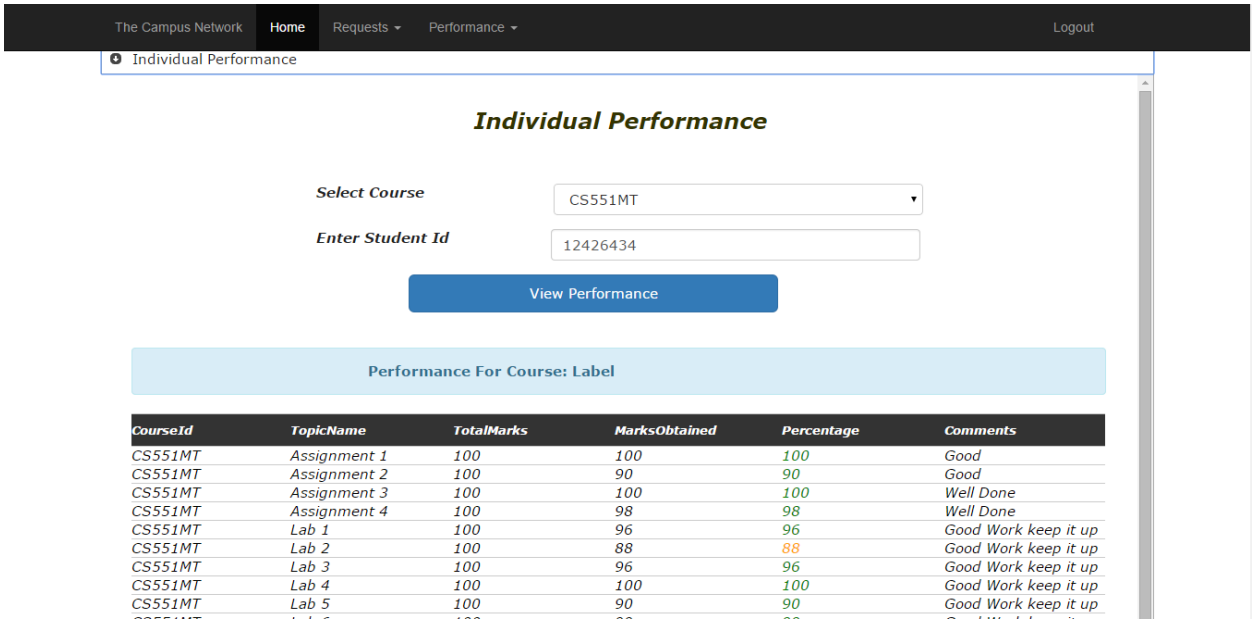


Fig 14: check individual performance

Instructor can see individual student performance by selecting respective student ID

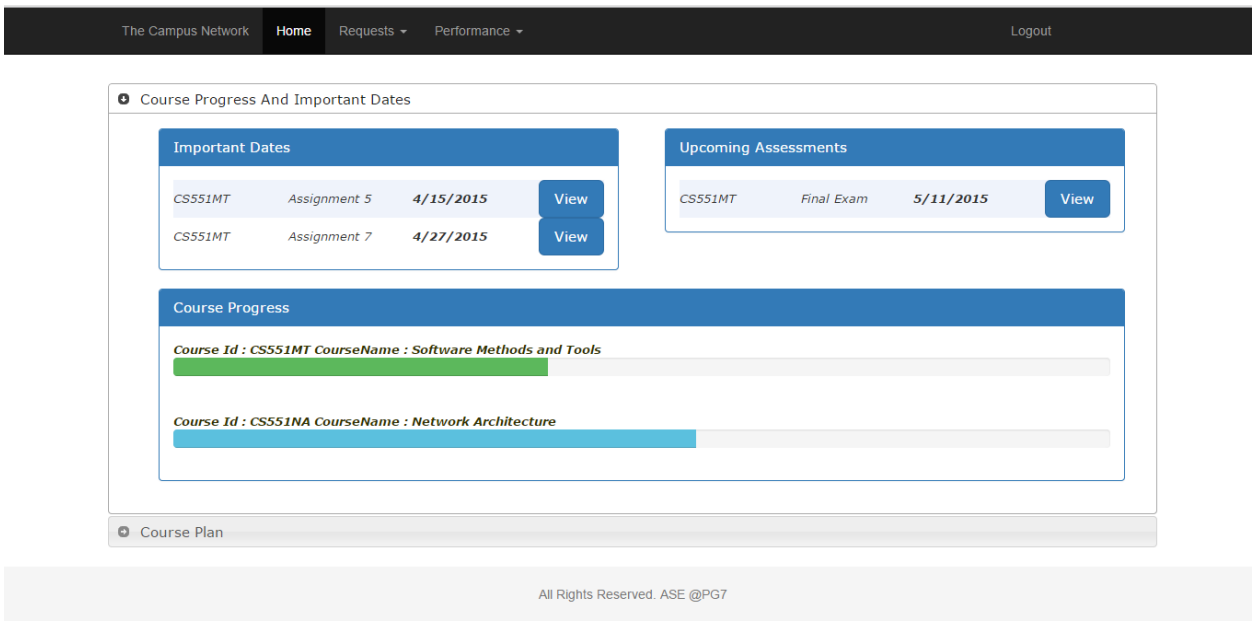


Fig 15: Student's view of course progress

This is the page showing course progress and all important assessments for students. The **works for the coming two weeks** will be displayed to the students in important date's panel. In upcoming assessments **next exam** will be displayed.

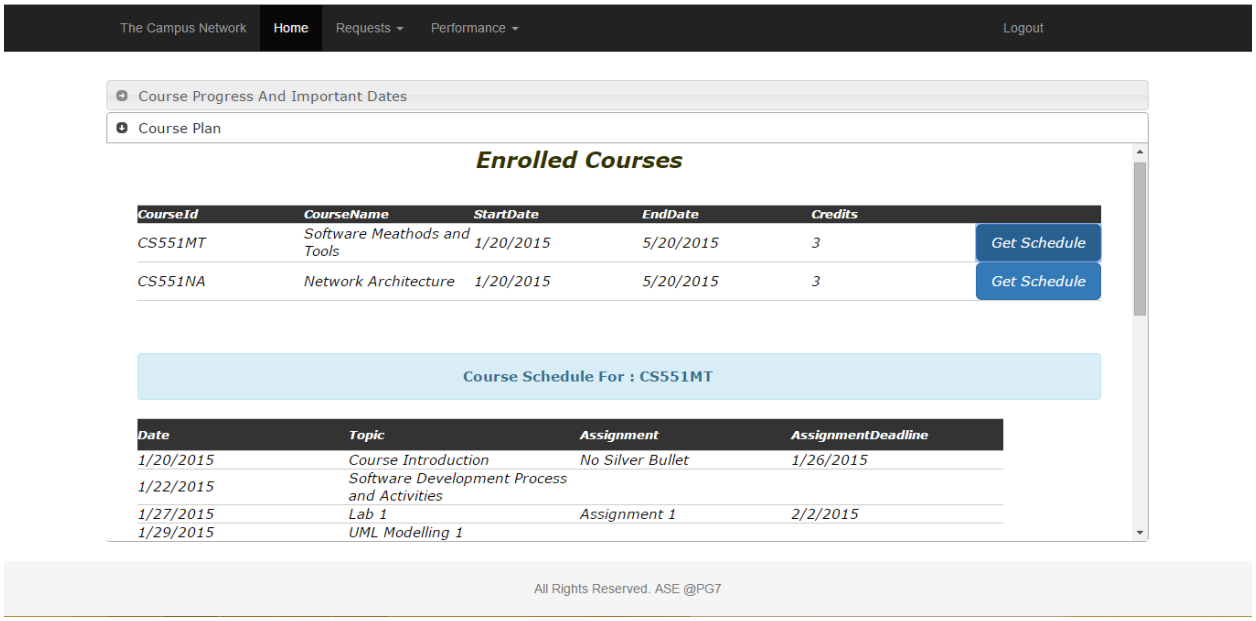


Fig 16: Student's view of his enrolled courses

Student viewing what courses he enrolled and can get schedule for his course

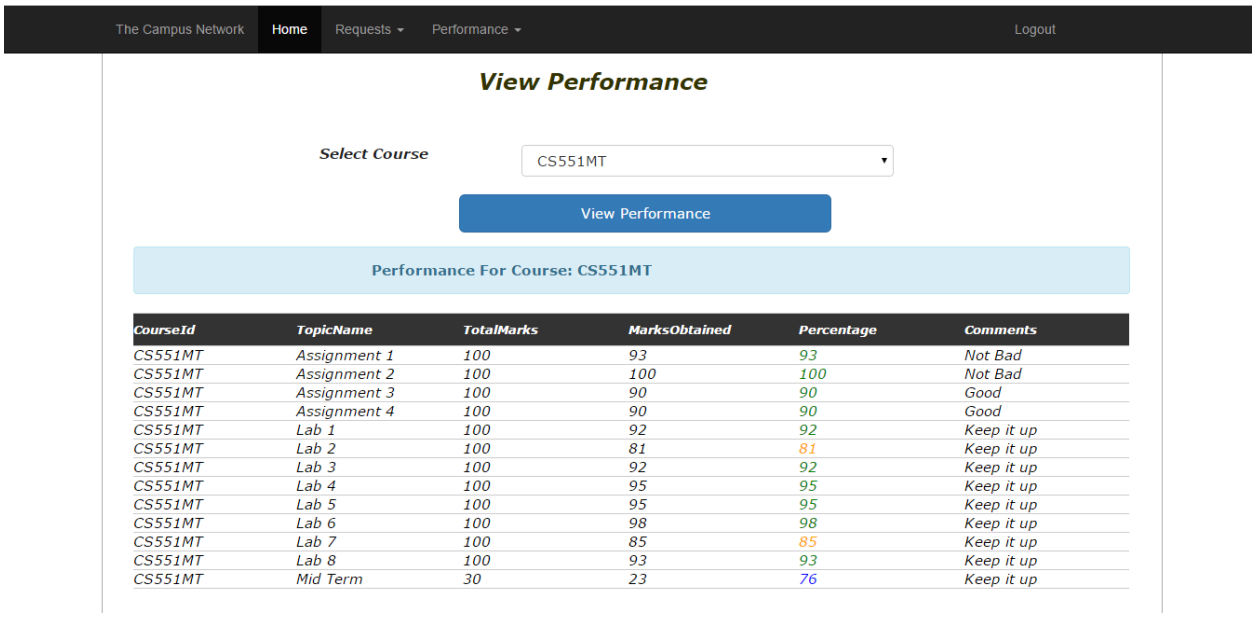


Fig 17: Student can view his performance

Student can check all his scores and respective comments by Instructor

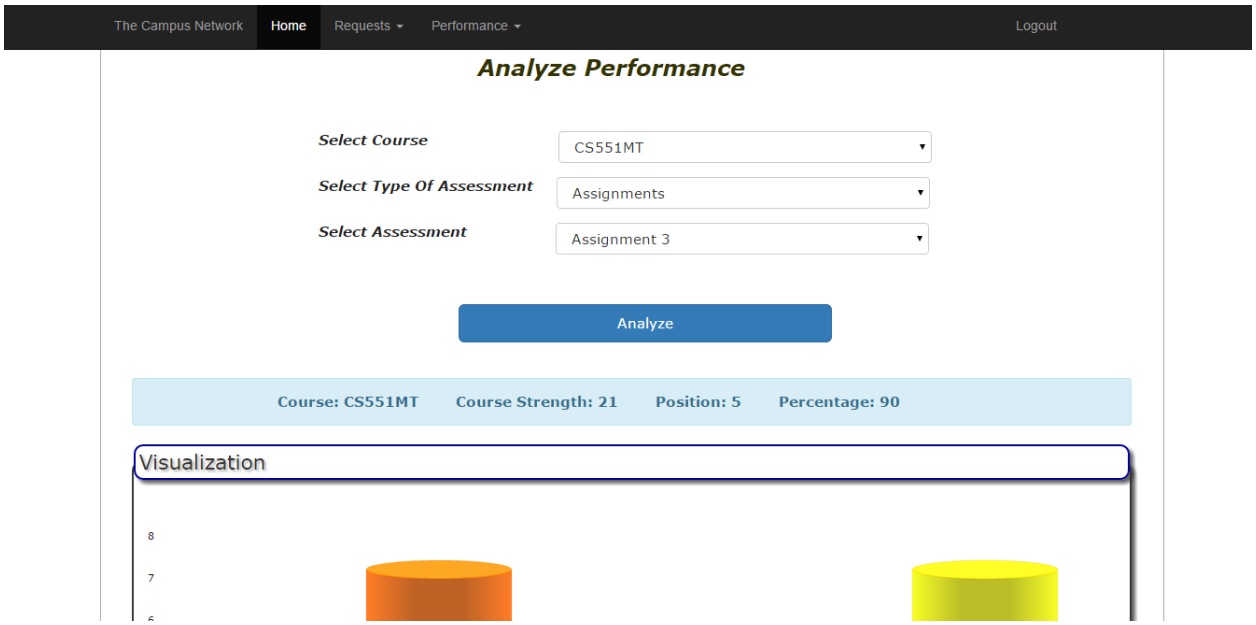


Fig 18: Student can analyze his performance

Student can analyze his performance in class and he can **check his position** in particular class

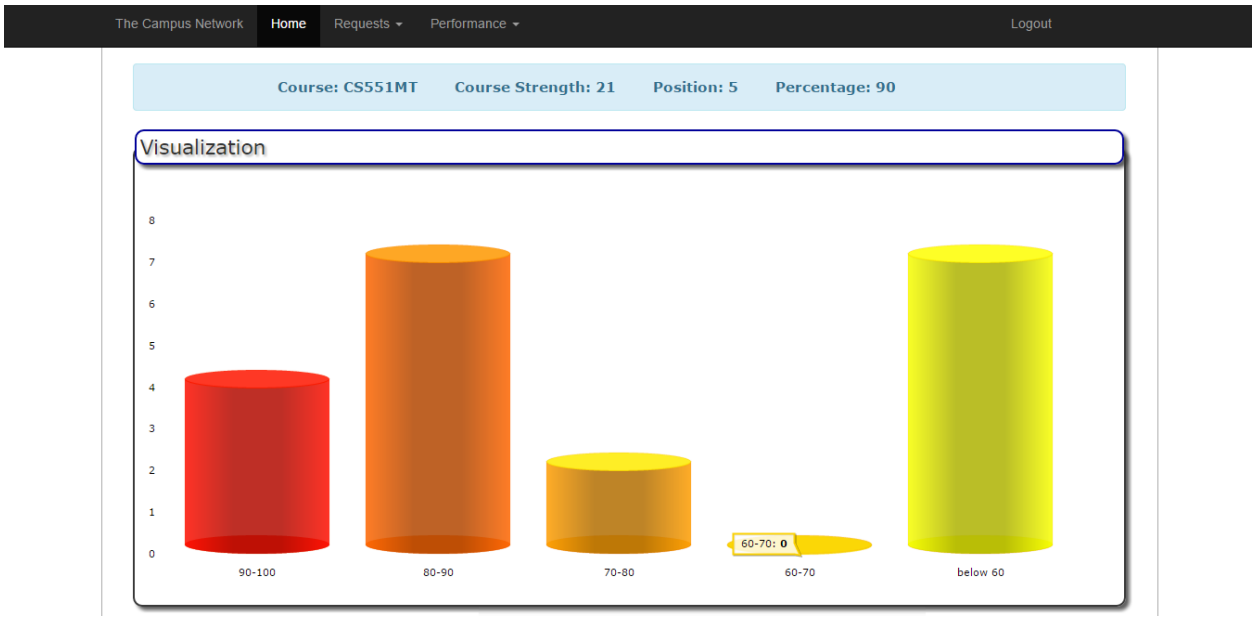


Fig 19: Graphical representation of student's performance

This is the graphical representation of all students' performance. If you take your **mouse pointer on to the bar** number of students scored in that level will be shown to you.

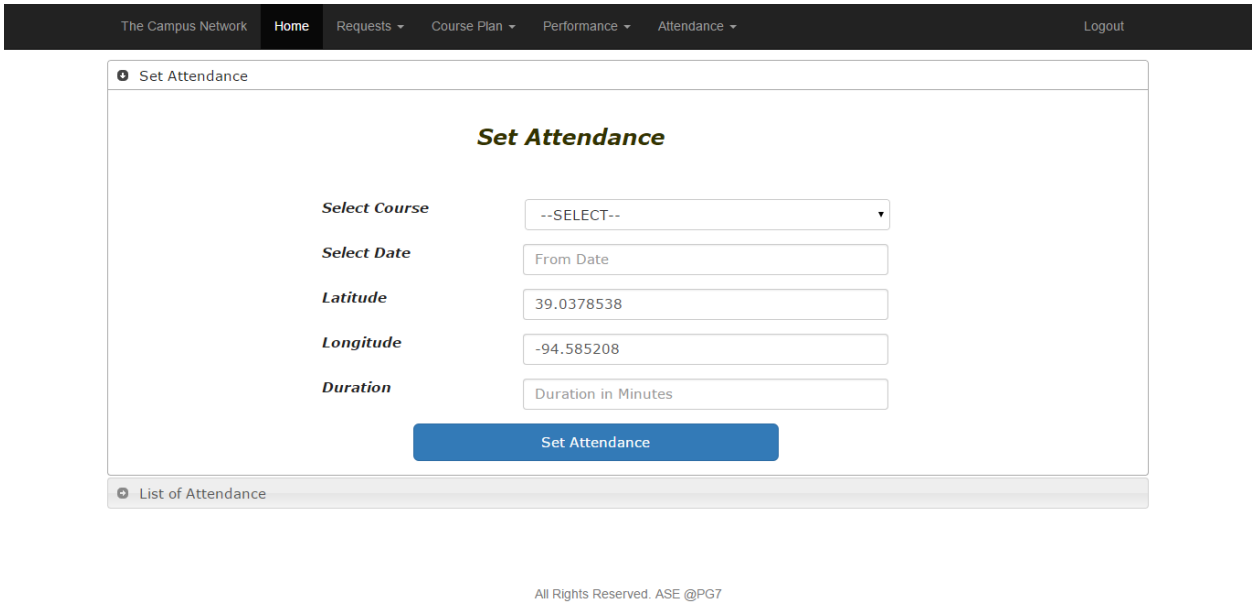


Fig 20: Set Attendance

This page will allow Instructor to set the attendance for that course. It takes the **latitude and longitude** positions such that students who are outside the class cannot mark their attendance.

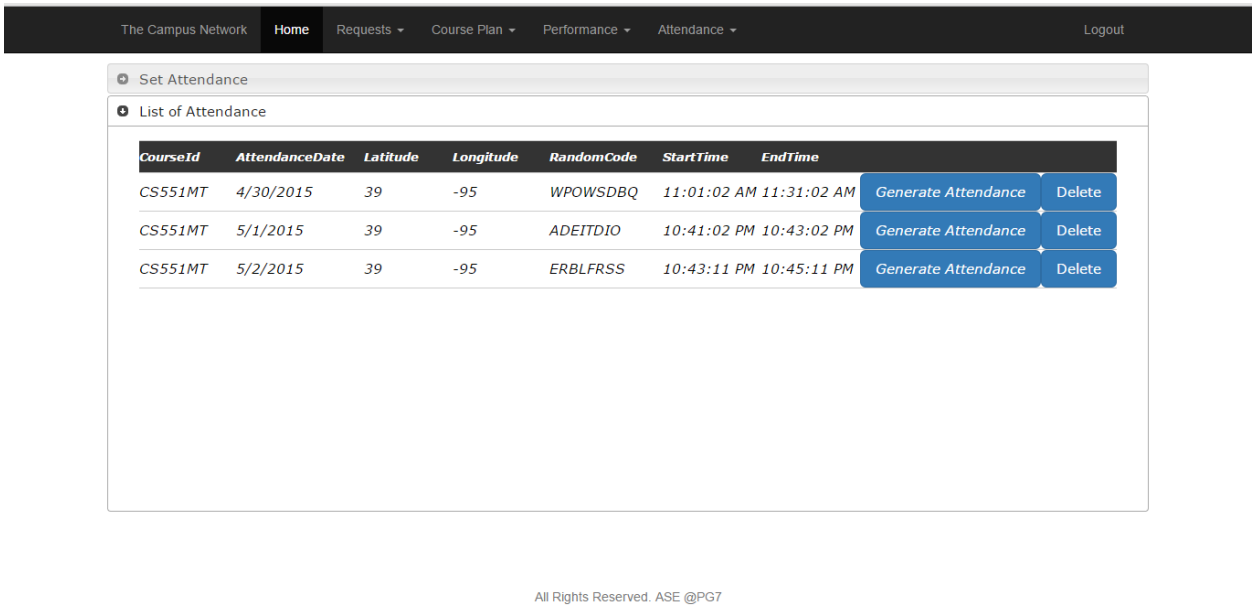


Fig 21: List of Attendance

Generate Attendance option helps instructor to **mark absent** for those students who failed to mark the attendance on a particular day. He can also delete the attendance for that particular day.

Fig 22: Analysis of whole class attendance

Here Instructor can analyze attendance for whole class for a particular day. If we take the **cursor on to the bar** it shows the number of students

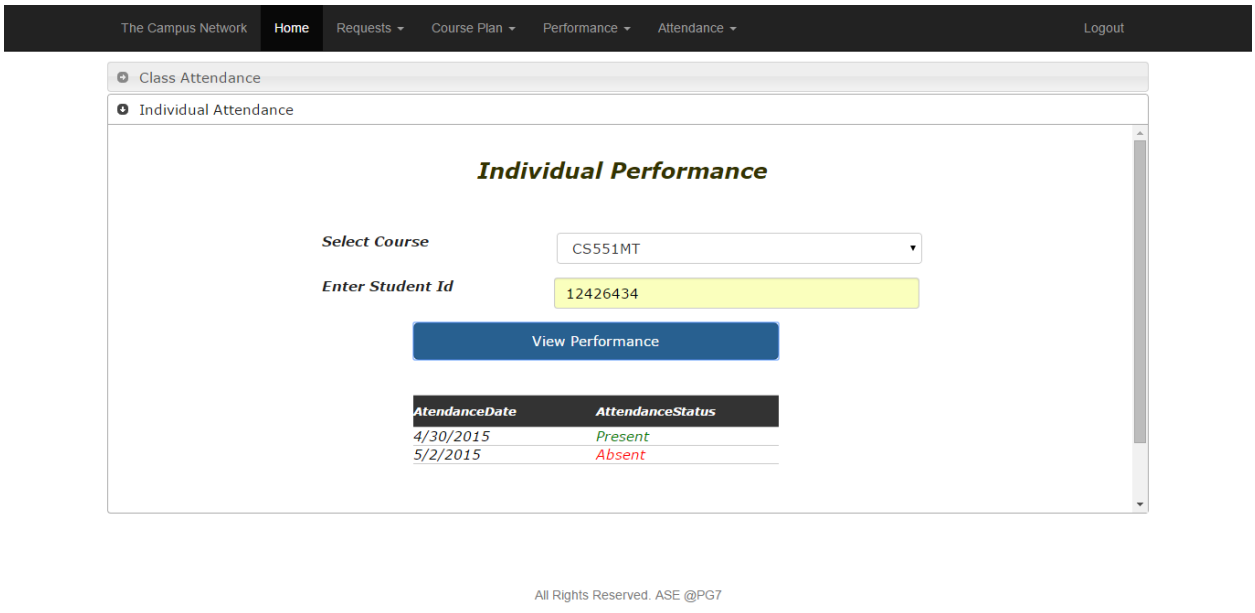


Fig 23: Individual Attendance

Here Instructor can track the attendance for particular student by providing his ID. By this way he can check students attendance individually.

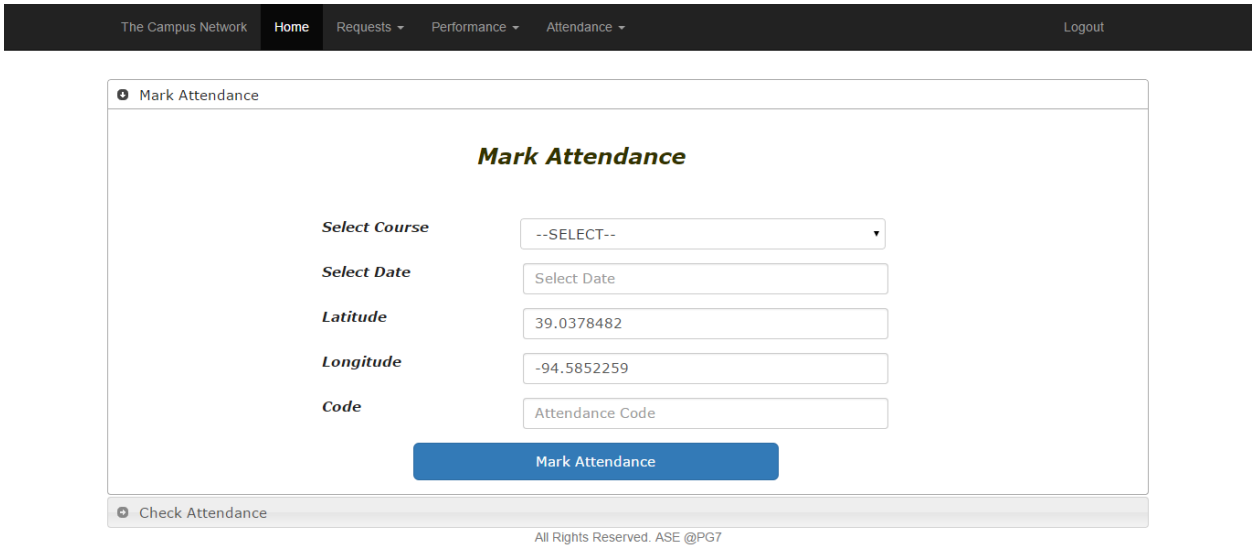


Fig 24: Mark Attendance

Student can mark his attendance by only providing the random **password** given by the Instructor and if they are in that particular **location** only

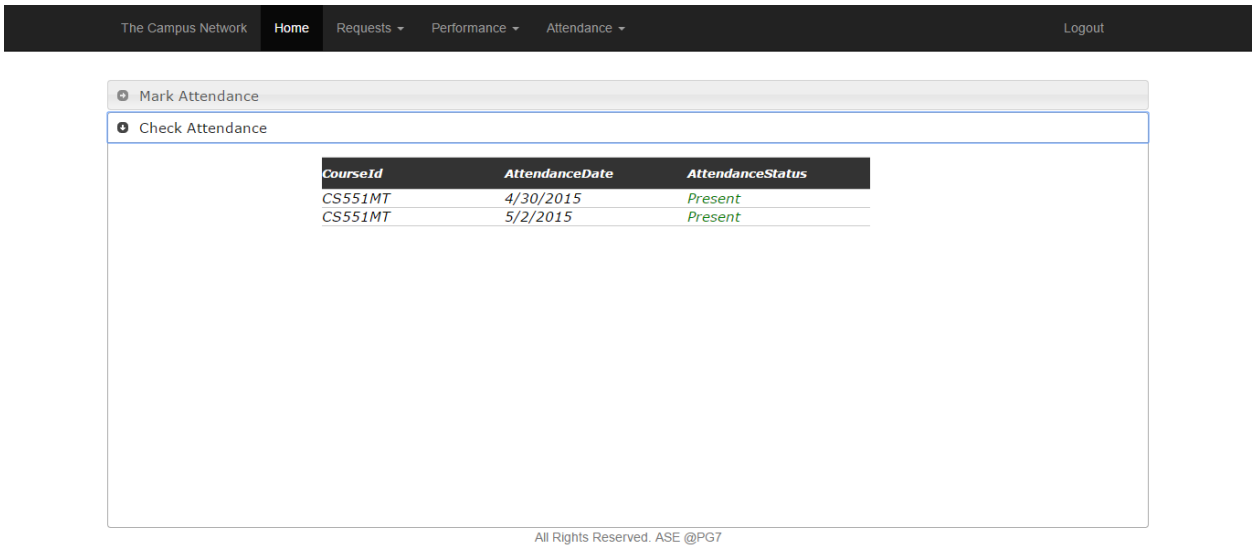


Fig 25: Student can check his attendance

Student can check his attendance overall the course. This helps him to know how many classes he missed so that he can take care of his attendance.

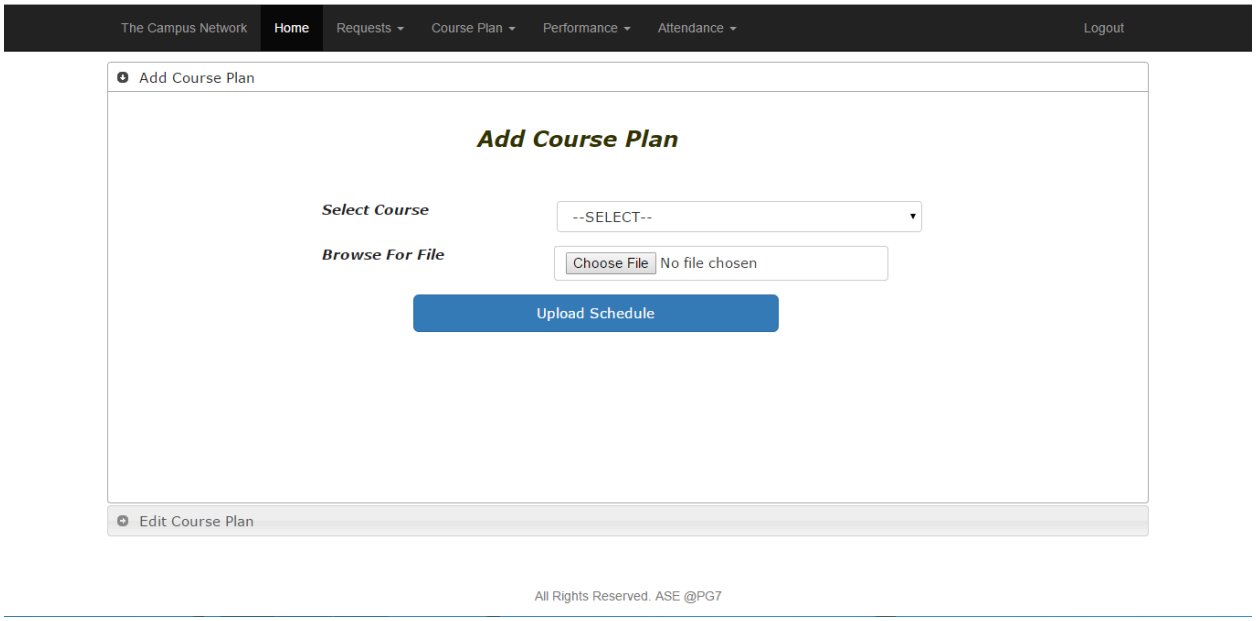


Fig 26: Course plan

Instructor can **add their course plan** by uploading plan through excel sheet. It is easy for them to add their course plan through this instead of maintaining a separate spread sheet for it.

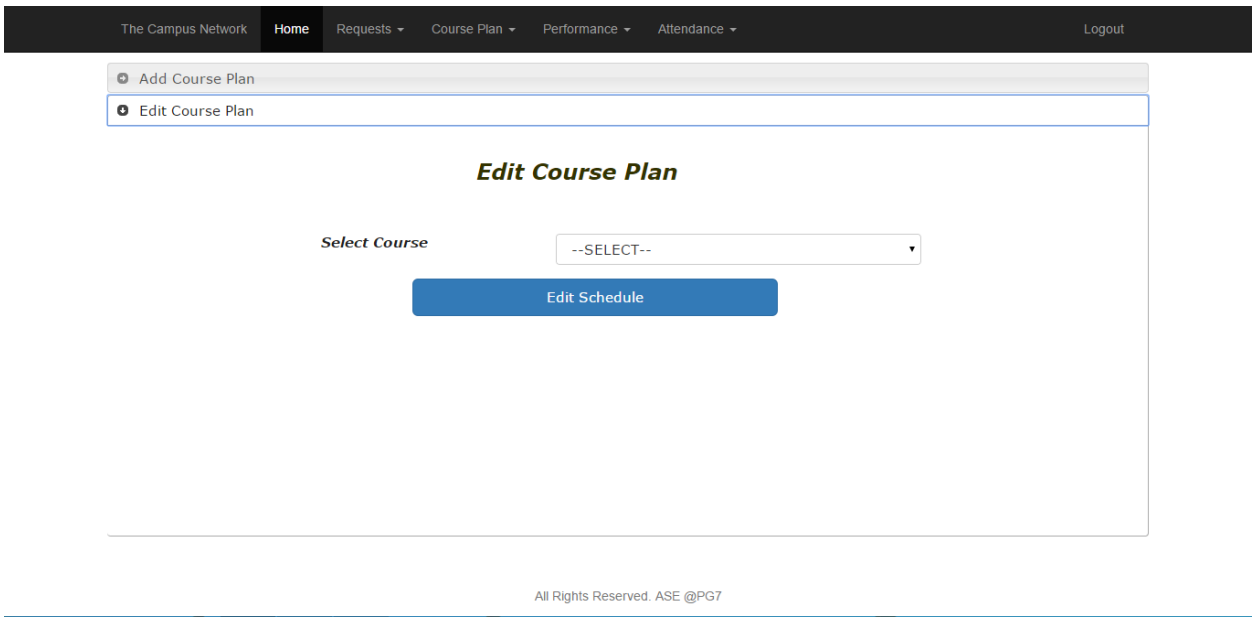


Fig 27: Edit course plan

Instructor can edit their course plan by choosing edit schedule option. This helps them to **change the schedule at any time** based on the circumstances.

VI. TESTING

To make sure the quality of the software is good testing is a critical component of the development. Software testing is done to ensure quality and it is also the review of the design and specification.

NUnit Testing:

This tool is used to unit test the application that is developed in C# ASP.Net. It is a Unit Testing Framework when the code that was written in C# can be tested.

The test cases that were designed to test the Campus Network Application are:

- **MasterPage()**

  This function test for the successful loading of the master page when a content page has been inherited from the master page.

- **HomePage()**

  This function tests for successful loading of the dashboard of both the student and instructor after successful loading.

  This function also tests whether the session variables are stored or not.

- **LoginPage()**

  This function tests for successful loading of the login page and also tests for all the validation controls on the page. This function tests whether the appropriate error messages are being displayed.

- **Validations()**

  This function test whether the validations are working properly before the page is sent back to the server.

- **ManageAppointment()**

  This function tests whether approve or reject works properly and the web page redirected to expected page or not

- **ApplyAppointment()**

  This function tests whether after applying appointments Instructor gets the request or not

- **AppoinmentRequest()**

  This function tests whether appointment request came to respective Instructor or not

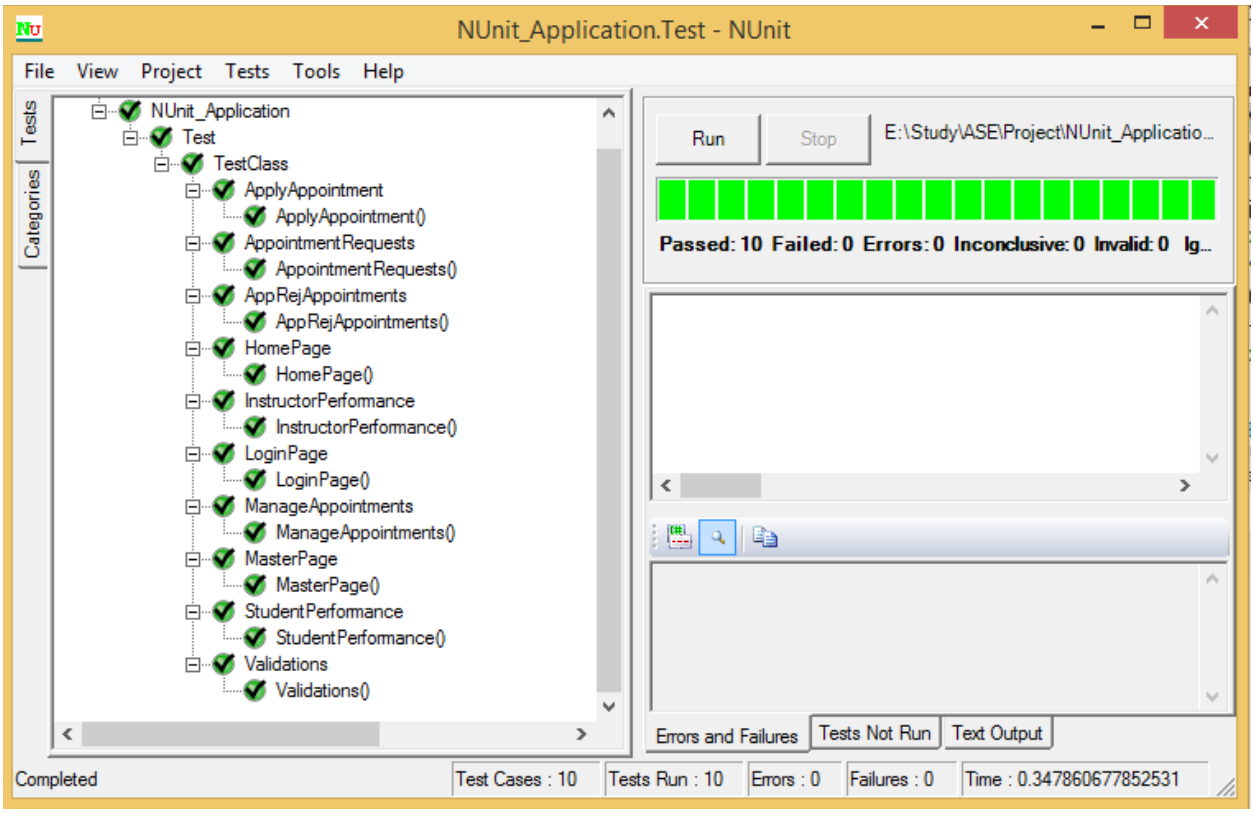
- **InstructorPerformance()**

  This function tests for all the validations in performance module on the instructor login.

- **AppoinmentRequest()**

  This function tests for all the validations in th performance module on the student module.

Screen shot for NUnit Testing:

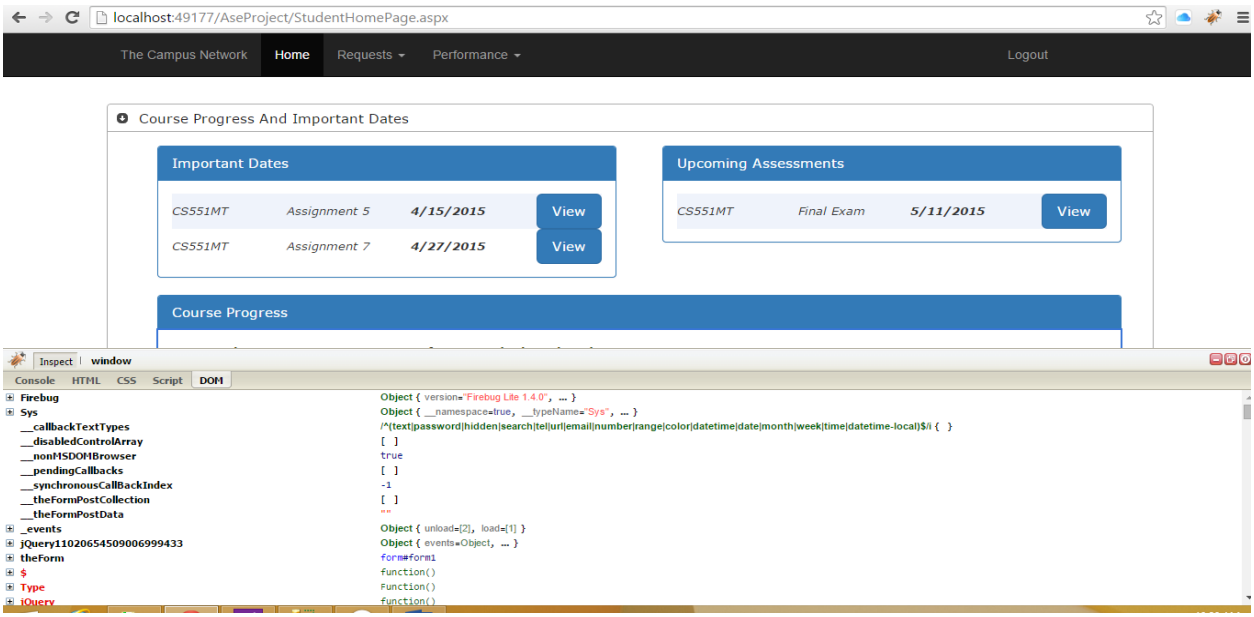


Implementation of Test Cases

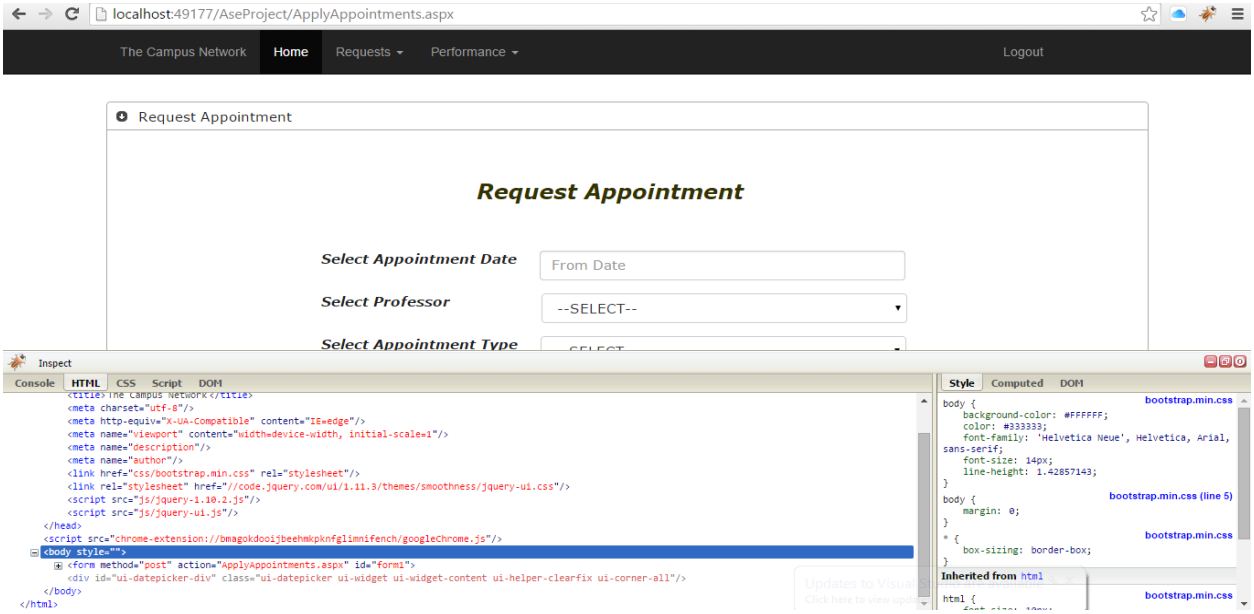
VII. PERFORMANCE TESTING

- **Firebug Testing:**
  It is a web browser extension that will facilitate the live debugging and also monitoring of any websites

We have tested student home page and apply appointments page using firebug for this Increment.



Firebug Testing for Student Home Page

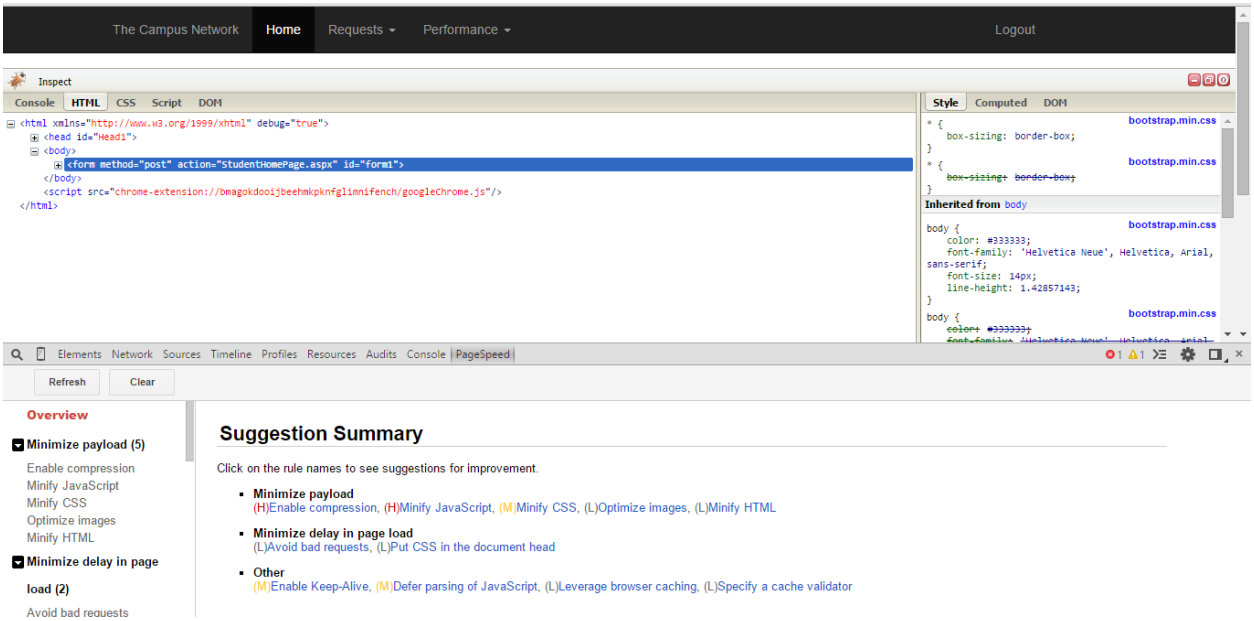


Firebug Testing for Apply Appointments Page

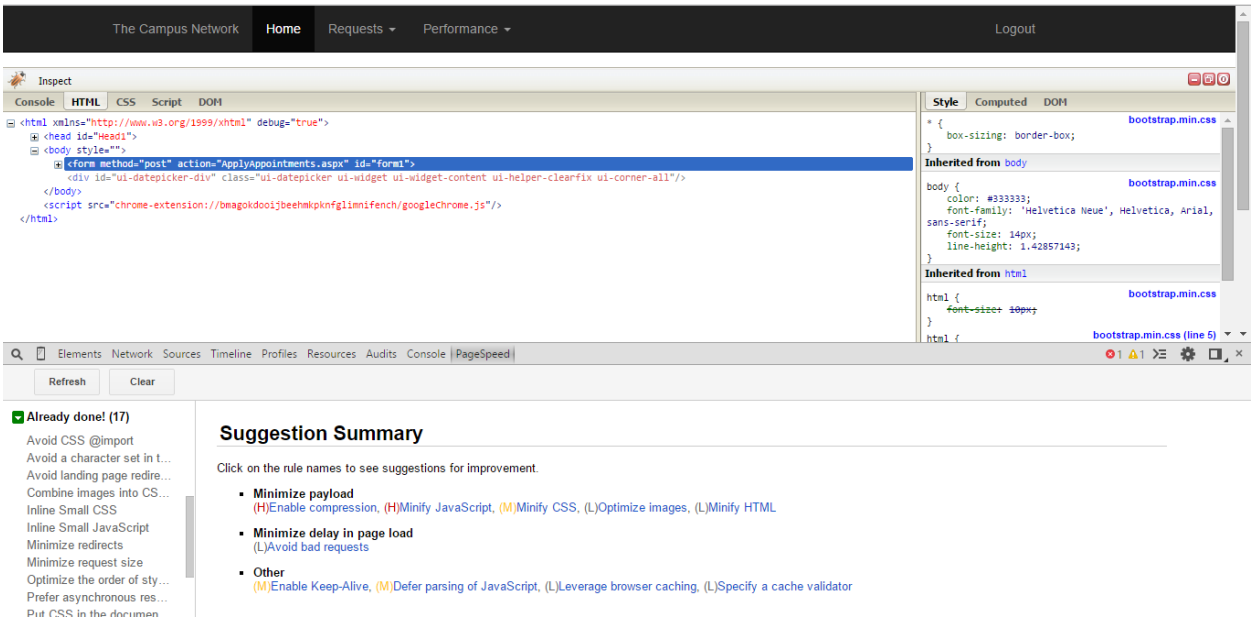
- **Page Speed Insights:**

  It is now available as a google chrome extension to test the load time of the page and also it gives the suggestions for the page so that it can be made faster.

  We have tested Student Home Page and Apply Appointments page using Page Speed Insights to analyze the speed of the pages and get some suggestions



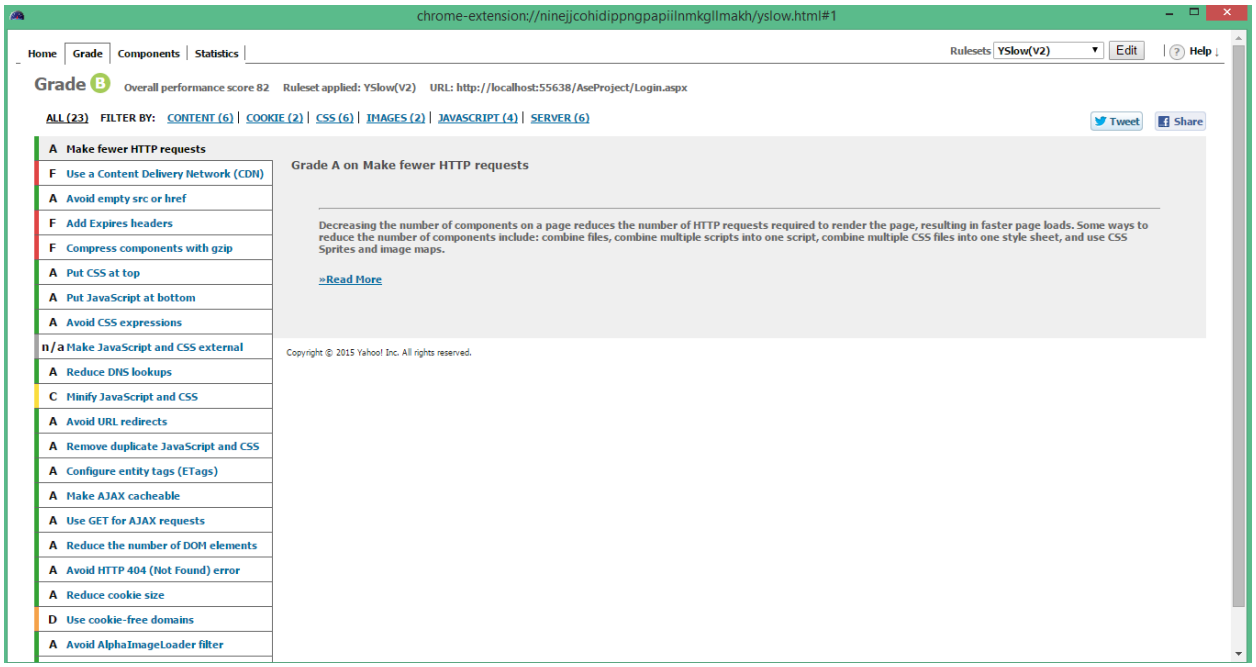
Page Speed Insights for Student Home Page



Page Speed Insights for Apply Appointments page

YSlow analyzer for Web Page

- **YSlow Analyzer:**
  - It Grades web page based on one of three predefined ruleset or a user-defined ruleset
  - It offers suggestions for improving the page's performance
  - Summarizes the page's components
  - Displays statistics about the page



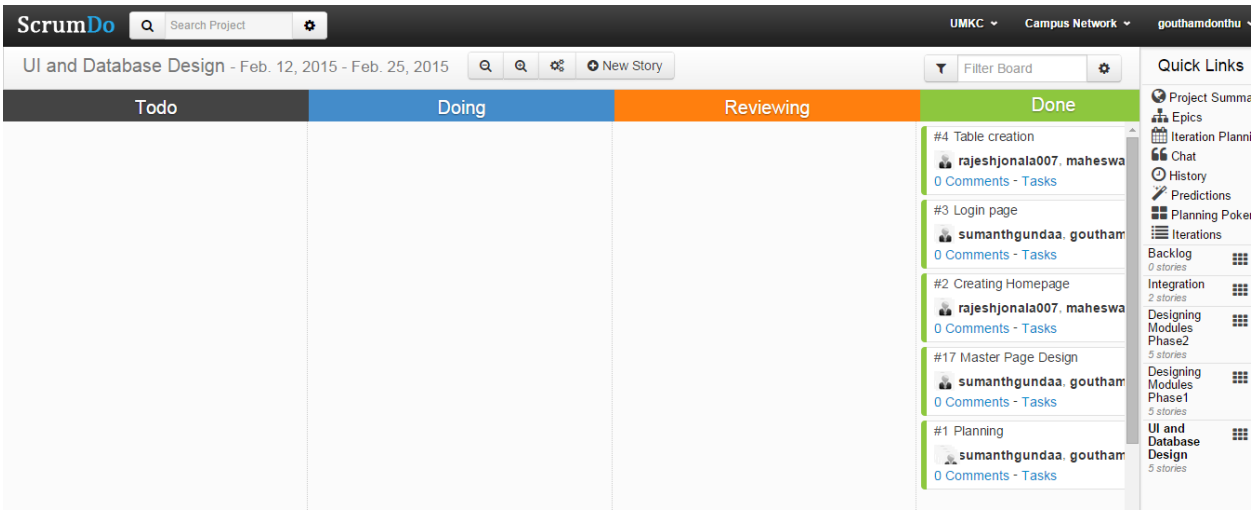
We analyzed our web page through it. YSlow **graded B** for our web page and overall performance score is **82**.

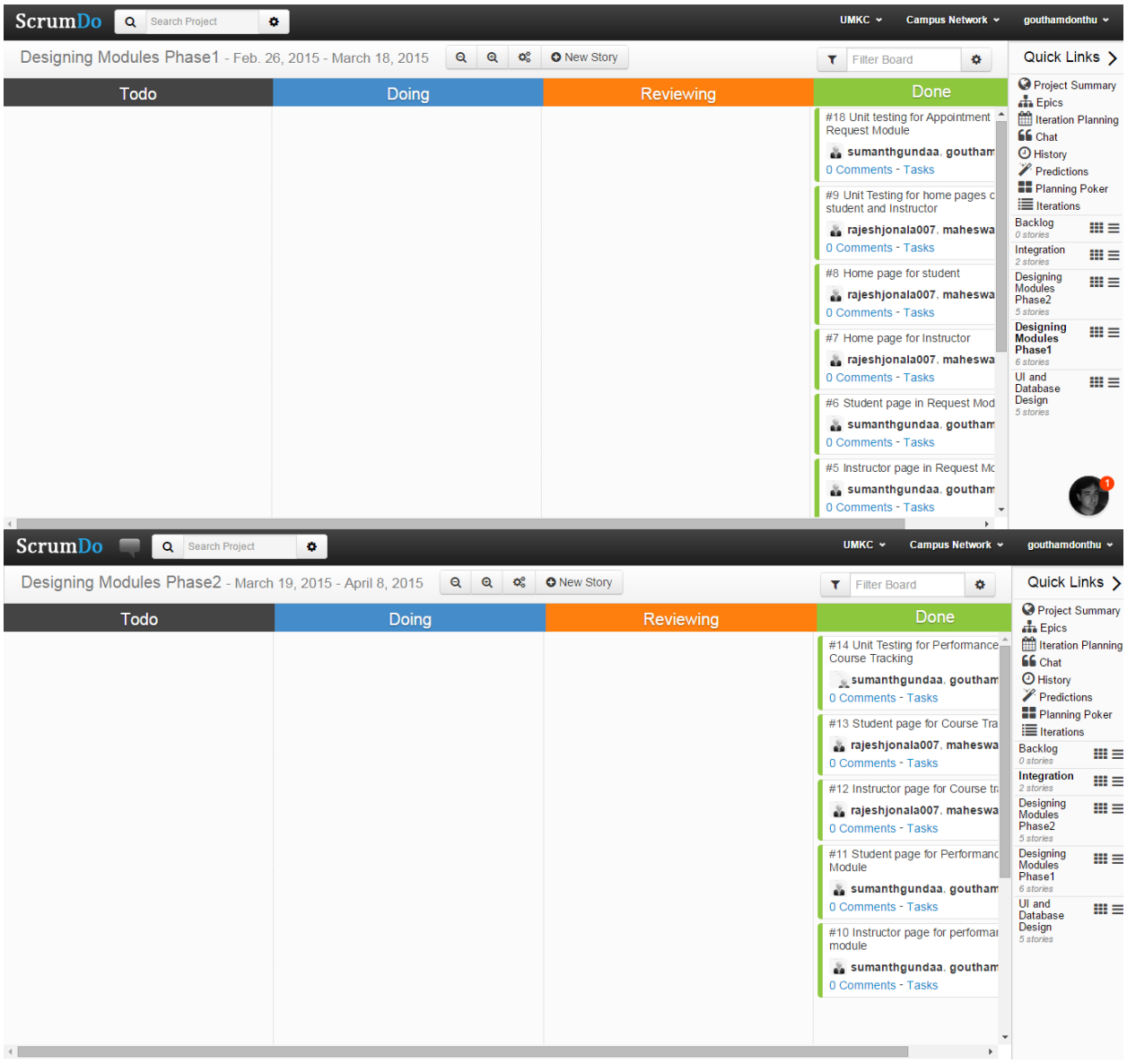
VIII. DEPLOYMENT

Project Scrum Do Link:

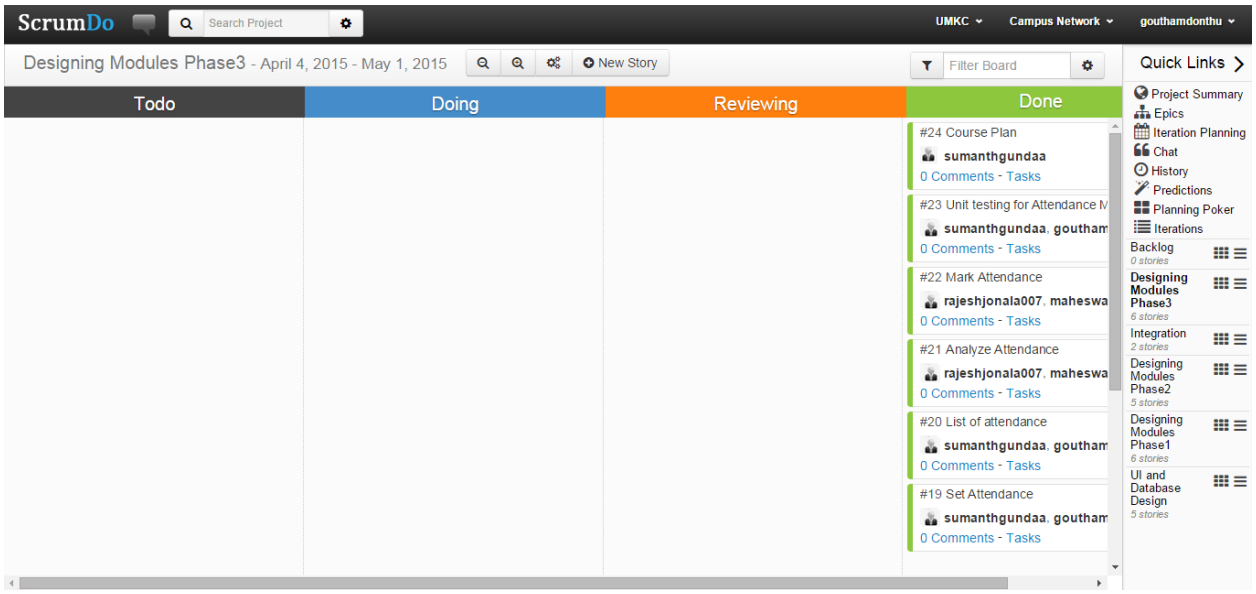
http://www.scrumdo.com/projects/project/campus-network/iteration/121658/board

Screenshots:




ScrumDo
Search Project
UMKC
Campus Network
gouthamdonthu
Designing Modules Phase1 - Feb. 26, 2015 - March 18, 2015
New Story
Filter Board
Quick Links
Todo
Doing
Reviewing
Done
#18 Unit testing for Appointment Request Module
sumanthgundaa, goutham
0 Comments - Tasks
#9 Unit Testing for home pages c student and Instructor
rajeshjonala007, maheswa
0 Comments - Tasks
#8 Home page for student
rajeshjonala007, maheswa
0 Comments - Tasks
#7 Home page for Instructor
rajeshjonala007, maheswa
0 Comments - Tasks
#6 Student page in Request Mod
sumanthgundaa, goutham
0 Comments - Tasks
#5 Instructor page in Request Mc
sumanthgundaa, goutham
0 Comments - Tasks
Project Summary
Epics
Iteration Planning
Chat
History
Predictions
Planning Poker
Iterations
Backlog
0 stories
Integration
2 stories
Designing Modules Phase2
5 stories
Designing Modules Phase1
6 stories
UI and Database Design
5 stories

ScrumDo
Search Project
UMKC
Campus Network
gouthamdonthu
Designing Modules Phase2 - March 19, 2015 - April 8, 2015
New Story
Filter Board
Quick Links
Todo
Doing
Reviewing
Done
#14 Unit Testing for Performance Course Tracking
sumanthgundaa, goutham
0 Comments - Tasks
#13 Student page for Course Tra
rajeshjonala007, maheswa
0 Comments - Tasks
#12 Instructor page for Course tr
rajeshjonala007, maheswa
0 Comments - Tasks
#11 Student page for Performanc Module
sumanthgundaa, goutham
0 Comments - Tasks
#10 Instructor page for performa module
sumanthgundaa, goutham
0 Comments - Tasks
Project Summary
Epics
Iteration Planning
Chat
History
Predictions
Planning Poker
Iterations
Backlog
0 stories
Integration
2 stories
Designing Modules Phase2
5 stories
Designing Modules Phase1
6 stories
UI and Database Design
5 stories



GitHub Site URL:

https://github.com/sumanthgundaa/ASE-Project

Youtube URL:

https://www.youtube.com/watch?v=VFoPULzk6M0

IX. PROJECT MANAGEMENT

Implementation Status Report:

Work Completed:

Story 1: Planning

Description:

Whole Team participates in this story. Here we plan the whole structure of project like what features are we providing and how to implement them.

Responsibility:

Here the project scope and appropriate methods for completing the project is determined.

Time Taken: 6hours

Contributions: All team members equally participated in this story.

Story 2: Master Page Design

Description: Here we create Master page for our project

Responsibility: In this we create common structure for our project

Time Taken: 4 hours

Contributions: Sumanth and Goutham.

Story 3: Creating Home Page

Description: Here we design the Home Page for both Student and Instructor. Navigation in the Home Page changes based on the roles of the user.

Responsibility: Home Page will be designed and the Navigation bar will be changed based on the User Login.

Time Taken: 3hours

Contributions: Mahesh and Rajesh.

Story 4: Creating Login Page

Description: Here we design the login page for the portal.

Responsibility: All the Validations should be done from the database and respective stored procedures should be written to validate the user.

Time Taken: 6hours

Contributions: Sumanth and Goutham.

Story 5: Database Design

Description: All the tables will be created and also the stored procedures will be written.

Responsibility: Table Scripts are created and the sample data will be inserted into the database.

Time Taken: 5hours

Contributions: Mahesh and Rajesh.

Story 6: Instructor Home Page

Description: This is the homepage after Instructor login into his account. In this page he can see all the courses taught by him and corresponding schedule.

Responsibility: All Courses taught by the Instructor are displayed in his home page and the schedule is also available

Time Taken: 9hours

Contributions: Mahesh and Rajesh.

Story 7: Student Home Page

Description: This is the homepage for the student after he logged in. He can see what all courses he enrolled and the syllabus of respective courses. He can also view works to be done.

Responsibility: All Courses registered by the student are displayed in his home page

Time Taken: 8hours

Contributions: Mahesh and Rajesh.

Story 8: Instructor Page in Request Module

Description: We create page for instructor such that he can accept or reject the students request for appointment and he can view all the requests

Responsibility: All the requests are displayed in his page and he can accept or reject the request. After the reply from Instructor result is reflected in student page

Time Taken: 9hours

Contributions: Sumanth and Goutham.

Story 9: Student Page in Request Module

Description: Here student can request for an appointment with desired instructor and he can later view the status of the request

Responsibility: Instructors available timings are displayed in his page so that he can seek appointment. After student requests for appointment it is reflected in Instructors page that particular student is asking for his appointment.

Time Taken: 10hours

Contributions: Sumanth and Goutham.

Story 10: Unit Testing for Home Pages of Student and instructor

Description: Here we test the home pages independently such that they are working correctly according to our specifications without any bugs

Responsibility: Unit testing is done on home pages of student and Instructor

Time Taken: 4hours

Contributions: Mahesh and Rajesh.

Story 11: Unit Testing for Appointment Request Module

Description: Here we test the Instructor and Student pages in Appointment Request module such that they are working correctly according to our specifications without any bugs

Responsibility: Here we do the unit testing for student page and Instructors page in appointment request module. Our goal is to run this module without any bugs.

Time Taken: 5hours

Contributions: Sumanth and Goutham

Story 12: Unit Testing for Home Pages of Student and instructor

Description: Here we test the home pages independently such that they are working correctly according to our specifications without any bugs

Responsibility: Unit testing is done on home pages of student and Instructor

Time Taken: 4hours

Contributions: Mahesh and Rajesh

Story 13: Instructor page for performance module

Description: Here Instructor can track all the students performance and he can generate report of all the students

Responsibility: Instructor can track all the students' performance. We are providing data through excel sheet.

Time Taken: 15 hours

Contributions: Sumanth and Goutham

Story 14: Student page for Performance Module

Description: Here student can view his test scores and he can view his position in the class

Responsibility: Students score can be viewed by him and they are graphically represented

Time Taken: 15hours

Contributions: Sumanth and Goutham

Story 15: Instructor page for Course tracking

Description: Instructor can track the status of the course and he can generate a warning message to all the students about any assignment or project deadlines

Responsibility: Instructor can see how much course has been completed and he can see curriculum of course

Time Taken: 13hours

Contributions: Mahesh and Rajesh

Story 16: Student page for Course Tracking

Description: Here student can track the status of the course and they were eventually notified about their tasks

Responsibility: Student can see how much course has been completed and he can see curriculum of course

Time Taken: 16hours

Contributions: Mahesh and Rajesh

Story 17: Unit Testing for Performance and Course Tracking

Description: Here we test the modules independently such that they are working correctly according to our specifications without any bugs

Responsibility: Here we do the unit testing for student page and Instructors page in performance module. Our goal is to run this module without any bugs

Time Taken: 10hours

Contributions: Sumanth and Goutham

Story 18: Set Attendance

Description: Instructor can set the attendance. He should select course name, date and duration to set the attendance.

Responsibility: Portal identifies coordinate positions of the class room and if student is in that particular location can mark his attendance

Time Taken: 12hours

Contributions: Sumanth and Goutham

Story 19: Attendance List

Description: This belongs to Instructor's page. Here Instructor can check the attendance for any day.

Responsibility: There is an option called generate attendance which means if any students failed to attend the class they will be marked absent for that day.

Time Taken: 10hours

Contributions: Sumanth and Goutham

Story 20: Analyze Attendance

Description: In analyze attendance Instructor can check the whole class attendance and individual attendance

Responsibility: This helps Instructor to know who are the students having low attendance so that he can alert them about losing their grade

Time Taken: 15hours

Contributions: Rajesh and Mahesh

Story 21: Mark Attendance

Description: This belongs to student's side. Students can only mark their attendance if they are in classroom coordinates and by knowing code given by Instructor

Responsibility: Students can mark their attendance only if they are in that particular class and they can also check their overall attendance

Time Taken: 10hours

Contributions: Rajesh and Mahesh

Story 22: Unit Testing for Attendance Module

Description: Here we test the module independently such that it is working correctly according to our specifications without any bugs

Responsibility: Here we do the unit testing for student page and Instructors page in Attendance module. Our goal is to run this module without any bugs

Time Taken: 10hours

Contributions: Sumanth and Goutham

Story 22: Course Plan

Description: Instructor can add a plan for his course through an excel sheet and he can also edit the plan at any time

Responsibility: This feature helps Instructor to add his course plan himself to the portal. If Instructor wants to edit the course plan he can easily change it.

Time Taken: 15hours

Contributions: Sumanth

Future Enhancement:

Each university is maintaining a portal in which Students and Professors can login and do their tasks. But we designed a portal in such a way that it includes more enhanced features like Appointment scheduler, Performance tracker, Attendance tracker etc. Google drive is very useful for providing guidelines about the course to the students and students can post the links of the work done in that. So, as a future enhancement we will try to integrate Google drive with our portal.