Acceptance Test

Requested by: Mr. James Matthews
              Professor
              Siena College
              Computer Science Department

              Dr. Scott Vandenberg
              Head of Department
              Siena College
              Computer Science Department

Programming Contest Submission and Scoreboard

SEG

Prepared by:

Thomas Hyne, Team Leader
Paul Califano
Adam Pasquerella
George Reese
Mark St. Hilaire
Melissa Hoffmann

Presentation:
April 26, 2005
7:00pm – Roger Bacon 328

Programming Contest Submission and Scoreboard Acceptance Test

Table of Contents

Section 1: Product Overview and Summary

Each school year the Siena College Computer Science Department hosts a programming contest for local High Schools. Teams made up of High School students are placed in separate locations and given problems to solve. When they have a solution it will be submitted to a board of judges. After the judges have reviewed the solution they will determine if it is correct and a message will be sent back to the students stating if the solution was accepted or not. This process was previously used as a web-based application that was poorly organized and very difficult to manage. Communication between teams and judges was done through email and teams were not able to re-submit incorrect solutions if a judge had opened a previous submission.

The Program Contest Submission and Scoreboard will not be using email as a way of communication. This will allow for messages to be sent smoothly between teams and judges. Teams and judges will use GUIs in order to navigate through the processes that they need when submitting a solution, clarifying a question, or sending a response. There will also be a scoreboard, which will update automatically showing rankings of the teams based on number of problems they answered correctly and the time they have spent.

Section 2: Development, Operating and Maintenance Environments

This system was developed on the workstations in the Siena College Software Engineering Lab. All data stores are in directories on the Turing server, any web page development and maintenance was done using Dream Weaver, and the Graphical User Interfaces were programmed in HTML with PHP using Dream Weaver.

The teams and the judges will mainly be using the Programming Contest Submission and Scoreboard software. Outside users may view the contest website to see past contest problems, to learn the history of the contest or during a current contest to see the up-to-date scoreboard. Both the teams and the judges will be given a set of GUIs to use during a contest. The teams set will include GUIs to submit solutions, ask questions to the judge and to view the updated scoreboard. The judges set will include GUIs to reply to the team with either a problem clarification or a response to the solution. The judges will be able to view a queue that will show the problems that have been graded or need to be graded, and the problems that need clarification. Everyone with access to the Internet will be able to view the scoreboard of the current contest. This program will be able to do this as well as have "hooks" to allow expansion in the future.

Section 3: Functional Requirements Review

The Programming Contest Submission and Scoreboard will serve four types of users, teams in the contest, judges facilitating the contest, outside users, and an administrator. The following is a list of functionality requirements that the system will perform for each user:

- Teams in the Contest:
  1. A Graphical User Interface will be displayed when the teams log into the system when the contest begins that has the status of all the teams submitted solutions, a link for the team to submit a solution and a link to submit a question, and a message board.
  2. The submission link will bring the team to another web page that allows teams to select the problem they are submitting, the language in which it was produced. A file based on what the team selected is sent to the judges when they submit. When the teams submit the problem, the file being submitted will be copied onto a directory called "Submitted but not Judged", on Oraserv.
  3. The clarification link will allow teams to select a problem and enter a question into a text box to be sent to the Judges.
  4. The Message Board will be a place where Teams can see the questions asked to the Judges by other teams, with the Judges clarification.
  5. The Scoreboard allows each team to see the standings of the contest, based on the number of problems that have been answered correctly and the cumulative time used.

- Judges of the Contest
  1. The judges GUI contains a web page that has a file browser that the judges will use to receive a teams submitted solutions and submitted questions.
  2. After the judges run and evaluate the solution file, they can automatically notify the team of the status of the solution.
  3. The file browser will enable them to receive text files with questions from the teams about specific problems. They can open these files and amend a response, which can be viewed by the teams.
  4. The judges will be able to send an amended response to either a public message board, or can be sent privately to the team who asked the question.

- Outside Users (including teams and judges before the contest)
  1. Outside users can view a website that will contain history of the contest, old contest problems, and how you sign up to join the contest.

2. A Web-based Scoreboard will be created to allow any individual, in the contest or not, to view the standings of a contest at it is being run. Teams will be sorted based on their ranks, which are determined by the number of questions answered correctly and the cumulative about of time to answer those questions.

- Administrator:
  1. The Administrator must update all of the GUI's to make sure all of the correct information is there, including anything new for an upcoming contest (number of problems, programming language, judges responses, etc.).
  2. Update the website with more problems from previous contest and new winners from the contests.
  3. Insert the names of the teams on the scoreboard of an upcoming contest.
  4. Make sure all of the correct information is displayed on the message board and the scoreboard.
  5. Make sure the copying and moving of the files from directory to directory is being done correctly on Oraserv. This should be tested before the contest begins, as well as being monitored during the contest.

Functional Requirements Test:

1. When the Team logs in, does a GUI display that allows them to submit problem solutions and inquire about difficulties they run into?	☑ Yes ☐ No
2. Prior to submitting a solution, can the team select the proper problem number and language that the solution was written in?	☑ Yes ☐ No
3. Are the teams able to submit a question to the judges and specify which problem it corresponds to?	☑ Yes ☐ No
4. Does the message board display the questions asked to the judges by other teams with the judge's clarification?	☑ Yes ☐ No
5. Does the scoreboard allow each team to see the standings of the contest based on the number of problems that have been answered correctly?	☑ Yes ☐ No
6. Does the scoreboard rank the teams in the order?	☐ Yes ☑ No
7. Does the judge's file browser receive a team's submitted solutions and submitted questions?	☑ Yes ☐ No
8. After the judge's run and evaluate the solution file can they automatically notify the team of the status of the solution?	☑ Yes ☐ No

9. Does the file browser enable the judges to receive text files with questions from teams about specific problems?	☑ Yes ☐ No
10. Can the judges open these files and amend a response that can be viewed by the teams?	☑ Yes ☐ No
11. Can the judge send an amended response to the public message board?	☑ Yes ☐ No
12. Can the judge send an amended response privately to the team who had asked the question?	☐ Yes ☑ No
13. Can the outside users view a website that contains the history of the contest, old contest problems, and how to sign up to join the contest?	☑ Yes ☐ No
14. Can the outside user view the Web-based scoreboard that contains the standings of the contest as it is being run?	☑ Yes ☐ No
15. Can the administrator update all of the GUIs to make sure that the correct information is there, including anything new for an upcoming contest?	☑ Yes ☐ No
16. Can the administrator update the website with more problems from previous contests and new winners from the contests?	☑ Yes ☐ No
17. Can the administrator insert the names of the teams on the scoreboard of an upcoming test?	☑ Yes ☐ No
18. Can the administrator make sure that all of the correct information is displayed on the message board and the scoreboard?	☑ Yes ☐ No
19. Can the administrator make sure that the copying and moving of the files from directory to directory is being done correctly on Oraserv?	☑ Yes ☐ No

Section 4: Acceptance Criteria Review

The Programming Contest Submission and Scoreboard will allow each user to complete the following tasks within the application:

- Teams to:
    1. Securely login to their team home page, ensuring a lower security risk.
    2. Select whether they want to submit a solution to a problem or ask a question about a problem to the judges.
    3. Submit a solution to the judges, for one particular problem, and also indicating which programming language it was developed with. Teams will be notified to use a specific naming convention when naming their solutions, which will allow the submission application to find the file and copy it into the judge's directory.

4. Ask a question to the judges about any problem by selecting the problem they have a question about, and entering text describing their inquiry;
5. View a message board of all question asked by other teams, as well as the judges response, which should be ordered by the problem number;
6. View the scoreboard of the contest they are competing in, which ranks teams based on number of problems answered correctly and the amount of time used to produce their solutions.
7. View the website with history of the contest, previous problems, and previous winners, before the day of the contest;

- Judges to:
  1. Select a solution to grade from a queue, check the correctness of the solution, and then send an automated response based on whether the solution was correct or not. If the response indicates that the solution is correct, the scoreboard will be automatically updated;
  2. Select an inquiry from a team to clarify from a queue and answer the question. When answering the question, the judges may send an automated response by typing in text to respond to just the team that asked the question, or post it to the message board to allow it to be viewed by every team;
  3. View the scoreboard to see the rankings of the team, as well as make sure all the information is correct;

- Outside Users to:
  1. View the scoreboard of the current contest;
  2. View the website with history of the contest, previous problems, and previous winners.

- Administrator to:
  1. Make sure that the right information being posted to the message board and the scoreboard is accurate.
  2. View files in the specific directories on the Oraserv machine, making sure they are being copied and moved correctly throughout the process.
  3. Update the GUI's with anything new (new judge responses, number of problems in the contest, programming languages to be used, etc).
  4. Update the website with previous problems from prior contests as well as update the winners on the website.
  5. Insert new names of the teams on the scoreboard.

Acceptance Criteria Document

Teams To:

Course of action	Pass/Fail		Comments
Securely login to their team home page, ensuring a lower security risk?	**P**	F	
Select whether they want to submit a solution to a problem or ask a question about a problem to the judges?	**P**	F	
Submit a solution to the judges, for one particular problem, and also indicating which programming language it was developed with. Teams will be notified to use a specific naming convention when naming their solutions, which will allow the submission application to find the file and copy it into the judge's directory?	**P**	F	
Ask a question to the judges about any problem by selecting the problem they have a question about, and entering text describing their inquiry?	**P**	F	
View a message board of all question asked by other teams, as well as the judges response, which should be ordered by the problem number?	**P**	F	
View the scoreboard of the contest they are competing in, which ranks teams based on number of problems answered correctly and the amount of time used to produce their solutions?	**P**	F	
View the website with history of the contest, previous problems, and previous winners, before the day of the contest?	**P**	F	

Judges To:

Course of action	Pass/Fail		Comments
Select a solution to grade from a queue, check the correctness of the solution, and then send an automated response based on whether the solution was correct or not. If the response indicates that the solution is correct, the scoreboard will be automatically updated?	**P**	F	

Select an inquiry from a team to clarify from a queue and answer the question. When answering the question, the judges may send an automated response by typing in text to respond to just the team that asked the question, or post it to the message board to allow it to be viewed by every team?	**P** F	
View the scoreboard to see the rankings of the team, as well as make sure all the information is correct?	**P** F	

Outside Viewers to:

Course of action	Pass/Fail	Comments
View the scoreboard of the current contest?	**P** F	
View the website with history of the contest, previous problems, and previous winners?	**P** F	

Administrator to:

Course of action	Pass/Fail	Comments
Make sure that the right information being posted to the message board and the scoreboard is accurate?	**P** F	
View files in the specific directories on the Oraserv machine, making sure they are being copied and moved correctly throughout the process?	**P** F	
Update the GUI's with anything new (new judge responses, number of problems in the contest, programming languages to be used, etc)?	**P** F	
Update the website with previous problems from prior contests as well as update the winners on the website?	**P** F	
Insert new names of the teams on the scoreboard?	**P** F	

Section 5: Website Testing Document

In order for the Siena College High School Programming Contest Software System to be acceptable for delivery, SEG has rigorously tested each web page included in the system. The following documents (found on pages 7 - 10) detail the testing that we did.

Test One: Test to ensure that every entity in the team website works as expected, starting from the status bars showing the correct status to when a team chooses to view the scoreboard.

Course of action	Excepted Results	Pass/Fail	Comments
Enter Team 1 Username and Password	Team 1 should login to Team 01 Home Page	**P** F	
Enter Team 2 Username and Password	Team 2 should login to Team 02 Home Page	**P** F	
Enter Team 3 Username and Password	Team 3 should login to Team 03 Home Page	**P** F	
Enter Team 4 Username and Password	Team 4 should login to Team 04 Home Page	**P** F	
Enter Team 5 Username and Password	Team 5 should login to Team 05 Home Page	**P** F	
Enter Team 6 Username and Password	Team 6 should login to Team 06 Home Page	**P** F	

Enter Team 7 Username and Password	Team 7 should login to Team 07 Home Page	**P**	F
Enter Team 8 Username and Password	Team 8 should login to Team 08 Home Page	**P**	F
Enter Team 9 Username and Password	Team 9 should login to Team 09 Home Page	**P**	F
Enter Team 10 Username and Password	Team 10 should login to Team 10 Home Page	**P**	F
Enter Team 11 Username and Password	Team 11 should login to Team 11 Home Page	**P**	F
Enter Team 12 Username and Password	Team 12 should login to Team 12 Home Page	**P**	F
Enter Team 13 Username and Password	Team 13 should login to Team 13 Home Page	**P**	F
Enter Team 14 Username and Password	Team 14 should login to Team 14 Home Page	**P**	F
Enter Team 15 Username and Password	Team 15 should login to Team 15 Home Page	**P**	F
Enter Team 16 Username and Password	Team 16 should login to Team 16 Home Page	**P**	F
Enter Team 17 Username and Password	Team 17 should login to Team 17 Home Page	**P**	F
Enter Team 18 Username and Password	Team 18 should login to Team 18 Home Page	**P**	F
Enter Team 19 Username and Password	Team 19 should login to Team 19 Home Page	**P**	F
Enter Team 20 Username and Password	Team 20 should login to Team 20 Home Page	**P**	F

Enter Team 21 Username and Password	Team 21 should login to Team 21 Home Page	**P**	F	
Enter Team 22 Username and Password	Team 22 should login to Team 22 Home Page	**P**	F	
Enter Team 23 Username and Password	Team 23 should login to Team 23 Home Page	**P**	F	
Enter Team 24 Username and Password	Team 24 should login to Team 24 Home Page	**P**	F	
Enter Team 25 Username and Password	Team 25 should login to Team 25 Home Page	**P**	F	
Enter an incorrect username or password.	Login error is received with a link to go back to the login page.	**P**	F	
Enter a correct username but no password	Login error is received with a link to go back to the login page.	**P**	F	
User tries to directly access TeamHome.php without logging in.	Error page is displayed and directs user to go back and log in.	**P**	F	
User closes team GUI	User is logged out of the system.	**P**	F	
Select "Submit a Solution" from Home Page	Team solution page should be displayed.	**P**	F	
Select radio button on solutions page.	Only one can be selected	**P**	F	
To submit a solution: 1. A language and problem must be selected. 2. Click on "Submit"	Solution is sent to the Judges panel and the page is refreshed with updated status bar.	**P**	F	

If "Submit" is clicked without selecting a language	Error: You must select a language for your solution	P	**F**	Page refreshes with an unsuccessful submission, but no error.
Select and submit P1.cpp	Status bar must update for each of the following events: Not Submitted Submitted Being Judged Correct Incorrect Incomplete Output Incorrect Output Incorrect Output Format Runtime Error CPU Time Exceeded	**P**	F	
Select and submit P1.java	Status bar must update for each of the following events: Not Submitted Submitted Being Judged Correct Incorrect Incomplete Output Incorrect Output Incorrect Output Format Runtime Error CPU Time Exceeded	**P**	F	
Select and submit P2.cpp	Status bar must update for each of the following events:	**P**	F	

	Not Submitted
	Submitted
	Being Judged
	Correct
	Incorrect
	Incomplete
	     Output
	Incorrect Output
	Incorrect Output
	     Format
	Runtime Error
	CPU Time
	     Exceeded		

Select and submit P2.java	Status bar must update for each of the following events:
	
	Not Submitted
	Submitted
	Being Judged
	Correct
	Incorrect
	Incomplete
	     Output
	Incorrect Output
	Incorrect Output
	     Format
	Runtime Error
	CPU Time
	     Exceeded
		P F	

Select and submit P3.cpp	Status bar must update for each of the following events: Not Submitted Submitted Being Judged Correct Incorrect Incomplete Output Incorrect Output Incorrect Output	**P** F	

	Format Runtime Error CPU Time Exceeded		

Select and submit
P3.java
	Status bar must
	update for each of
	the following
	events:
	
	    Not Submitted
	    Submitted
	    Being Judged
	    Correct
	    Incorrect
	    Incomplete
	        Output
	    Incorrect Output
	    Incorrect Output
	        Format
	    Runtime Error
	    CPU Time
	        Exceeded
		P F	

Select and submit
P4.cpp
	Status bar must
	update for each of
	the following
	events:
	
	    Not Submitted
	    Submitted
	    Being Judged
	    Correct
	    Incorrect
	    Incomplete
	        Output
	    Incorrect Output
	    Incorrect Output
	        Format
	    Runtime Error
	    CPU Time
	        Exceeded
		P F	

Select and submit P4.java	Status bar must update for each of the following events:	**P** F	

	Not Submitted
	Submitted
	Being Judged
	Correct
	Incorrect
	Incomplete
	    Output
	Incorrect Output
	Incorrect Output
	    Format
	Runtime Error
	CPU Time
	    Exceeded			

Select and submit
P5.cpp
	Status bar must
	update for each of
	the following
	events:
	
	Not Submitted
	Submitted
	Being Judged
	Correct
	Incorrect
	Incomplete
	    Output
	Incorrect Output
	Incorrect Output
	    Format
	Runtime Error
	CPU Time
	    Exceeded
		P	F	

Select and submit P5.java	Status bar must update for each of the following events: Not Submitted Submitted Being Judged Correct Incorrect Incomplete Output Incorrect Output	**P**	F	

	Incorrect Output Format Runtime Error CPU Time Exceeded			
Select and submit P6.cpp	Status bar must update for each of the following events: Not Submitted Submitted Being Judged Correct Incorrect Incomplete Output Incorrect Output Incorrect Output Format Runtime Error CPU Time Exceeded	**P**	F	
Select and submit P6.java	Status bar must update for each of the following events: Not Submitted Submitted Being Judged Correct Incorrect Incomplete Output Incorrect Output Incorrect Output Format Runtime Error CPU Time Exceeded	**P**	F	
Select and submit P7.cpp	Status bar must update for each of	**P**	F	

	the following events:
	
	    Not Submitted
	    Submitted
	    Being Judged
	    Correct
	    Incorrect
	    Incomplete
	        Output
	    Incorrect Output
	    Incorrect Output
	        Format
	    Runtime Error
	    CPU Time
	        Exceeded		

Select and submit P7.java	Status bar must update for each of the following events:
	
	    Not Submitted
	    Submitted
	    Being Judged
	    Correct
	    Incorrect
	    Incomplete
	        Output
	    Incorrect Output
	    Incorrect Output
	        Format
	    Runtime Error
	    CPU Time
	        Exceeded
		P F	

Click "Scoreboard" button on solutions page.	Scoreboard will be displayed in a new window.	**P** F	
Click "Home" button on solution	Team Home page should be displayed.	**P** F	

page.		**P**	F	
Click "Submit a Question" on team home page.	Team question page should be displayed.	**P**	F	
Select radio button on questions page.	Only one can be selected	**P**	F	
Use of text box.	User should be able to type a question into the text box.	**P**	F	
User browses links on message board.	There are no broken links and all messages that are on the server are displayed to the user.	**P**	F	
Submission of question	After user enters a question about a particular problem and selects corresponding radio button, question should be submitted to judges.	**P**	F	
Click "Home" button on question page.	Team Home page should be displayed.	**P**	F	
Click "Scoreboard" button on question page.	Scoreboard will be displayed in a new window.	**P**	F	
Click "Scoreboard" button on Team Home page.	Scoreboard will be displayed in a new window.	**P**	F	
Select problem in message board.	Window will show all responses to any questions relating to that problem	**P**	F	
Select a response.	Response and corresponding question will be displayed.	**P**	F	

Test 2: Test to ensure that every entity in the judge website works as expected, starting from viewing items in the queues to sending clarification answers to correct message board.

Course of action	Expected Result	Pass/Fail		Comments
Enter URL to access main judge page.	Judges home page should load showing a link to solutions and questions.	**P**	F	
Check problems queue	Clicking on link to "Problems" folder should display "submitted_not_judged" link, and links to go up one level and back home.	**P**	F	
Click "submitted_not_judged" folder	Clicking this folder should display all solutions that have been submitted, but not yet judged. Also, "Being Judged" folder should be displayed.	**P**	F	
Select solution and select "Being Judged" folder. Then select "move".	This action should prep the solution to be judged by moving into the correct location.	**P**	F	
Select "Being Judged" folder.	This should display all solutions currently being graded, and a list of all correct/incorrect categorized folders.	**P**	F	
After judging, move solution to an incorrect folder.	This will send a message to the team's status bar and notify the team that their solution is incorrect.	**P**	F	

After judging, move solution to a correct folder.	This will send a message to the team's status bar and notify the team that their solution is correct and will also update the scoreboard.	**P** F	
Click on any categorized folder inside "Being Judged"	Will display all judged solutions within that category.	**P** F	
Check questions queue	Click on link to "Questions" folder should display single folder "Questions to be Answered"	**P** F	
Select "Questions to be Answered" link	Shows all questions that have not been responded to and the "Questions with Answer" folder.	**P** F	
Select "E" (for edit) on an unanswered question.	Question is displayed in an editor window.	**P** F	
Type response to question beneath question in editor window and click the save button.	Response to question is saved beneath the question. The judge is returned to the page that lists of all the questions.	**P** F	
Select checkbox for answered question and radio button for "Questions with Answer". Then click "Move"	Question is moved to "Question with Answer" Folder	**P** F	
After moving down to "Questions with Answer" folder, select checkbox for same question and radio button for corresponding problem number. Then click "Move"	Question is moved to problem number that is relates to. Teams can now see this in that problem's folder on the message board.	**P** F	

Test 3: Test to make sure that the scoreboard is viewable and the most importantly that the information in there is correct and accurate, starting from displaying the information in a two-dimensional array to ranking the teams in the correct standing order.

Course of action	Expected Result	Pass/Fail		Comments
Display Scoreboard	1. The information is viewable and is in a two-dimensional array (matrix)	**P**	F	
Scoring	Correct calculation	**P**	F	
Timing	Correct calculation of timing	**P**	F	
Ranking	Correct ranking based on timing and scoring	**P**	F	
Updates	The scoreboard updates constantly and continues to rank teams accordingly.	**P**	F	

Section 6: Prototypes

Team Web Sites

Team Login Page

This is the first page that a team will encounter on the programming contest system.

Team Home Page

This is the first page the team sees upon logging in. From this page, the team can choose to <u>Submit a Solution</u> or <u>Submit a Question</u>. They can also view all questions that have been answered by the judges, and see the status of their submitted solution as it progresses through the judging process.

Team Submit Solution

This page will be displayed when clicking on "Submit a Solution".

Team Submit Questions

This page will be displayed when clicking on "Submit a Question".

Not Logged In

File   Edit   View   Favorites   Tools   Help

Address http://oraserv.cs.siena.edu/~perm_seg/NoLogin.html

Error

You have not logged in.

Please log in using your provided username and password.

Login Page
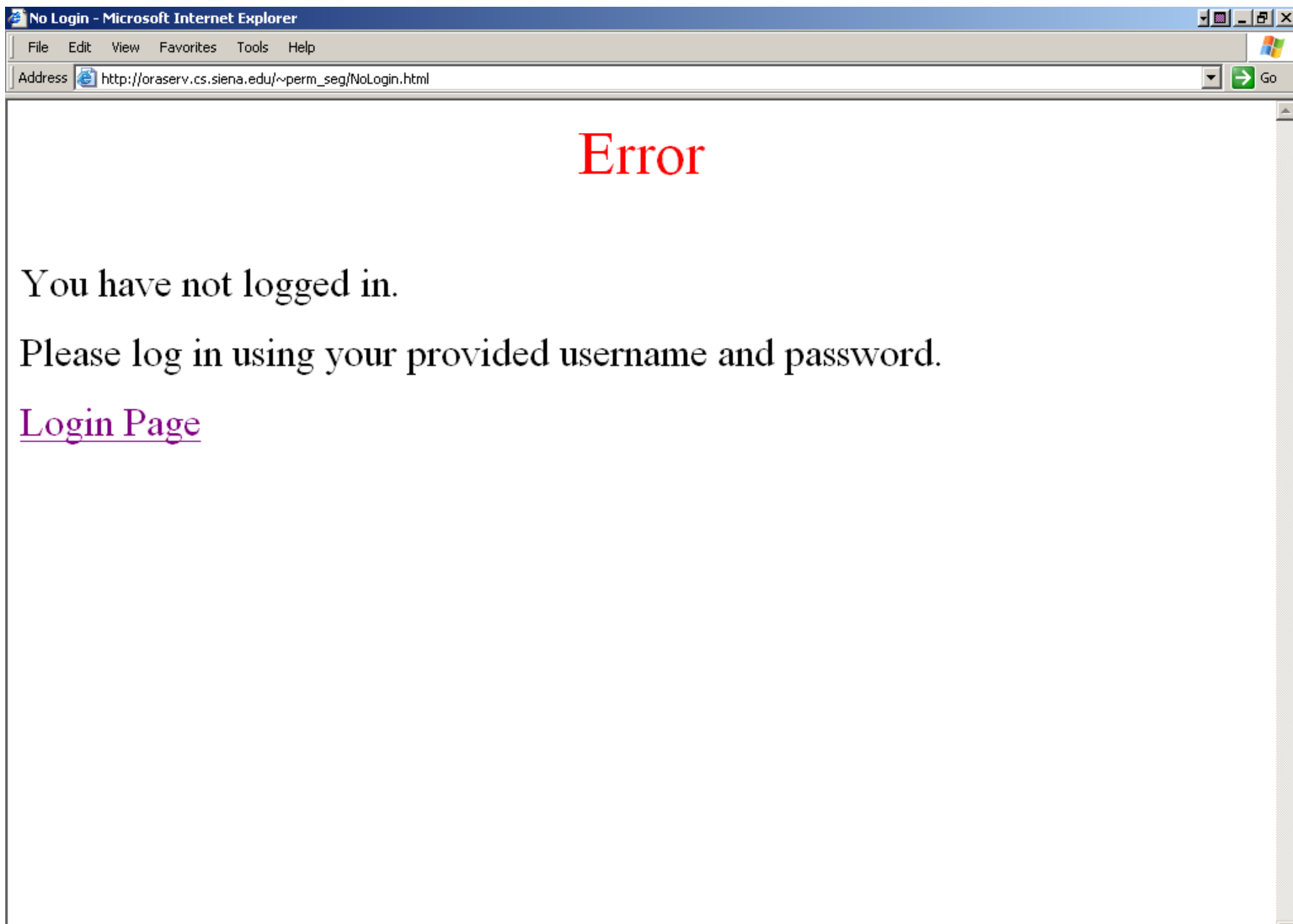
If a team maliciously tries to access the TeamHome.php file without logging in, they will encounter an error.

Incorrect Login
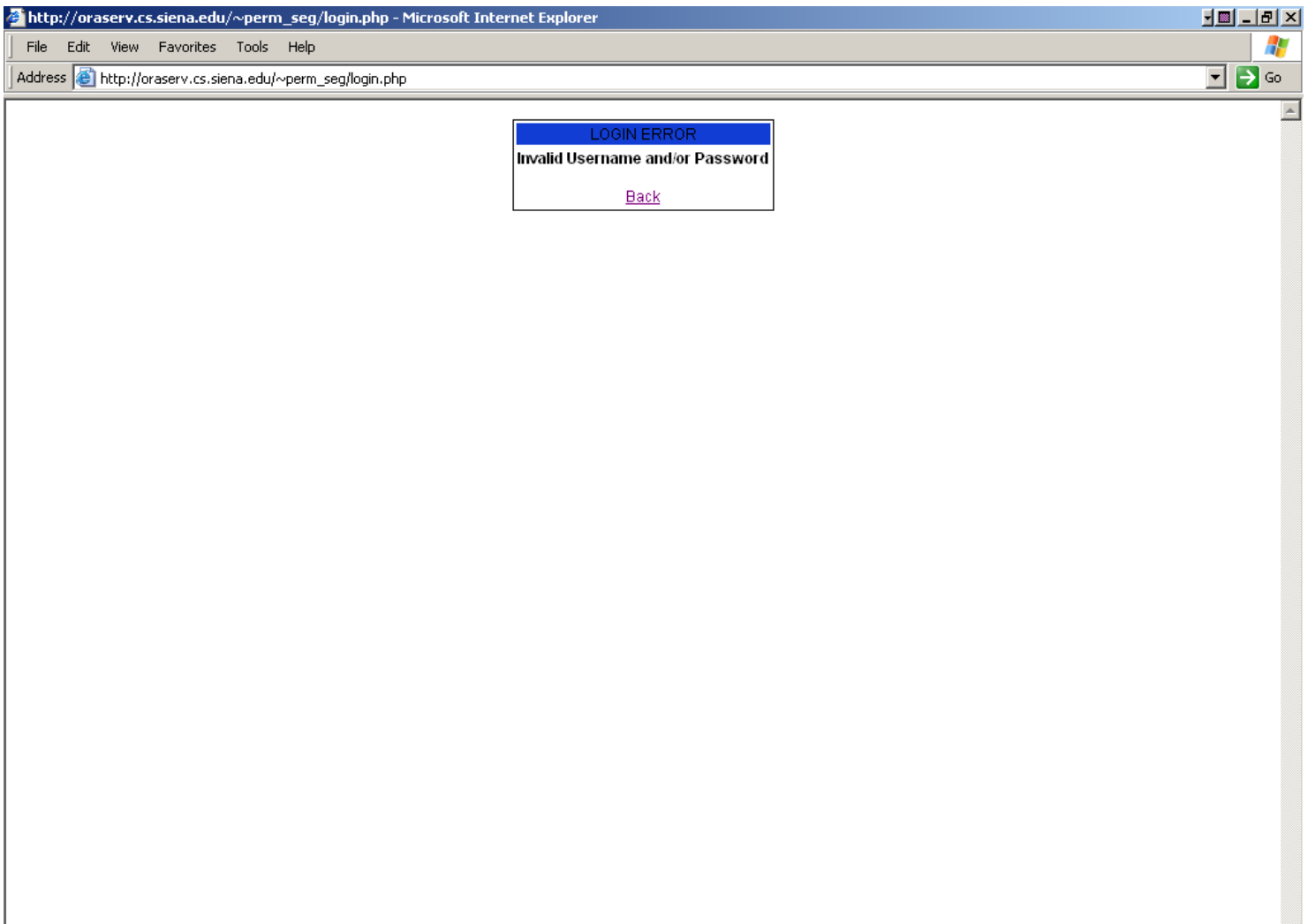
If a team does not enter their correct username and/or password, this error will be displayed.

<u>Judges Websites</u>

Judges Home
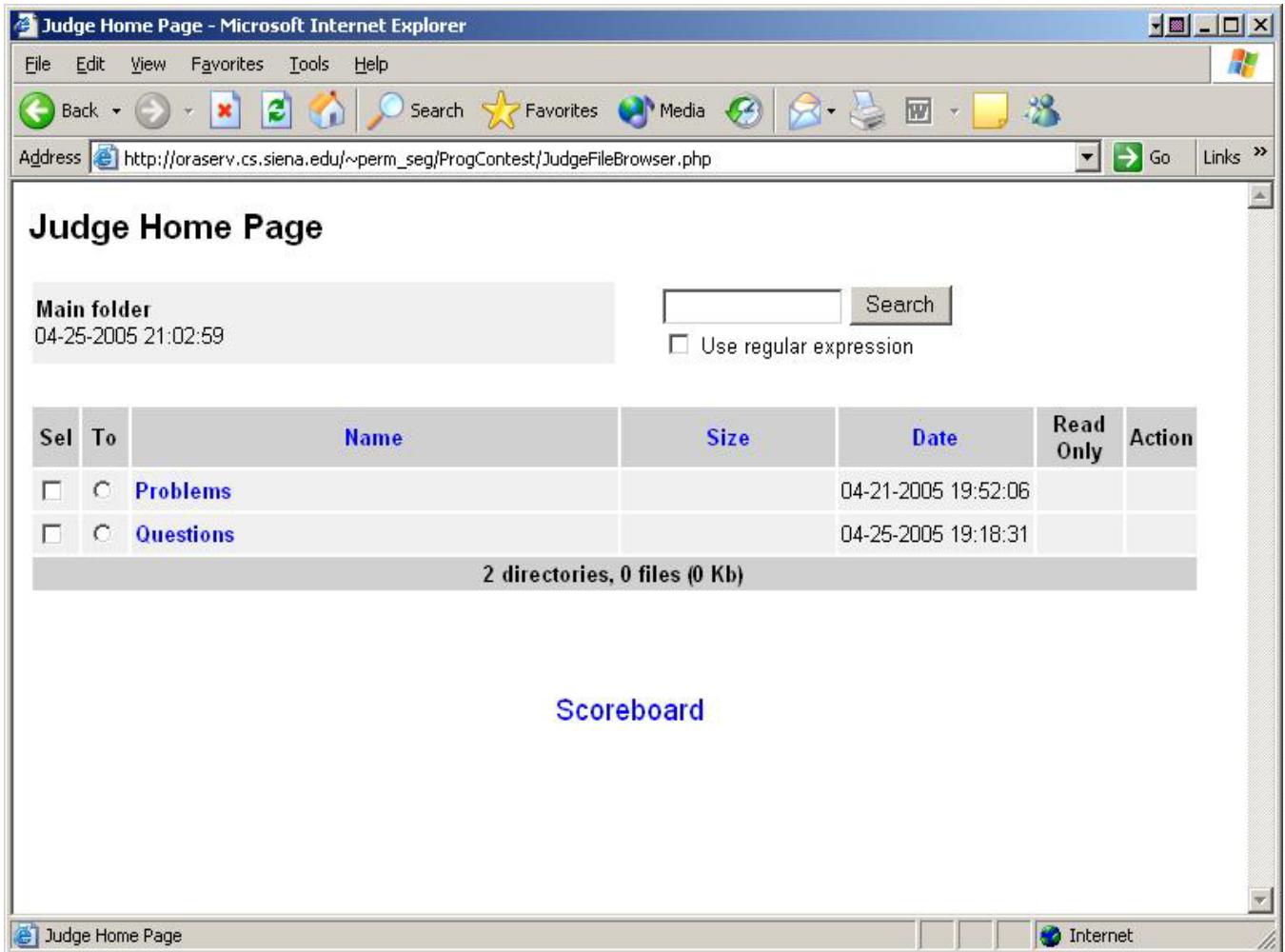
When accessing the URL /JudgeFileBrowser.php, this page will be displayed.

Judges Next Page
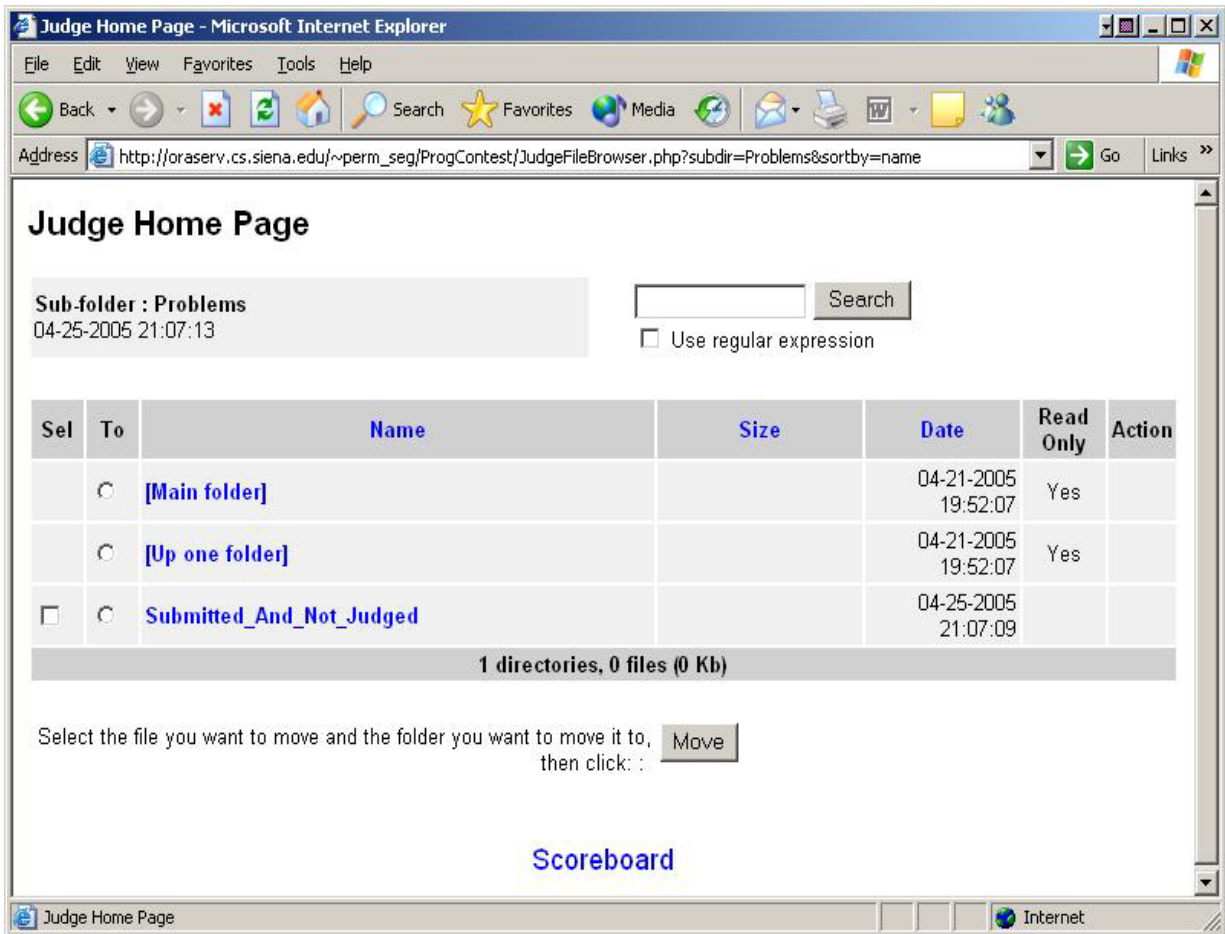
When clicking the "Problems" link, this page is displayed.

Judges Submitted and not Judged Page
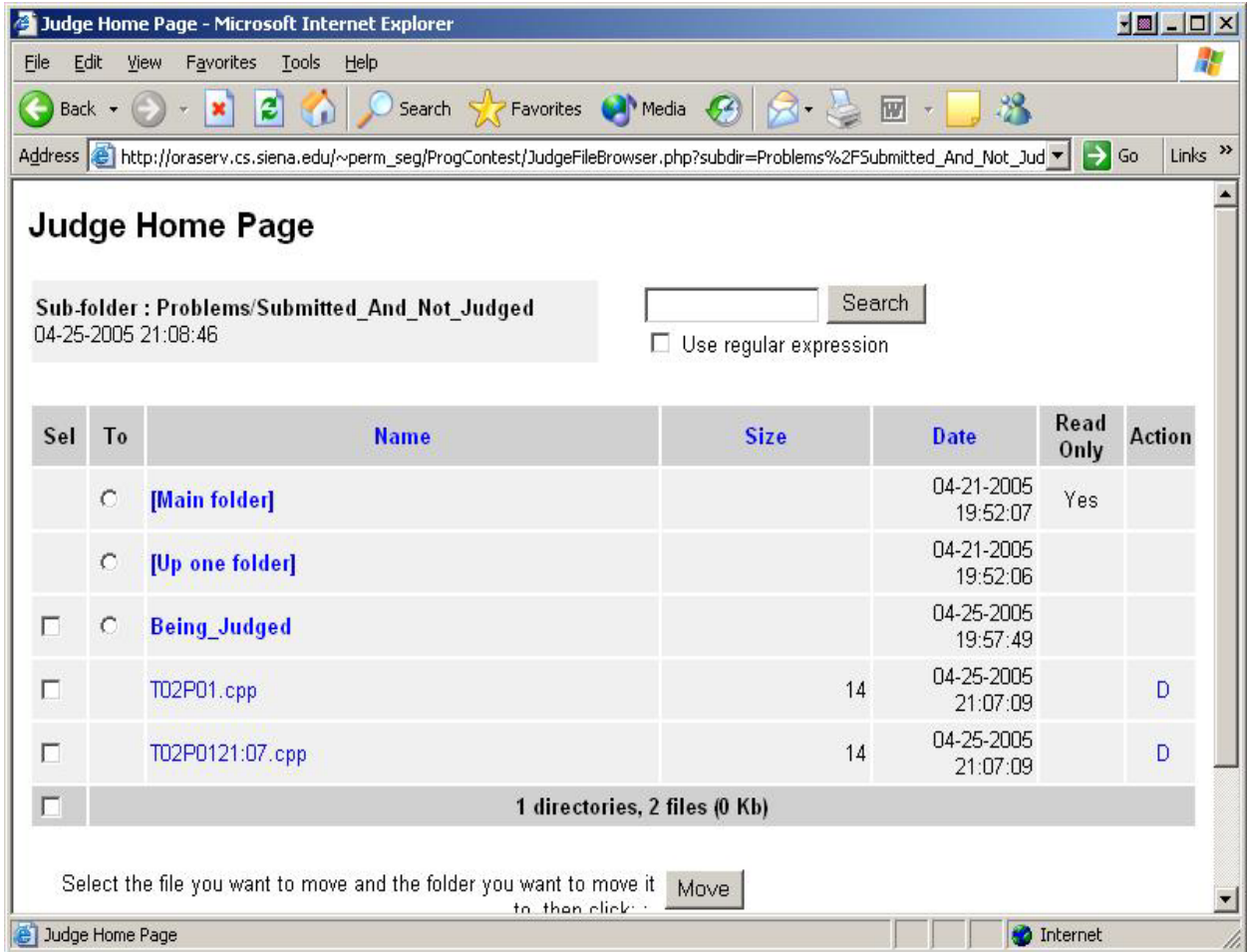
When clicking the "Submitted_Not_Judged" folder, all submissions that have yet to be looked at by the judges are displayed.

Judges Move to being Judge Page
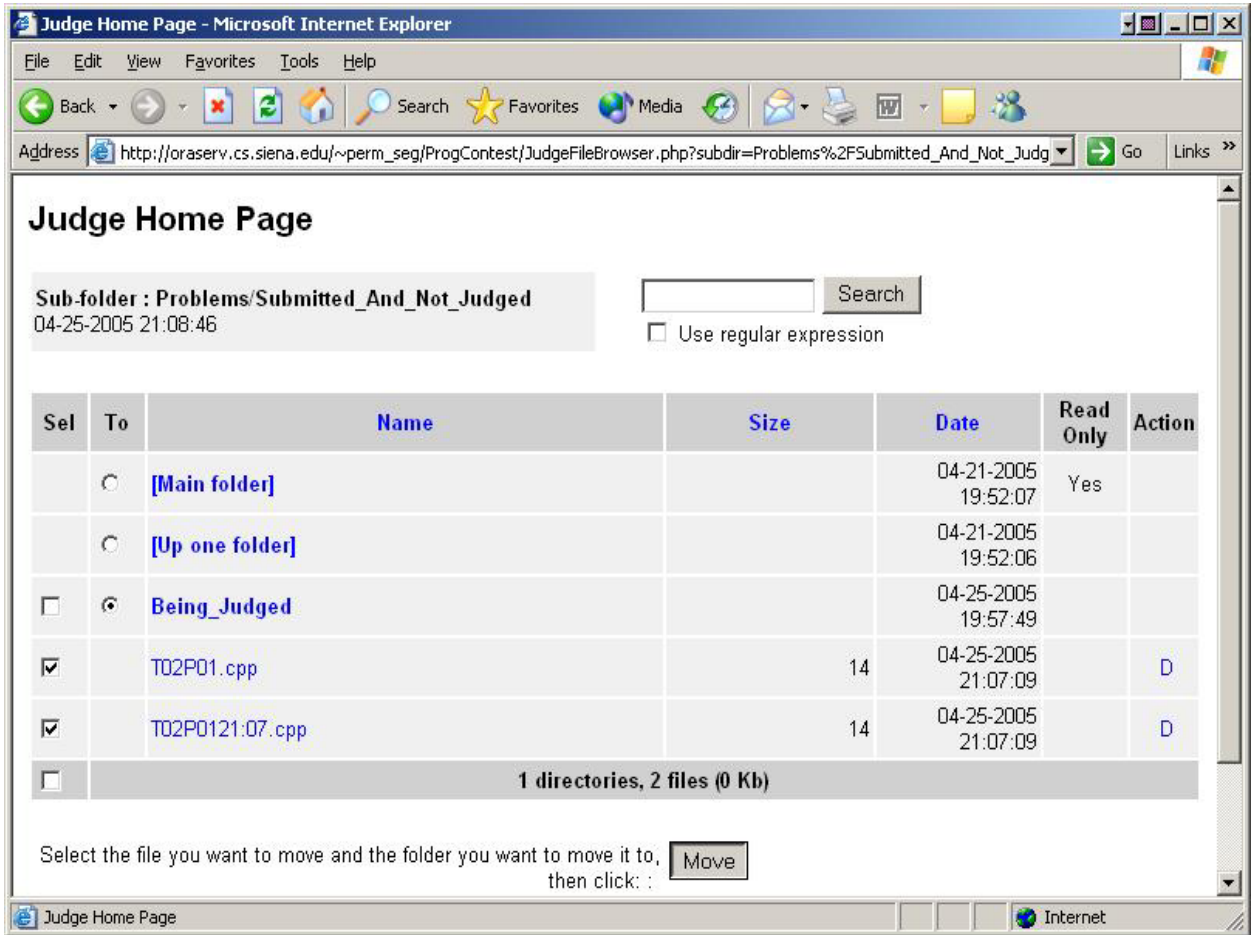
The judges being the grading process by selecting the checkbox of the solution, then selecting the radio button for the "Being_Judged" directory, then clicking the "Move" button. This ensures only one judge will be working on the same file at a time.

Once the judge compiles and runs the solution file, they move the solution file to the folder which corresponds to its performance. If it is incorrect, they can essentially describe what it did wrong by placing it in that folder.
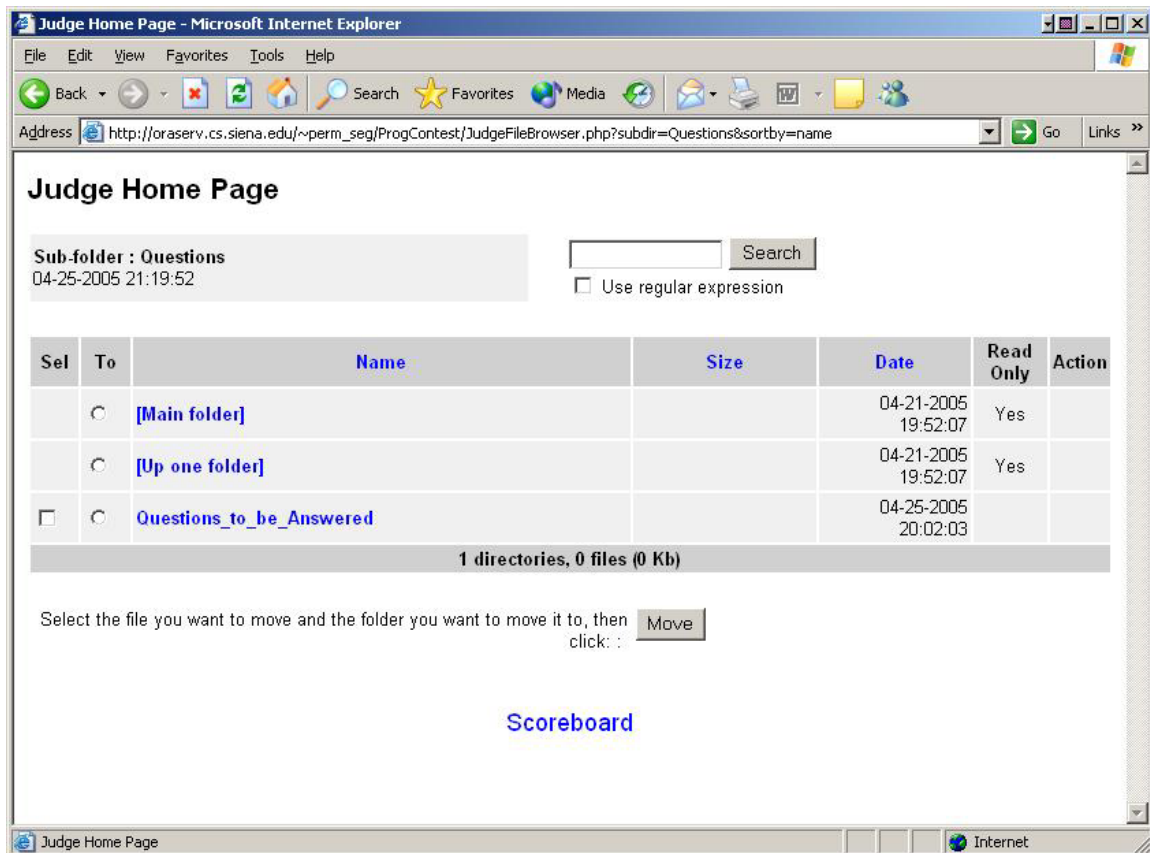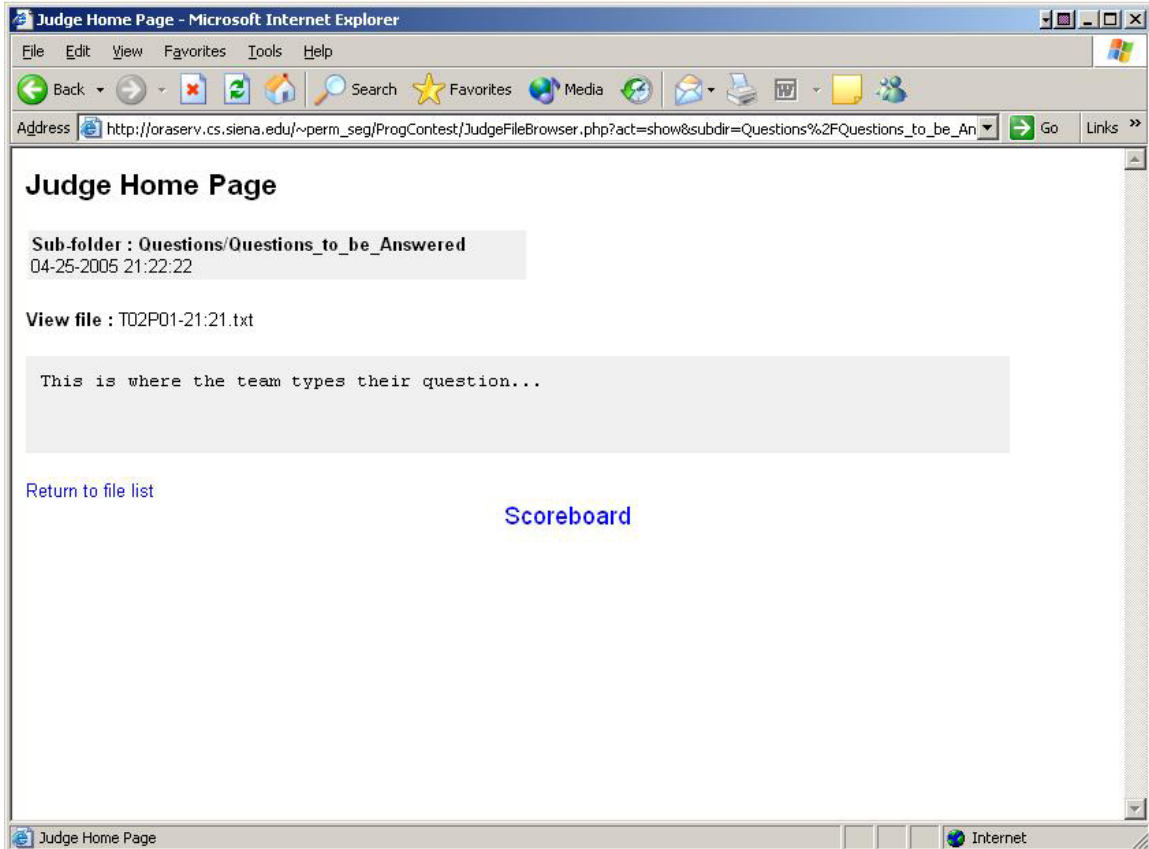
From the judge's home page, if the "Questions" link is clicked, this page will be displayed. To display a list of submitted questions, the judges must select the "Questions_To_Be_Answered" directory.
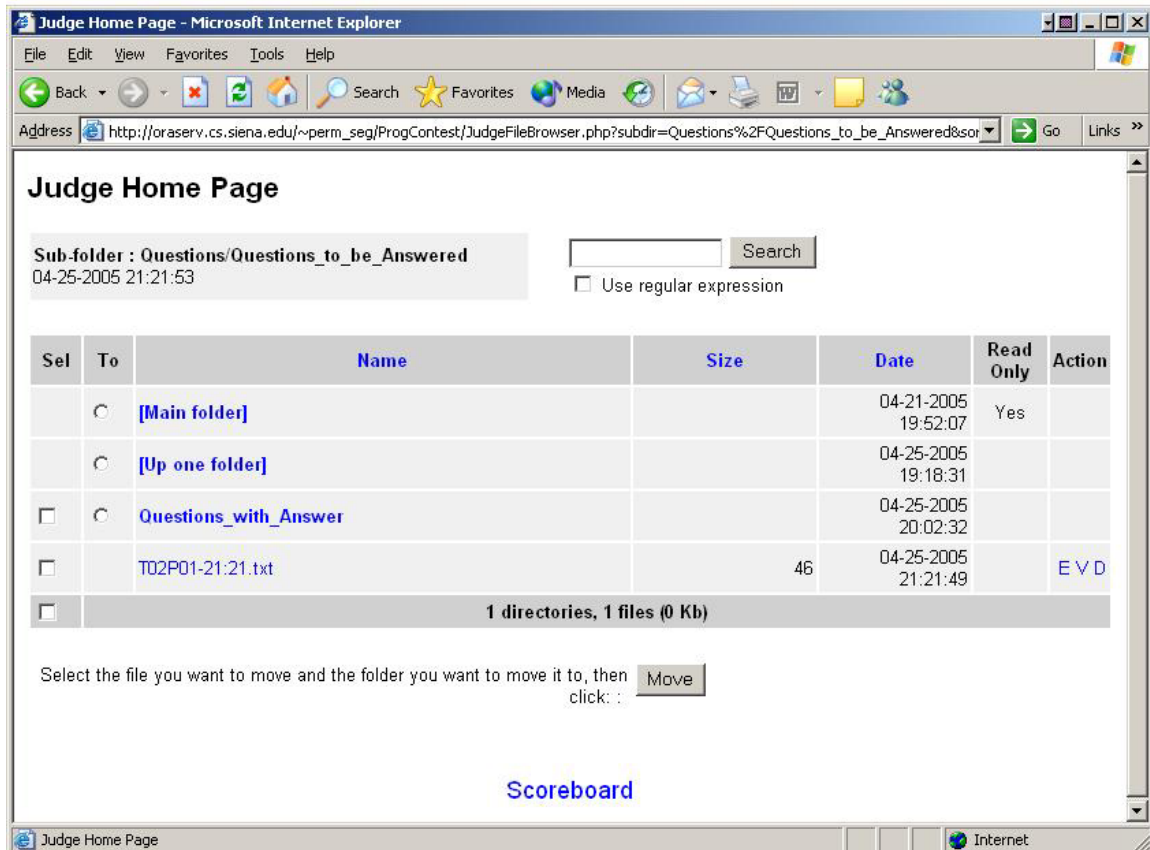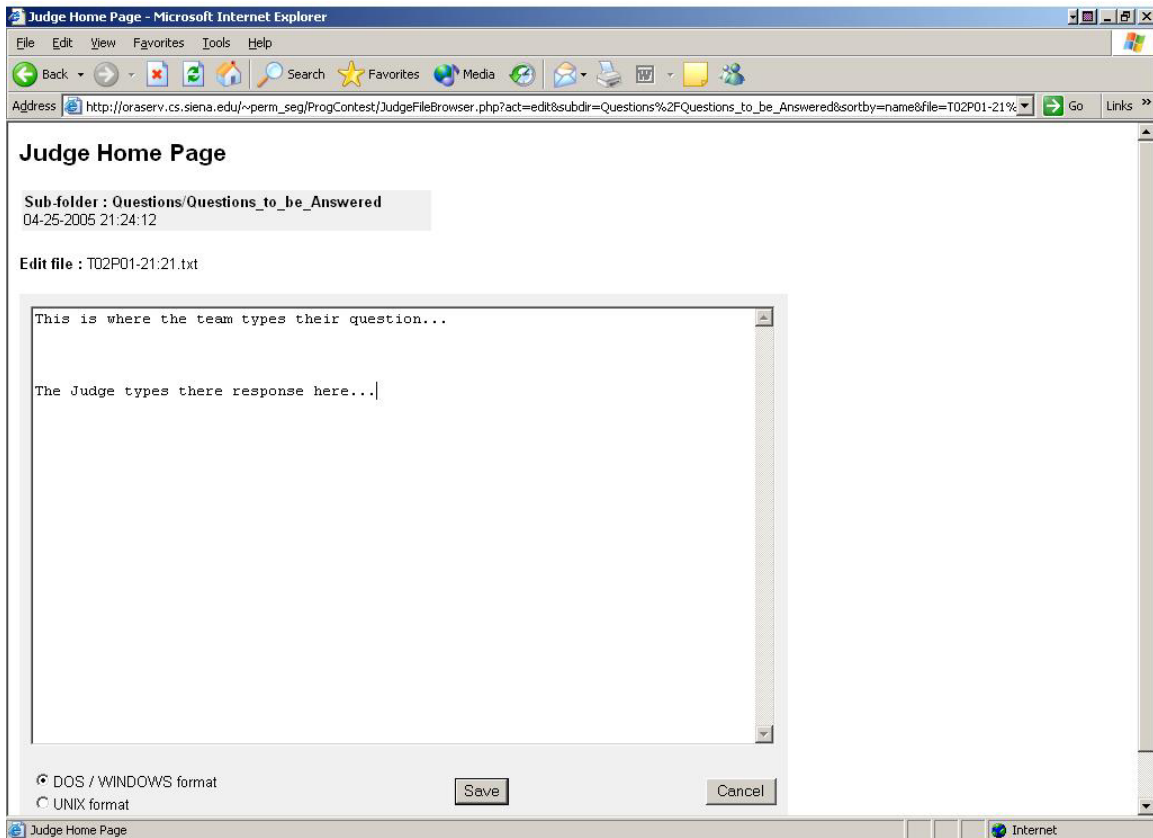
This page will display the list of all questions submitted by the teams.
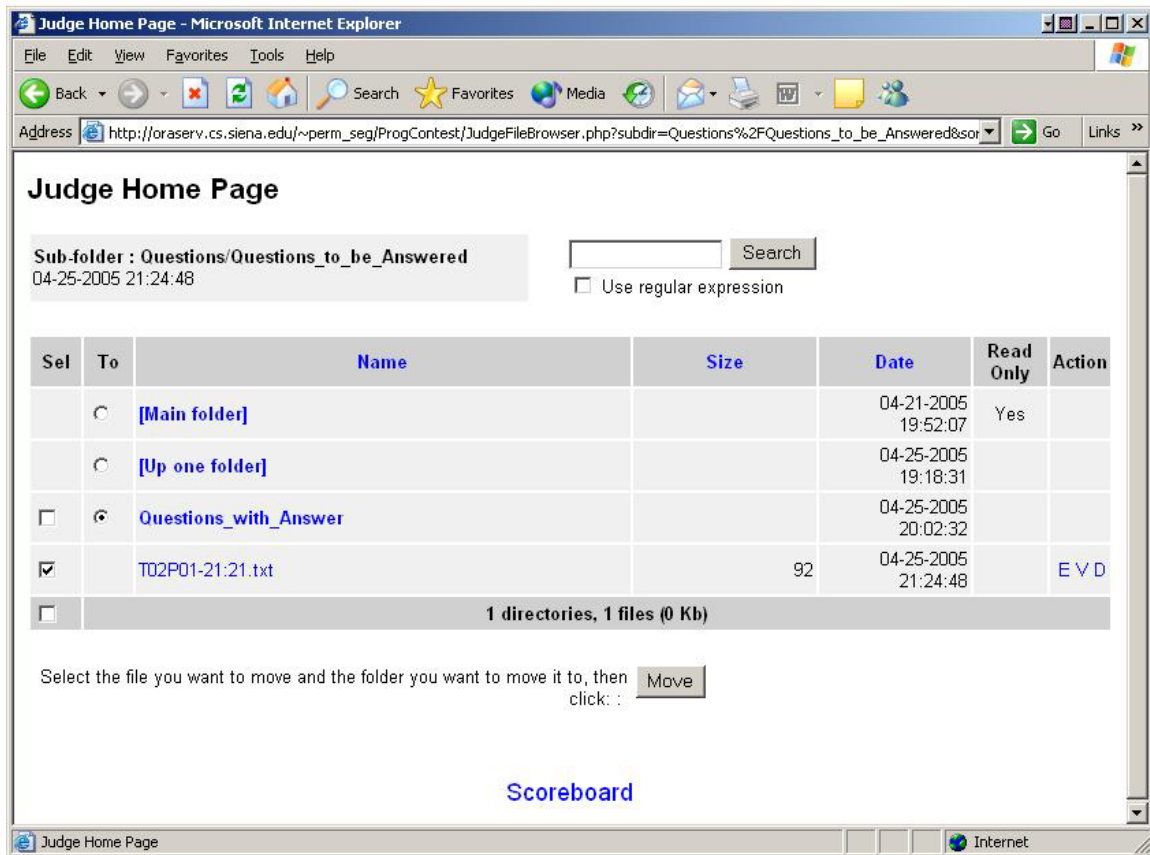
By selecting the "V" on the right hand side of the file, the judge can view the question.

Judges Edit Question

By selecting "E" on the right side of the file, the judge can edit and amend an answer to the team's question.

Judges Move to "Answered" Directory

After the judge saves the amended question, they must move it to the directory "Questions_With_Answer".

Judges Move to Problem Folder

Once in that folder, they must move it to the problem that it corresponds to.

Judges Can View Problem Folder
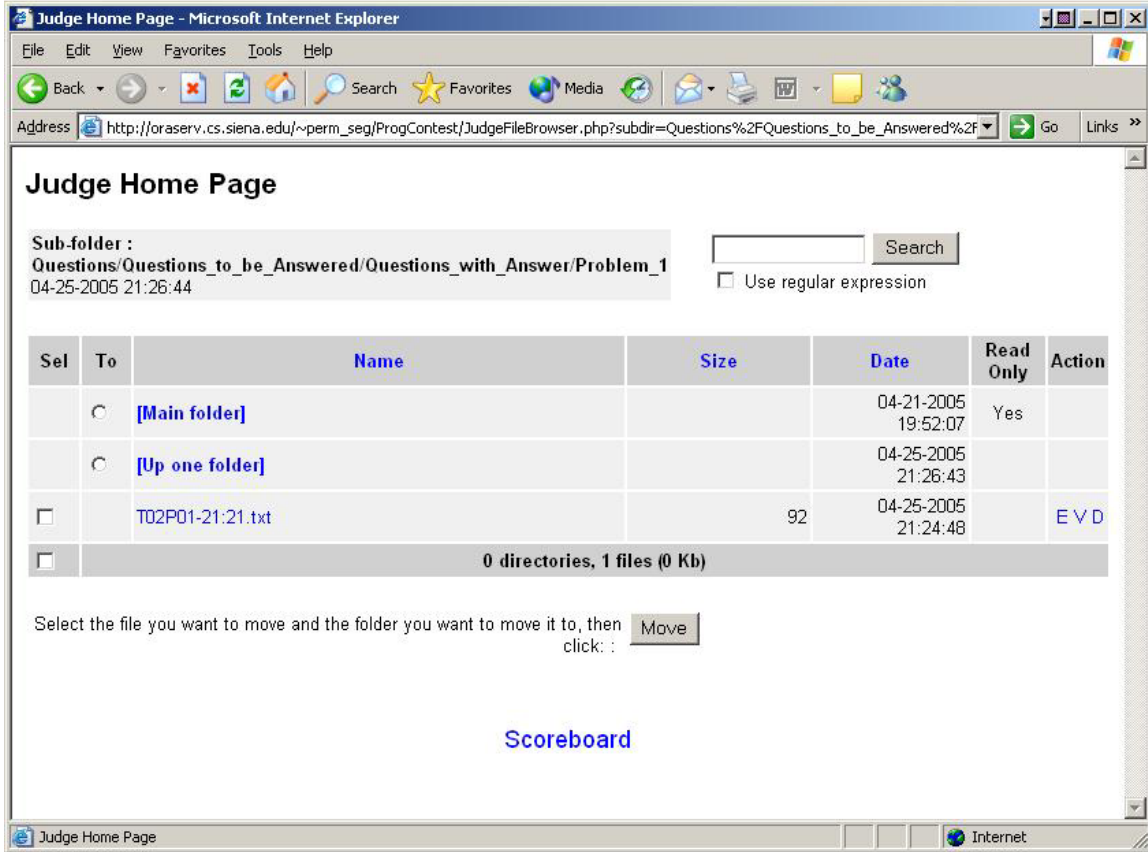
The judge can also view all questions and answers about a single problem by clicking on the link to that problem.

Scoreboard:

Siena College High School Programming Contest Scoreboard

17 March 2004 - 9:00 am - 12:00 pm

(Time Submitted | Elapsed Time)

	1	2	3	4	5	6	7	Correct Solutions	Total Elapsed Time (hours)
Bethlehem									
Shaker									
Guilderland									
Mechanicville									
Niskyuna									
Troy									
Albany									
Schenectady									
Brunswick									

Scoreboard Update 1:

Siena College High School Programming Contest Scoreboard

17 March 2004 - 9:00 am - 12:00 pm

(Time Submitted | Elapsed Time)

	1	2	3	4	5	6	7	Correct Solutions	Total Elapsed Time (hours)
Bethlehem	9:30 | 0:30		10:00 | 1:00					2	1:30
Niskyuna		10:00 | 1:00			11:00 | 2:00			2	3:00
Mechanicville			9:20 | 0:20					1	0:20
Guilderland								0	
Shaker								0	
Troy								0	
Albany								0	
Schenectady								0	
Brunswick								0	

Scoreboard Update 2

Scoreboard - Microsoft Internet Explorer

File   Edit   View   Favorites   Tools   Help

Address C:\SEG Common\Outside Sites\scoreboard2.htm

Siena College High School Programming Contest Scoreboard

17 March 2004 - 9:00 am - 12:00 pm

(Time Submitted | Elapsed Time)

	1	2	3	4	5	6	7	Correct Solutions	Total Elapsed Time (hours)
Bethlehem	9:30	0:30		10:00	1:00					2	1:30
Niskayuna		10:00	1:00			11:00	2:00			2	3:00
Mechanicville			9:20	0:20					1	0:20
Guilderland	11:30	1:30							1	1:30
Shaker								0	
Troy								0	
Albany								0	
Schenectady								0	
Brunswick								0	

Section 7: User Manual

Administrator: The administrator must follow the steps below to setup for the programming contest.

1. Open up the file called "Scoreboard.php". This is where the team names must be entered in the variables, labels, $TeamName##. In this file, the start time of the contest is entered into the variable $StartTime. Also, in this file you can change the size of the scoreboard as needed with a maximum of 25 teams. To do this, you go to the line that contains the team number that you don't need. In front of the HTML code put the following, <!--, then right before the end of the table close tag put the following, -->. Then, for each line of PHP code in between the two comments above, comment out each of the echo statements by putting two forward slashes (//) in front of echo.

2. Now team passwords have to be set by opening "passwords.txt" and entering the password the want each team to use. Then give each password to the corresponding team to log in with.

3. Instruct teams the correct path to save their files to and what name to give each file following our naming convention.
//home/perm_seg/public_html/team##/[filename]

How Teams Login
1. Type the following URL in your web browser and hit Enter:
http://oraserv.cs.siena.edu/~perm_seg/TeamLogin.html
2. The Username is "team" followed by the number of your team. For instance, if your team number is 05, then your Username is team05.
Note: login is case sensitive. Everything has to be in lower case.
3. Your team Password will be provided to you.

Team Side
1. In your team homepage click on "Submit a Solution" and a new page should open with all the problems on it with two groups of radio buttons: one to select a solution to submit to a specific problem and the other to select a language. The team then clicks on submit to send the information to the judges. After clicking on submit the team will be sent back to their home page.
2. In your team homepage click on "Submit a Question" and a new page should open with all the problems with a set of radio buttons to select a question and then a team types in their question into the text box and clicks on submit to send their question to the judges. After clicking on submit the team will be sent back to their home page.

Note: Before submitting a solution, you must first make sure you use the special naming convention for your .cpp or .java file. The naming convention is as follows: Team 1 submits their solution for problem number 5 using C++. They will save their file as T01P05.cpp. If this were a Java file, it would be T01P05.java. These files will be saved on your specified folder on the *Oraserv* server.

3. Teams can see the status of their submitted solution by looking at the status bar on each Team page.
4. Teams can see if their question was answered by the judge by looking at the message board on each team page which is separated by Problem number. Teams click on the Problem number then the file they want to view.

Judge

1. Judges start judging the contest by going to http://oraserv.cs.siena.edu/~perm_seg/JudgeFileBrowser.php
2. Judges then have two directories to choose from either Problems or Questions.
3. If judges click on Problems there is then a sub directory called Submitted and Not Judged where the Teams submitted solutions are waiting to be judged.
4. Judges then click on this directory to see the Teams files that are submitted.
5. The Judges will then have to select the file they want using the checkboxes next to the files and then click the radio button next to Being Judged and then click the Move button on the button of the screen.
   Note: When teams submit their solutions, **two** files will be sent, one with a timestamp and one without. When moving the files along the judging process, both files must be moved together. When viewing, compiling, and running the solution, you must only use the file with the timestamp. The other file is only used for tracking the solutions on the team's status bar and need not be opened.
6. Judges then run the Teams file using a compiler through x-term and determine the team's solution is correct or has errors.
7. The Judge then moves the file to the appropriate directory.
8. Judge can then maneuver back to the main directory of the file browser and click on the Questions Directory.
9. A judge then clicks on Questions to be Answered Directory and then selects a question file to answer by clicking on the E all the way to the right of the file.
10. The teams question file then opens up in a text box for editing where a judge can type their answer to the teams question and then saving it.
11. A judge then moves that file to the Questions with Answer Directory.
12. A judge then moves the answered question file to the appropriate problem number directory

Section 8: Source Listing

See attached source code.

Section 9: Glossary of Terms

Computer Science Programming Contest – A programming contest administered by the Computer Science Department for local high school teams to come to Siena and compete.

DFD – Data Flow Diagram – A graphical representation that depicts information flow and the transformations that occur as data moves from input to output.

Directories – Folders on Turing in which files will be stored, as well as being moved from and to them.

DreamWeaver – A program that will be used to develop all needed websites.

Gantt Chart - A chart that depicts progress in relation to time, often used in planning and tracking a project.

GUI – Graphical User Interface – The screens that users will see while using the Programming Contest Submission Software.

HTML - A markup language used to structure text and multimedia documents and to set up hypertext links between documents, used extensively on the World Wide Web.

Internet - An interconnected system of networks that connects computers around the world via the TCP/IP protocol.

Java- Programming language that our program will possibly be built in.

Linear Sequential Model / Classic Waterfall Model – A systematic, sequential approach to software development that begins at the system level and progresses through analysis, design, coding, testing, and support.

Scoreboard – A web-based list of teams, in ranked order, based on the number of problems each team has answered correctly, and the time used to answer the questions.

Turing – The Unix Server that every PC in the Computer Science Department Connects to. It is under the administration of Mr. Ken Swarner.

Web-based – A software that uses the World Wide Web on the Internet as a user interface

Team ranking – Teams in the contest are placed in an order based on the number of correct solutions and the about of time taken to submit those solutions.

Array - An arrangement of memory elements in one or more planes.

PHP – PHP Hypertext Processor - In an HTML document, PHP script (similar syntax to that of Perl or C) is enclosed within special PHP tags. Because PHP is embedded within tags, the author can jump between HTML and PHP (similar to ASP and Cold Fusion) instead of having to rely on heavy amounts of code to output HTML. And, because PHP is executed on the server, the client cannot view the PHP code.