

Interactive Web-Based Tests With Immediate Auto-Feedback Via E-Mail to the Instructor: Software and Illustration

Arnoldo Muyshondt, Ing-Chang Jong
Sandia National Laboratories / University of Arkansas

Abstract

Web-based tests are increasingly utilized in education. This paper is written to contribute a basic yet versatile software, with illustration, for creating interactive web-based modules for tests with immediate auto-feedback to a specified e-mail address for the instructor. Each module assists students in learning new material by offering “tips,” without providing answers right away, when wrong answers are chosen by students. The software is segmented into five program files, where the first two are subject specific while the other three are general. Being independent of web sites, the three general program files need no changes by users. All software files are freely downloadable from a given web site. Beginning instructors can readily create web-based modules by directly modifying the two subject specific program files, which are included in the paper.

I. Introduction

The **World Wide Web** (WWW) is an environment where the information created is accessible on demand, anytime, and anywhere. It has been used by organizations and individuals for various purposes. In education, it is capable of allowing on-demand access to any information and material for a given course or for tutoring, which an instructor provides at a web site. In particular, the WWW can be used to great advantage in teaching high enrollment courses as well as distance education. Increasing efforts to create web-based educational material on the WWW clearly point to the trend of using web publication as an alternative mode in modern education.¹⁻⁷

This paper is an extension of a recent contribution by the authors. It presents a basic yet versatile software that uses **Practical Extraction and Report Language** (PERL) and **Common Gateway Interface** (CGI) scripting⁸⁻⁹ to manipulate data files to readily create modules for interactive on-line testing with immediate feedback. It allows as well as promotes learning because appropriate “tips” are offered, without providing answers right away, when wrong answers are chosen by students. Note that PERL is a programming language like C or FORTRAN.

A CGI script is simply a program that is run on a web server based on input from a browser. It enables users to interact with the web pages. For example, when a web user fills in a form or a search dialog on the web, the user has used a CGI script. Generally, CGI scripts are basic sets of

variables and mechanisms for passing information from the browser to the server. A CGI script can be used as the **action** to a form or as a direct link. The script may take input from the browser, manipulate it, and display the results on a new browser page. Depending on what the web server supports, CGI scripts can be batch files, compiled programs, or any other executable entity. The scripts may be written in any language as long as the web server supports that language. For example, a CGI script can be written in C as long as the server supporting the web page supports the C language. In our case, the CGI scripts are written in PERL.

The software contributed in this paper is written for any educational professional with a moderate knowledge of how a web site works. The software consists of the following five files:

- (a) *One* gateway file: **test.html**
- (b) *One* data file: **test#.dat**
- (c) *Three* general program files: **testrad.cgi, ans1.cgi, ans2.cgi**

Only the files **test.html** and **test1.dat** are included in this paper for the ease of explanations. For the sake of brevity, the three general program files listed above have *not* been included in this paper. Nevertheless, *all* the software files used in creating web-based tests in this paper are downloadable by clicking the file name [webtest2.zip](#) contained in a listed item on the web page at

<http://comp.uark.edu/~icjong/>.

With this setup, an instructor can directly modify the gateway file **test.html** and the data file **test#.dat** for a given subject and simply employ the general program files with no change. We acknowledge Steven E. Brenner⁸ for using his library routine, **cgi-lib.pl**, which he has granted conditional permission to users, and which is downloadable from the web site:

<http://cgi-lib.stanford.edu/cgi-lib/>.

The aforementioned five contributed files and the library routine **cgi-lib.pl** may be saved in an appropriate directory in the instructor's computer account, where the server supports PERL and CGI scripting. For illustrative purposes, we have saved them in the directory "webtest2" at the web site:

<http://comp.uark.edu/~icjong/webtest2/>.

Naturally, other relevant files may also be saved in this directory. To be sure, an instructor should consult the system administrator of the local server about the protocol for the proper setup for CGI scripts written in PERL. Note that many C commands will work with PERL. A good on-line reference manual for PERL programming can be found at the web site:

<http://www-cgi.cs.cmu.edu/cgi-bin/perl-man>.

II. Interactive Web-Based Tests

Pictures are often invaluable in presenting new ideas to readers. The setup of a gateway page for students and a typical adventure in an interactive web-based test are illustrated in Figs. 1 through 7. Generally, a student uses a browser (e.g., Netscape Navigator™ 4.7 or later) on a computer to sign on to a gateway page for taking web-based tests, which have been set up by an instructor. In the presented illustration, the gateway page is at

<http://comp.uark.edu/~icjong/webtest2/test.html>

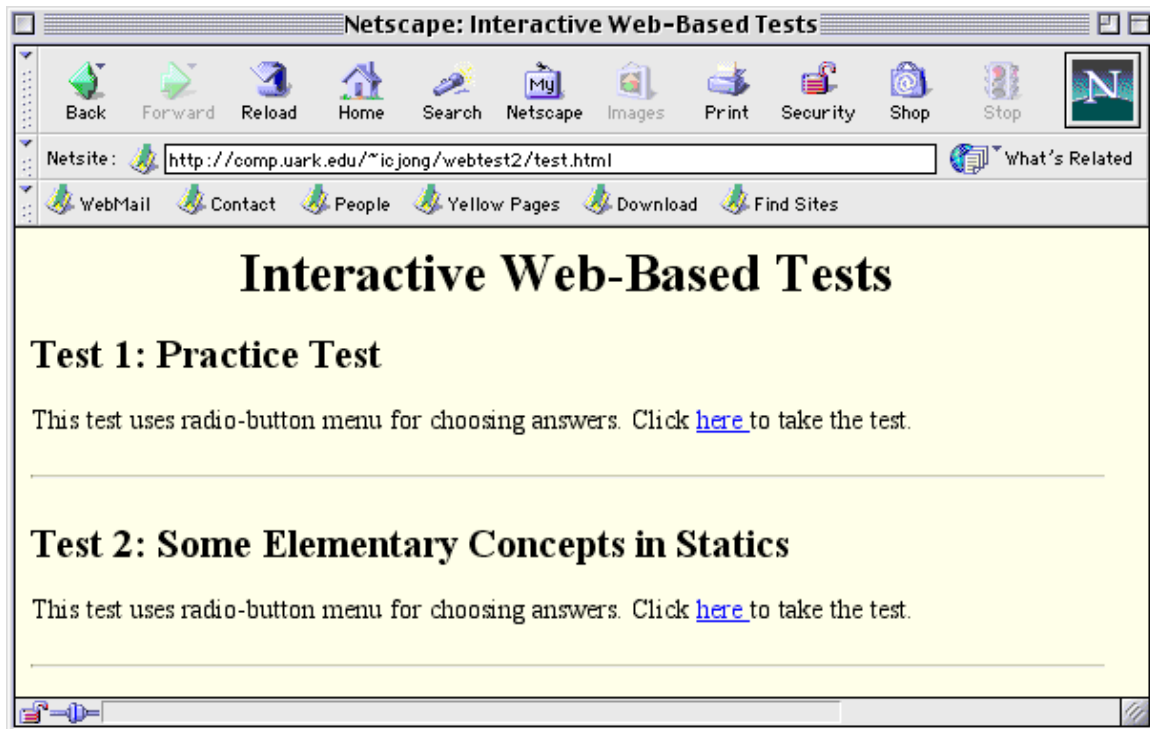


Fig. 1. The gateway page for taking web -based tests

A student may first take a simple practice test to gain familiarity with web-based tests. The tests available to students for testing and learning are listed on the gateway page as illustrated in Fig. 1. Suppose that a student, who wants to learn more about web-based tests, proceeds to click on the word [here](#) under Test 1 on this gateway page. The test created by the instructor will appear as illustrated in Fig. 2.

The captions and notes for Figs. 1 through 7 help one gain an overview of how it works in a web-based test. It is hoped that these figures would make the reading of this paper easy. Because of the confine of space, Test 2, as listed on this gateway page, will not be discussed in the paper.

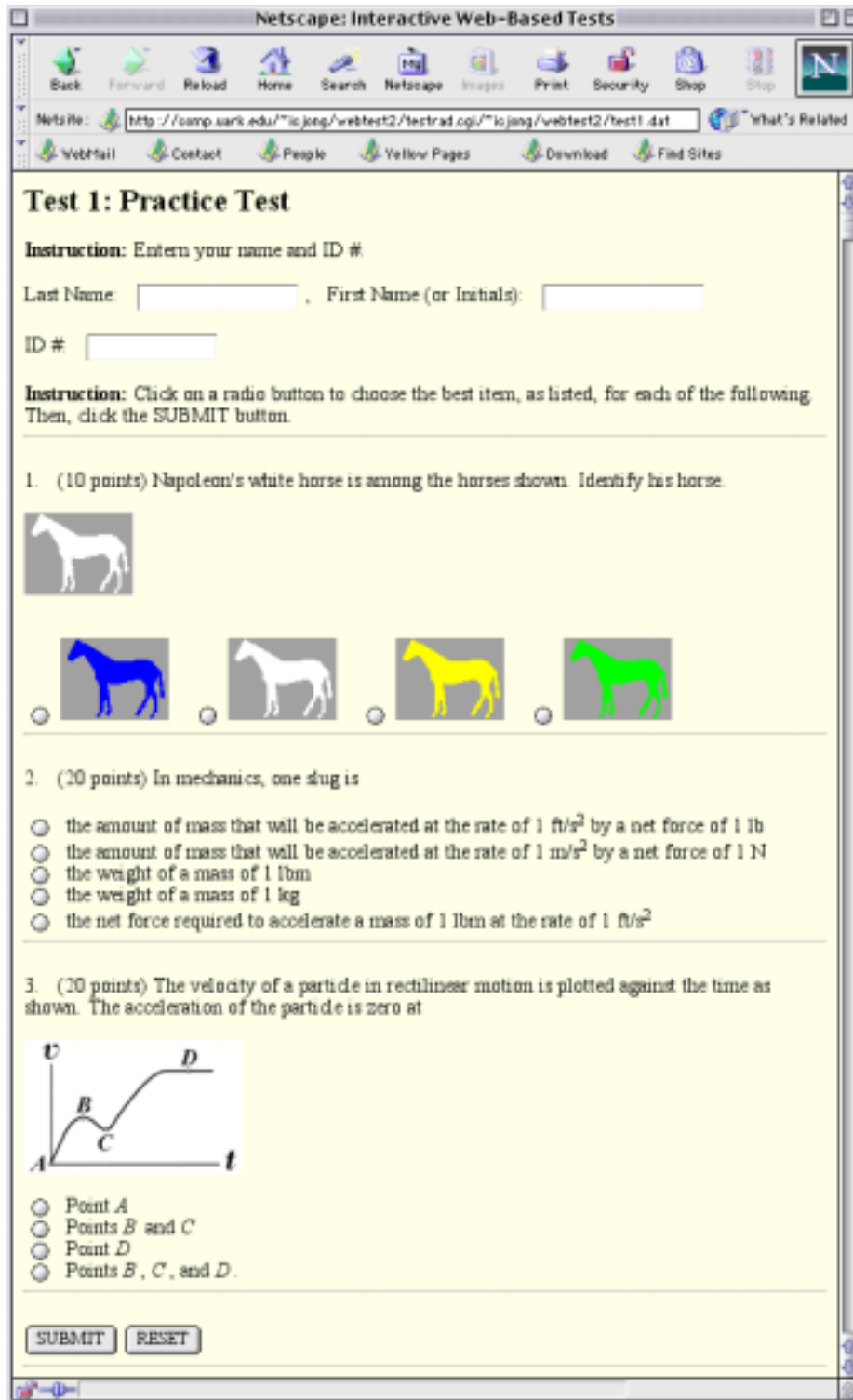


Fig. 2. The test that appears when the word [here](#) under Test 1 on the gateway page in Fig. 1 is clicked

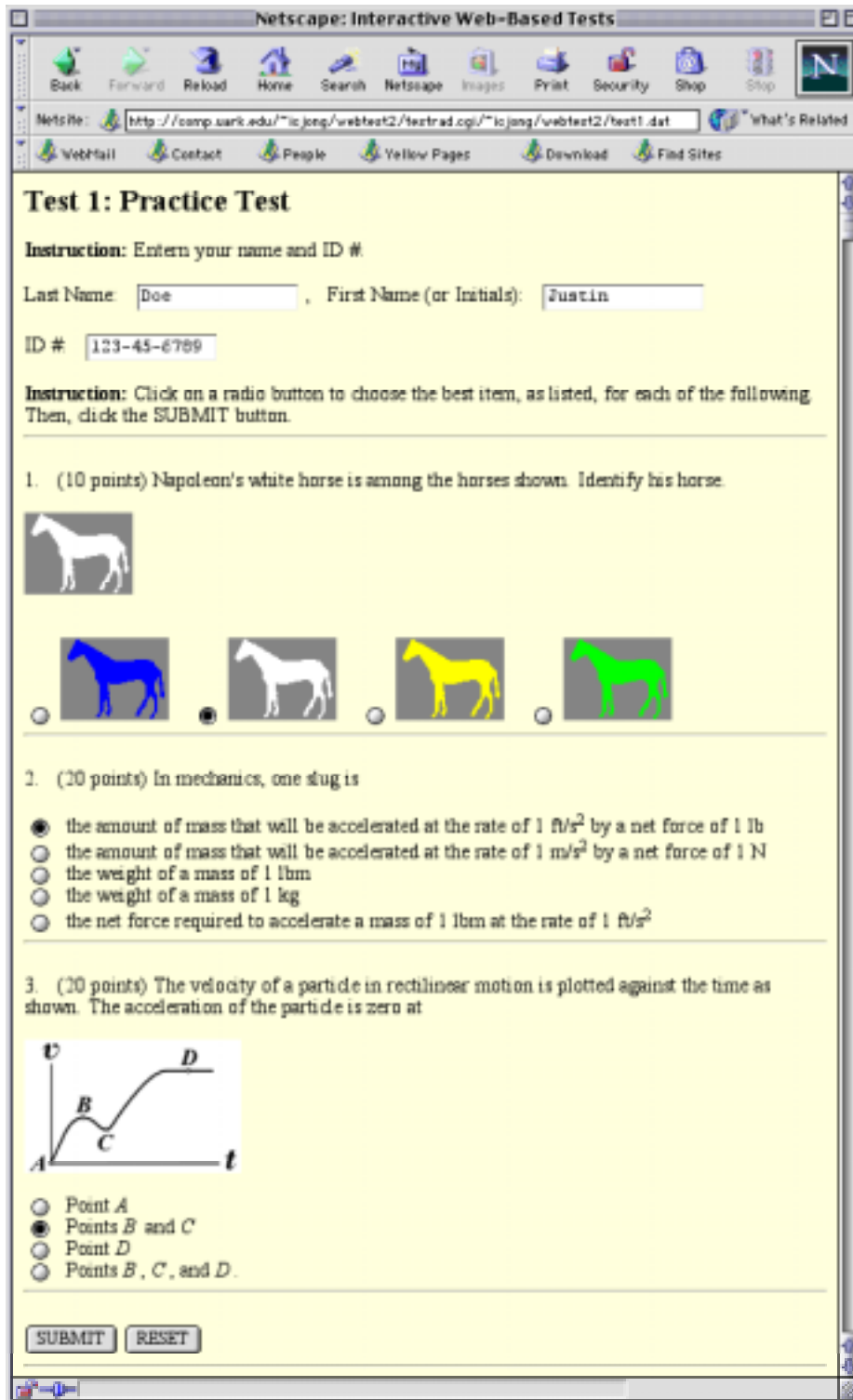


Fig. 3. Black dots clicked onto radio buttons to mark chosen answers on the test by a student who has some difficulty

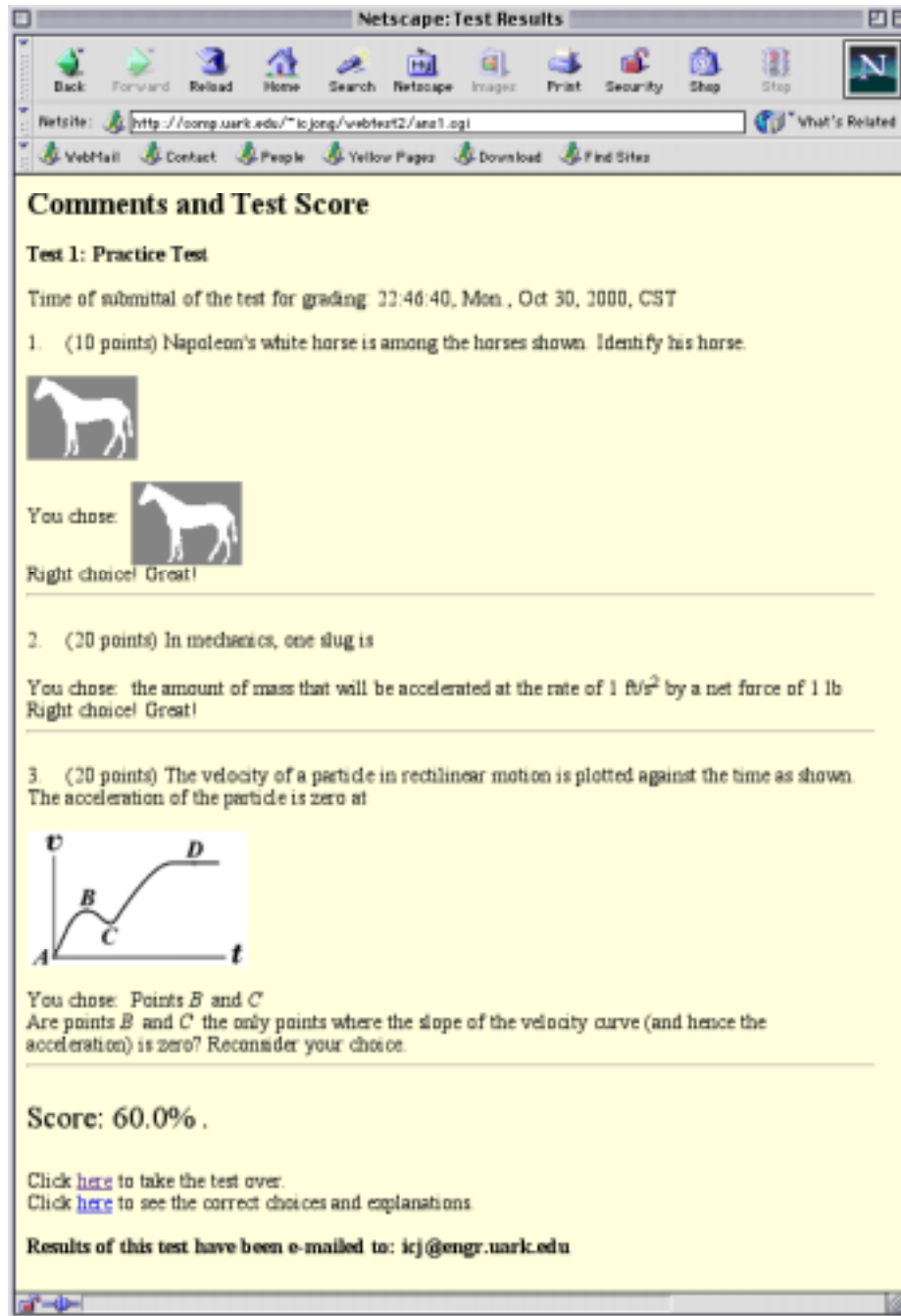


Fig. 4. The immediate feedback to the student, who SUBMITTED the chosen answers shown in Fig. 3 for grading

In Fig. 4, note the following: (a) the test results are automatically e-mailed to the instructor, (b) the time of submittal of the test for grading is recorded, (c) a tip is offered when a wrong choice was made by the student to promote learning, (d) a positive comment for immediate reinforcement is given whenever a correct choice was made.

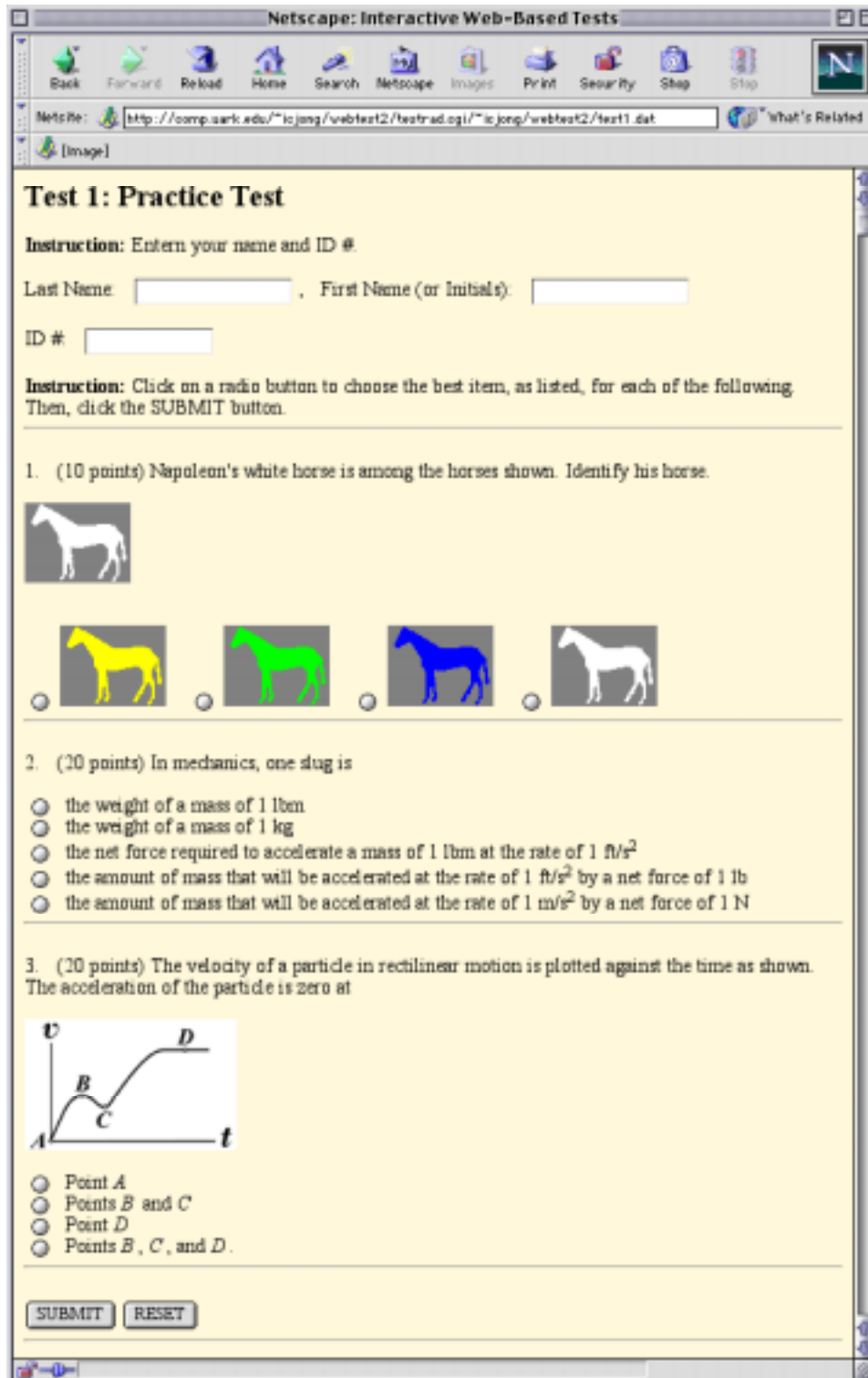


Fig. 5. The test that appears when the student clicks to choose *taking the test over*, where the order of answers is randomized by the software, except for Problem 3 where randomization is purposely turned off

Subject: Web-Test Results for: Doe, Justin
Date: Mon, 30 Oct 2000 22:46:40 -0600 (CST)
From: trapper@mail.uark.edu
To: icj@engr.uark.edu

Time of submittal of the test for grading: 22:46:40, Mon., Oct 30, 2000, CST
Test Title: Test 1: Practice Test

Student Name: Justin Doe
ID #: 123-45-6789

Problems answered incorrectly: 3
Test Score: 60.0%

Fig. 6. The e-mail that is automatically sent to the instructor to report the student's test results shown in Fig. 4

Subject: Web-Test Results for: Doe, Justin
Date: Mon, 30 Oct 2000 22:56:16 -0600 (CST)
From: trapper@mail.uark.edu
To: icj@engr.uark.edu

Time of submittal of the test for grading: 22:56:16, Mon., Oct 30, 2000, CST
Test Title: Test 1: Practice Test

Student Name: Justin Doe
ID #: 123-45-6789

Problems answered incorrectly: none
Test Score: 100.0%

Fig. 7. The e-mail that is automatically sent to the instructor to report the test results of the student who took the test over

Note that no auto-feedback to the instructor will occur if the instructor omits to specify the destination e-mail address for the **e-mail tag** in the data file for the test. This flexibility is desirable in creating sample tests for free review purposes.

III. Gateway File: **test.html**

The file **test.html** is a program to create the gateway page shown in Fig. 1, which the students first see when they sign on to the web site for Interactive Web-Based Tests. It is written in HTML (**H**yper**T**ext **M**arkup **L**anguage).¹⁰ It lists the tests that have been developed and are available to the students. When a test is selected from this gateway page, the browser causes the general program file **testrad.cgi** to be run on the server to process the data file **test#.dat** (e.g., **test1.dat** or **test2.dat**) and to display the test on the computer screen. Since the relevant files are saved in the directory “webtest2” at the site:

<http://comp.uark.edu/~icjong/webtest2/>

the source program of this HTML file, written to display just two tests as shown in Fig. 1, can readily be created and is illustrated as follows:

test.html

```
<HTML><HEAD><TITLE>Interactive Web-Based Tests</TITLE></HEAD>
<BODY LINK="#0000ff" VLINK="#800080" BGCOLOR="#fff8dc">
<H1 ALIGN="CENTER">Interactive Web-Based Tests</H1>

<H2>Test 1: Practice Test</H2>
<P>This test uses radio-button menu for choosing answers. Click <A
HREF="http://comp.uark.edu/~icjong/webtest2/testrad.cgi/~icjong/webtes
t2/test1.dat">here </A>to take the test.</P><HR>

<H2>Test 2: Some Elementary Concepts in Statics</H2>
<P>This test uses radio-button menu for choosing answers. Click <A
HREF="http://comp.uark.edu/~icjong/webtest2/testrad.cgi/~icjong/webtes
t2/test2.dat">here </A>to take the test.</P><HR>

</BODY></HTML>
```

The above HTML file does reflect that both of the files **testrad.cgi** and **test#.dat** are presently saved in the sub-directory [webtest2](#) under the user directory [~icjong](#) in the same server

comp.uark.edu

on the Internet. These directories and files must have “read” and “execute” attributes that allow public access on the WWW. Here, an instructor needs to correspondingly modify only the *names* of these directories and the server in this HTML file. Naturally, this file may be written to display more available tests.

IV. Data File: **test#.dat**

This is the file that provides data for creating an interactive web-based test. For illustration, we make a test, called **Test 1: Practice Test**. It is a multiple-choice test consisting of three different types of questions. Each question has four or five prospective answers. A question may have a figure in the statement of the question, or in the answer, or both. Credit points are not the same for all questions. Additionally, we shall use superscripts for powers and italics for names of points on a curve. Appropriate tips, comments, and explanations, are to be *keyed* to correspond to the various prospective answers. Such an interactive test, illustrated in Figs. 2 through 5, is created using the data file **test1.dat**, which is presented as follows, where the tags (highlighted in boldface) are explained in tables 1 and 2.

test1.dat

title:Test 1: Practice Test
loc:~icjong/webtest2/
e-mail:icj@engr.uark.edu

1

question:(10 points) Napoleon's white horse is among the horses shown. Identify his horse.

picture:tlfigs/horse1.gif

apic-

ture:tlfigs/horse4.gif;tlfigs/horse1.gif;tlfigs/horse2.gif;tlfigs/horse3.gif

answer:pic1;+pic2;pic3;pic4

tip:Napoleon's white horse is white. Is a white horse blue?;Right choice!;Napoleon's white horse is white. Is a white horse yellow?;Napoleon's white horse is white. Is a white horse green?

explanation:Napoleon's white horse is white. Clearly, the white one shown must be Napoleon's white horse.

points:10

&

2

question:(20 points) In mechanics, one slug is

answer:the weight of a mass of 1 lbm;the weight of a mass of 1 kg;the net force required to accelerate a mass of 1 lbm at the rate of 1 ft/s²;+the amount of mass that will be accelerated at the rate of 1 ft/s² by a net force of 1 lb;the amount of mass that will be accelerated at the rate of 1 m/s² by a net force of 1 N

tip:The weight of a mass of 1 lbm on earth is 1 lb. Is slug a unit of force?;The weight of a mass of 1 kg on earth is about 9.81 N. Is slug a unit of force?;Is slug a unit of force?;Right

choice!;The amount of mass that will be accelerated at the rate of 1 m/s² by a net force of 1 N is equal to 1 kg.

explanation:One slug is defined as the amount of mass that will be accelerated at the rate of 1 ft/s² by a net force of 1 lb.

points:20

&

3

question:(20 points) The velocity of a particle in rectilinear motion is plotted against the time as shown. The acceleration of the particle is zero at

picture:tlfigs/fig1.3.gif

answer:Point *A*;Points *B* and *C*;Point *D*;+Points *B*, *C*, and *D*.

tip:At point *A*, the velocity is zero. Is the slope of the velocity curve (and hence the acceleration) equal to zero at this point?;Are points *B* and *C* the only points where the slope of the velocity curve (and hence the acceleration) is zero?;Is points *D* the only point where the slope of the velocity curve (and hence the acceleration) is zero?;Right Choice!

explanation:The acceleration of the particle is zero wherever the slope of the velocity curve is zero. Note that the slopes of the velocity curve at points *B*, *C*, and *D* are all zero.

points:20

random:0

&

Table 1 Tags for Listing Title, Location, and E-Mail Address (*See* the file **test1.dat**.)

Tag Name	Purpose	Requirement	Remarks
title:	To list the <i>title</i> of the inter-active web-based test	Required	The title will appear in a second-level heading (<H2>) at the top of the test page.
loc:	To indicate the <i>location</i> of the sub-directory containing the general program files	Required	The files testrad.cgi, ans1.cgi, and ans2.cgi are located in the sub-directory indicated by it.
e-mail:	To specify e-mail address of the recipient of the feedback	Optional	This tag may be omitted if no feedback is needed or desired.

Table 2 Tags for Creating a Question (*See* the file **test1.dat.**)

Tag Name	Purpose	Requirement	Remarks
question:	To list the non-graphic portion of the statement of the question (<i>See</i> remarks for the tag points: in this table.)	Required	Credit points may be indicated to students in parentheses at the start of the question statement.
qpicture:	To list the location of the picture or graphic item to be displayed as part of the question statement	Optional	Most browsers accept pictures or graphic items in GIF or JPEG format.
apicture:	To list the several locations, separated by semicolons , of the several pictures or graphic items to be displayed as prospective answers	Optional	If this tag is used, the pictures listed under this tag, instead of the items listed under answer: , will be displayed.
answer:	To list the several non-graphic prospective answers, separated by semicolons , where the correct one, or the dummy correct one, must be preceded by a + sign	Required	If apicture: exists, items listed under answer: become dummy ones, but the + sign is still needed to indicate the corresponding correct answer.
tip:	To offer appropriate tips, separated by semicolons , <i>corresponding</i> to the prospective answers that may be chosen by students, without giving them answers	Optional	Appropriate tips will timely help students learn new concepts when they click the SUBMIT button to see Comments and Test Score. (<i>See</i> Fig. 4.)
explanation:	To provide in the key to the test explanations of the reason, rule, law, or principle to be applied to reach the correct choice of answer to the current question	Optional	The explanations for correct choices in the test are given when students choose to click on the word here , two lines below the test score.
points:	To specify <i>internally</i> the credit points to be earned by the student for answering the question correctly (Credit points are indicated to the students under question: .)	Optional	If points: is omitted, the question is assumed to be worth 1 credit point. Thus, this tag is not needed if all questions are worth the same.
random:	To deactivate (with value 0) or activate (with value 1) the randomization of the order of prospective answers	Optional	The default value of the random: tag is 1. Thus, omit this tag if randomization of order is desired.
&	To signal the end of data for a given question in the test	Required	This tag contains no colon. It serves as a delimiter.

V. Program File: **testrad.cgi**

The program file **testrad.cgi** consists of CGI scripts written in PERL. This program, as well as other CGI programs, starts with

```
#!/usr/local/bin/perl
```

as its first line, where the # sign is followed by the exclamation point !. This first line of the program points to the location of the PERL interpreter in the server so that the remaining code in the program can be properly executed. Make sure that a copy of PERL does reside in the sub-directory /usr/local/bin/ of the server. If a line starts with the # sign followed by one or more spaces, it is a *comment* line and will be ignored in the execution of the program.

This program file is first called by the gateway file **test.html** when a student clicks the word [here](#) on the gateway page to select a web-based test, as shown in Fig. 1. The program then reads the data for the test from the file **test#.dat** (e.g., **test1.dat**). It displays the test on the screen and uses radio-buttons to list the prospective answers on the test. The SUBMIT and RESET buttons are provided at the bottom of the test page. Upon completing the choices of answers, the student is to click the SUBMIT button, which causes the program **ans1.cgi** to be executed. This file, **testrad.cgi**, is contained in the downloadable file [webtest2.zip](#), mentioned earlier, and includes ample comments for the reader.

VI. Program File: **ans1.cgi**

The program file **ans1.cgi** is another program consisting of CGI scripts written in PERL. This program is first called into action by the program **testrad.cgi** to grade the submitted test, give tips or comments, and allow students opportunities to take the test over. This program calls the program **ans2.cgi**, which will handle the presentation of correct choices and explanations for the test. This file, **ans1.cgi**, is contained in the downloadable file [webtest2.zip](#), mentioned earlier, and includes ample comments for the reader.

VII. Program File: **ans2.cgi**

The program file **ans2.cgi** is the last contributed program, which consists of CGI scripts written in PERL. This program is called into action by the program **ans1.cgi**. The main task of the program **ans2.cgi** is to handle the display of the correct answers and explanations for the logic behind the correct answers. In other words, this program displays the key to the test when a request to see the correct choices and explanations is made by clicking the word [here](#) at the lower end of the feedback page (*see* Fig. 4). This file, **ans2.cgi**, is contained in the downloadable file [webtest2.zip](#), mentioned earlier, and includes ample comments for the reader.

VIII. Benefits and Issues

An educational software tool, which is practical, versatile, powerful, general, user-friendly, well-documented, and *freely available*, is highly desirable. We thought the software tool that we presented and illustrated in this paper has attributes that match or are close to those just described. That is why we would like to share it with other educational professionals.

A. Benefits

The software presented in this paper can readily be modified to require passwords from users for using the web-base tests. It can be expanded to incorporate audio and video clips. At present, the software presented is, however, intended as a versatile tool to do the following:

- Teach *new* or *remedial* concepts and material in any subject via interactive web-based tests to any number of students, on demand, any time, and anywhere. This is possible once the modules for testing and learning (e.g., **test1.dat**) have been developed by an educational professional and the access to the World Wide Web on the Internet is available. Although the name of the game is “web-based test,” there is actually a lot of *learning* experience to be gained by the students because it offers “tips” to them, without providing answers right away, when they made incorrect choices. Students could take the test again and still get something out of the experience. This feature is not readily matched by the conventional tests!
- To accommodate different *biological clocks* and different *learning styles* of the students. Some students are of the early-bird type, whose most effective hours of learning is in the early part of the day. Some students are of the night-owl type, who have difficulty in getting up early but have productive hours in the night. There are students who find it fun and challenging to interact with a computer and to learn from it at their own pace and in a time of their choice. Since computers are not persons, students are less likely to feel embarrassed to be initially given a low test score. The questions in each retake of a web-based test involve the same concepts to be mastered; thus, there is little uncertainty and much more learning for the students.
- To allow instructors to create teaching material of lasting value and to save overall efforts of teaching any size classes over the long run. A well developed module for interactive web-based test can be improved or enriched over time, and it is there once it is done. Efficient teaching with less effort will be realized over time, and the instructor can use the time saved for other productive scholarly activities.

B. Issues

The issues with web-based tests may likely arise when one tries to use them for replacement of conventional tests. The use of passwords alone could not solve the problem associated with cheating by students. The identity of the student taking a web-based test will require a more chal-

lenging and well thought-out solution. Perhaps, a testing center with proctors, as in distance education, may offer a viable solution for problems associated with cheating.

Any server that supports CGI scripts and PERL can be used to run the software presented in this paper. As the developed modules for web-based tests are kept in an account of a web server, the system administrator will undoubtedly be involved in ensuring security of all the files.

IX. Concluding Remarks

This paper presents a basic yet versatile software that will enable interested beginning instructors to readily create modules for interactive web-based tests on virtually any subject with immediate auto-feedback via e-mail to the instructor. The software is user-friendly and is segmented into five related program files. The three general program files are written using PERL and CGI scripts to manipulate inputs. Sample web pages providing an overview of the interactive on-line testing and learning have been included. The benefits and issues associated with interactive web-based tests have been discussed.

Web publishing on the Internet is fast becoming a powerful tool to provide an alternative mode in education. The use of well-developed web publishing packages will certainly play an important role in education in this new century. For a try out of interactive web-based tests, visit the gateway page at the location: <http://comp.uark.edu/~icjong/webtest2/test.html>.

For quick development of interactive on-line testing and learning on the WWW, one can directly modify the two template files **test.html** and **test1.dat** (which are included in this paper for easy reference), and simply utilize the three general program files **testrad.cgi**, **ans1.cgi**, and **ans2.cgi** (which need no changes and are contained in the downloadable file [webtest2.zip](#)).

To download *all* the software used in creating Test 1 and Test 2, simply click the name of the zipped file [webtest2.zip](#) contained in a listed item on the web page at

<http://comp.uark.edu/~icjong/>

After unzipping [webtest2.zip](#) to a WWW accessible directory in a computer account at a local server, one should correspondingly change the *web addresses* (locations) contained in the files **test.html** and **test#.dat**. Note that each server could have its particular protocol for setting up the software. Therefore, additional assistance can come from the system administrator of the local server.

It is our hope and pleasure to share our easy-to-use software with other educational professionals who are being encouraged to utilize web publication to enhance education. Many colleagues will surely report further innovations of web-based education in the future.

X. References

1. Jong, I. C., and A. Muyshondt, "Simple Scripts to Produce Interactive Web-Based Tests with Immediate Feedback: Software and Illustration," *Journal of Engineering Education*, October 1999, pp. 509-525.
2. Shawki, T. G., "TAM 221 Mechanics of Materials," Department of Theoretical and Applied Mechanics, University of Illinois at Urbana-Champaign, IL, 1995.
<http://e2.tam.uiuc.edu/TAM221/index/>.
3. Kayser, J., and Massetti, M., "Statics Tutorial," Department of Civil Engineering, Lafayette College, Easton, PA, 1995. <http://www.lafayette.edu/kayserj/statics/cover.htm>.
4. Martin, T., "The Development of Interactive World Wide Web Courseware for Students of Engineering and Technology at Deakin University," Faculty of Science and Technology, Deakin University, Victoria, Australia, 1995.
<http://www.scu.edu.au/sponsored/ausweb/ausweb95/papers/education1/martin/>.
5. Kwok, P., Flory, E., and Rencis, J. J., "Bar and Beam Element Learning Modules for Finite Element Method," Mechanical Engineering Department, Worcester Polytechnic Institute, Worcester, MA, 1996. <http://femur.wpi.edu/Learning-Modules/Stress-Analysis/>.
6. Cabell, B., Rencis, J. J., and Alam, J., "Using Java to Develop Interactive Learning Material for the World Wide Web," Session 3268, *1996 ASEE Annual Conference Proceedings*.
7. Alam, J., and Rencis, J. J., "Use of Internet in Information Content Creation and Delivery for Promoting Active Cooperating Learning," Session 2220, *1997 ASEE Annual Conference Proceedings*.
8. Herrmann, E., *Teach Yourself CGI Programming with PERL in a Week*, Sams Net, 1996.
9. Brenner, S., and Aoki, E., *Introduction to CGI/Perl: Getting Started with Web Scripts*, IDG Books, 1995. (This book covers CGI scripting and **cgi-lib.pl**.)
10. Lemay, L., *Teach Yourself Web Publishing with HTML 4 in 14 Days*, Second Professional Reference Edition, Sams Net Publishing, 1997.

ARNOLDO MUYSHONDT

Arnoldo Muyshondt is currently a Principal Member of Technical Staff at the Sandia National Laboratories. He received a BSME in 1986 and an MSME in 1988 from Texas Tech University, and a Ph.D. in Mechanical Engineering in 1995 from Texas A and M University. He worked as a Propulsion Senior Engineer during 1988-92 at General Dynamics, and as an Assistant Professor of Mechanical Engineering during 1996-98 at the University of Arkansas.

ING-CHANG JONG

Ing-Chang Jong is currently a Professor of Mechanical Engineering at the University of Arkansas. He received a BSCE in 1961 from the National Taiwan University, an MSCE in 1963 from South Dakota School of Mines and Technology, and a Ph.D. in Theoretical and Applied Mechanics in 1965 from Northwestern University. He served as Chair of the Mechanics Division, ASEE, in 1996-97. His research interests are in mechanics and engineering education.