

Teen Pregnancy ePrimer: Senior Design Project
by

Mark Hoar

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements
for
the Degree of Bachelor of Science
in the Information Engineering Technology

University of Cincinnati
OMI College of Applied Science

March 2000

Teen Pregnancy ePrimer

by

Mark T. Hoar

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the Requirements
for
the Degree of Bachelor of Science
in Information Engineering Technology

© Copyright 2000 Mark T. Hoar

The author grants to the Information Engineering Technology Program permission to reproduce and distribute copies of this document in whole or in part.

Mark T. Hoar
Author: Mark T. Hoar

3-15-2000
Date

Sam Geonetta
Faculty Project Advisor: Dr. Sam Geonetta

3-13-2000
Date

Kim Myers
Department Head: Dr. Kim Myers

3/14/2000
Date

Acknowledgement

I am grateful for the assistance I received from my advisor, Dr. Geonetta. He was a valuable resource for obtaining the software tools necessary to complete the project, contacting professionals for advice, securing Internet references, and shaping the project's design. Also, his continual affirmations and suggestions kept me personally committed to the topic and motivated to complete the Teen Pregnancy ePrimer as a senior design project.

Also, I wish to acknowledge the constructive criticisms, suggestions, and resources contributed by my wife and peers.

Dedication

You don't have to suffer to be a poet. Adolescence is suffering enough.

-Frank Ciardi

This project is dedicated to Jennifer White and the other health professionals who have chosen to help pregnant teens as their vocation.

This is also dedicated to my family.

List of Figures

Figure 1. Note the browser setting for the behaviors executed by this page.	10
Figure 2. Typical page format and layout.	13
Figure 3. This diet guide resides in the final layer of information.	15
Figure 4. Frame layout for the bulk of the project.	17
Figure 5. The initial cover page after it loads.	19
Figure 6. Note the rollover effect on the “Journals” graphic.	20
Figure 7. The user enters data into this create mailings form.	21
Figure 8. Note the formatted letter, the date object, and the user data.	22
Figure 9. By clicking on the image, a message box appears to the user.	23
Figure 10. Each answer sends an alert message.	24

Table of Contents

Section	Page
Acknowledgement	ii
Dedication	iii
List of Figures	iv
Table of Content	v
Abstract	vii
1. Statement of the Problem	1
2. Literature Review	1
2.1 Teen Pregnancy Literature	1
2.2 User Interface Literature	2
2.3 Programming Specific Documentation	2
2.4 On-line Manuals and Tutorials	2
3. Description of the Solution	3
3.1 User Profile	3
3.1.1 Statistically Speaking	3
3.1.2 User Ability	4
3.1.3 House Keeping (Installation/Administration)	4
3.2 Design Protocols	4
4. Objectives of the Project	5
4.1 Interactivity	5
4.2 Appropriateness	5
5. Design and Development	6
5.1 Project timeline	6
5.2 Budget	7
5.3 Software	7
5.4 Hardware	7
6. Proof of Design	7
6.1 Software Analysis	8
6.1.1 Browser Requirements	8
6.1.2 HTML Features	8
6.1.3 JavaScript Features	9
6.1.4 Visual Basic Contribution	10
6.2 Organizational scheme	11
6.3 Interface Design/Navigation	12
6.3.1 Knowledge of your audience	13
6.3.2 User Expectations	15
6.3.3 Consistency	15

6.3.2 User Expectations	15
6.3.3 Consistency	15
6.3.4 Essential Elements of Interactivity	17
6.4 Interactive Resources	17
6.4.1 Dynamic Effects	18
6.4.2 Print Outs	19
6.4.3 Interactive Tests	21
6.4.4 Interactive Resources	23
7. Conclusions and Recommendations	24
Appendix A	
JavaScript	26
Appendix B	
Visual Basic	35
Appendix C	
Additional Statistics and Charts	36
References	40
Notes	40

Abstract

This document records the information that accompanies the conception, progress, procedures, and conclusions of my senior design project. The logical construct includes a concise statement of the perceived problem, the solution to this problem, the design objectives for this solution, the process for meeting these objectives, the proof of the validity of the process in meeting these objectives, and the conclusions and recommendations.

The Teen Pregnancy ePrimer is a multimedia-classified project submitted to satisfy the Senior Design criteria established by the Information Engineering Technology curriculum. The ePrimer is an experiment in electronic data manipulation in HTML format. Paper-based brochures and pamphlets are integrated from multiple sources to provide user interaction, dynamic effects, and communication tools.

The design features a portable, client-side solution that requires little computer knowledge from the end user. The intended user is a pregnant teen who resides in Hamilton County, Ohio. The user interface simulates a combination of a brochure and an efficient web page; both media are familiar to the user. Interactive forms that create print outs, test the expectant teens, and provide communication tools that extend the functionality of the paper medium.

Overall, the ePrimer meets the objectives specified herein. This completed project serves as a prototype that can readily meet the needs of the health/social industries who are seeking an alternative, interactive, and comprehensive resource to educate pregnant teens.

1. Statement of the Problem

My experience working with pregnant teens has led me to conclude that there is an information void in our community dealing with this issue. Services exist in the healthcare, social, and educational fields; yet, bringing these resources together in a productive manner for teens in crisis presents a challenge. Information technologies provide tools that can facilitate the situation by quickly filtering through community resources for relevancy and aiding teens with an all-in-one reference. Moreover, this project features interactive components such as:

- a create mailings form that gathers information from the user, creates a printable letter, and prompts the user to confirm print out of the letter,
- a create calendar tool based upon a teen's due date that plans the doctor visits and highlights pregnancy milestones,
- a communication element via the Internet that enables the user to chat with other teens at supervised, secure sites,
- a test section that reinforces key information.

2. Literature Review

The literature for this project is categorized in four main groups. The first group contains information about the topic of teen pregnancy. The second group provides information about user interface design. The third set of references includes documentation specific to HTML, Visual Basic, and JavaScript. The fourth group is composed of tutorials and application manuals. Whenever possible, electronic resources were used in lieu of traditional resources. A complete listing of resources is at the end of this document.

2.1 Teen Pregnancy Literature

For information regarding statistical analysis of teen pregnancy, the Allen Guttmacher Institute^A provides relevant information. This data shaped the development

of the content for the project. Many other sources refer to the statistics collected by this institute^B.

Medical, legal, and social information abounds on the Web. Sites that contribute to this project are www.teenpregnancy.org, www.plannedparenthood.org, and www.mayo-health.com^C.

2.2 User Interface Literature

The foundation for this information came from O'Reilly Books', *Information Architecture*^D. A style reference that is quick and specific to view is the "Guide to Web Style"^E located at the www.sun.com/style/ site. Also, several course readings shed light on this topic as well (such as entries from the *Multimedia Producers Bible*^F).

2.3 Programming Specific Documentation

There are definitive sources for this material. For Visual Basic, the online help and MSDN library provide references for VB programmers. Java script references are found in two camps. One is the "Client-side Reference for JavaScript" and "Client-side JavaScript Guide" found at the Netscape Developer's library. The other source is the "Jscript Reference" provided by Microsoft at the MSDN. Finally, HTML specifications are kept at the World Wide Web Consortium (W3C). There is a link to the "Bare bones Guide to HTML" that serves as a quick reference for tags and properties.

2.4 On-line Manuals and Tutorials

The Webmonkey is the master^G. This site contains tutorials for Dreamweaver, JavaScript, HTML, etc., in a pleasing and productive manner. Other tutorials came with the software tools^H that developed the ePrimer. These include Dreamweaver Attain, Dreamweaver 2.0, Fireworks, Paint Shop Pro, Adobe Photoshop, Adobe Illustrator, *et al.*

3. Description of the Solution

The solution to the perceived information void is an electronic resource that provides information to the pregnant teen population. Discussion about the user population precedes a general overview of the design protocol.

3.1 User Profile

The intended user is an audience of pregnant teens who are seeking information regarding health, legal and social aid (statistical analysis of this population is provided). A basic level of competence with Internet browsing is presumed. In particular, familiarity with navigation frames and hyperlinks is essential. However, tasks requiring file system level tasks are considered beyond the ability of the end user. Moreover, the end user may access this resource via a computer that resides in a health clinic setting that requires some basic “housekeeping” of the document.

3.1.1 Statistically Speaking

Factual information on teen pregnancy is obtainable from one definitive site that is often included as a reference for various health sites. The owner of this data is the Allen Guttmacher Institute¹. The basic information states that:

- About 9% of all female teens become pregnant.
- Of that group, over half deliver or go to term with their baby.
- One in three teen moms finish high school.
- Children born to teen mothers face abuse, poverty, health problems, and school difficulty at a greater rate than children born to older moms.
- Teen pregnancy is a high-risk health problem.

This information guided the content of the document. (See Appendix C for additional charts and statistics.)

3.1.2 User Ability

Beyond the presumption that the user has some background with typical Web interfaces, the ePrimer facilitates file system tasks for the end user. Examples of these tasks include navigating through a menu item in order to print, to open/close a local file, to open/close an application, or to set preferences for a browser application. Many of these tasks are designed to automatically prompt the user for completion.

3.1.3 Housekeeping (Installation/Administration)

The ePrimer exists as a standalone application. Therefore, the program requires ownership for housing the CD-ROM, PC host, connected printer, and appropriate operating system software. The end user is dependent upon the owner maintaining available resources in order to execute the applications necessary to view this document. The “Tools” section is dependent upon an on-line printer (either local or networked) and an Internet connection. In addition, fully functioning browsers (for instance, JavaScript 1.2 must be enabled) and a Winxx OS must be present.

The health clinic is the typical owner profile^J. These facilities, in general, have some access to computing resources such as, a PC with Winxx operating system (and subsequently bundled web browser), a printer, and Internet access.

3.2 Design Protocols

In general, the design protocols are adapted to address the aforementioned problem statement and subsequent objectives. Specific details are discussed in the “Proof of Design” section of this document as to how these protocols are implemented. The table that follows gives an overview of the design protocols:

Feature	Description
HTML file format	HTML is the chosen format for the document because it provides a platform that is portable, script enabled, and event responsive.
Scripts	These are client-side Java Scripts that give the document interactivity.
User Interface	These include: 800X600 display 256 colors (non-dithering) Serif font Global Navigation Local Navigation Browser "look and feel"

4. Objectives of the Project

The objectives of this project are to address the problem as described previously. In short, this design provides an electronic document that deals with a specific health care issue. This document should be interactive and appropriate for the intended audience.

4.1 Interactivity

The ePrimer contains tools that dynamically present information based upon the user's response in an event driven manner. For instance, information is displayed after a user clicks on a hyper-link. Another interactive element is communication tools. These include a letter generator and Internet connection to health professionals and peer chats. Also, there is some testing to insure and reinforce learning.

4.2 Appropriateness

The information contained in the ePrimer conforms to reasonable user interaction factors. These include appropriate format, graphics, text, content, and platform. Knowledge of the audience and UI guidelines drive this goal.

5. Design and Development

The material in this section includes timelines for this project, budget considerations, software and computing resources.

5.1 Project timeline

Co-op Term 1

Learn Tools for Authoring Project
Add to References

Senior Design 2

Week 1
Meet with advisor for realignment or other concerns
Week 3
Review interface design
Week 5
Gather preliminary links, demo link page, build storyboard
Week 7
Review potential graphics
Week 10
Demonstrate overall grasp of project/presentation of prototype

Co-op term 2

Further tool development
Further references and research

Senior Design 3

Week 2
Code scripts and forms
Week 5
Prototype of presentation
Preliminary documentation
Week 8
Submit final project
Submit final documentation
Week 9
Project presentation

5.2 Budget

There are no budget items for this project. All software support is provided through the department.

5.3 Software

The following software tools contribute to the development of this project:

- Dreamweaver 2.0
- Dreamweaver Attain 1.2a
- Macromedia Fireworks
- Adobe Photoshop
- Adobe Illustrator
- Paint Shop Pro 6.0
- Microsoft Front Page 2000
- Microsoft Notepad
- Microsoft Visual Basic 6.0
- Internet Explorer 5.0
- Netscape Communicator 4.5

5.4 Hardware

The hardware includes typical PC components:

- Intel CPU with MMX
- CD-ROM RW
- 64 plus MB RAM
- Zip Storage Media
- 14/15 inch Display (800X600 res.)

6. Proof of Design

This product provides a packet of information for the user in an electronic format.

The format for the data is HTML instead of an executable (.exe), Word© document (.doc), or an Adobe© document (.pdf). The project requires a browser to interpret the data and to add interactivity via JavaScript programming.

This project discusses a health-related topic—teen pregnancy. The user gathers information about the various health, legal, and social services in the Greater Cincinnati

geographical area. In particular, the teen views information about the birthing process, pre- and postnatal care, parenting skills, drug references, and child development. The document is named the “Teen Pregnancy ePrimer.”

6.1 Software Analysis

This section moves beyond the previous discussion regarding intended design protocols and into the design protocols *in praxis*. Given the objectives to provide interactive and portability for this project, the implementation of the software design entailed browser functionality, Visual Basic, JavaScript and HTML development.

6.1.1 Browser Requirements

The design specification for this project is a fourth generation browser, either Netscape Navigator or Microsoft Internet Explorer. All HTML and JavaScript objects and methods are designed to be compatible. However, due to audience ability constraints (see audience section below), an autorun.inf file opens a managed instance of Internet Explorer by default. This allows the ePrimer to manage the initial window’s properties. The window is 800x600 with no menu bar or status bar, the position is centered on the display, and the icon and form caption show the ePrimer’s title and CD icon.

6.1.2 HTML Features

The HTML format of this document enables portability and functionality for this product. The required application needed to view the ePrimer is an HTTP browser such as Internet Explorer (4.0 or higher) or Netscape Navigator (4.0 or higher). The HTML standard is HTML 4.0 as is defined by W3C.^K

6.1.3 JavaScript Features

The JavaScript used in this project extends user interactivity and HTML functionality to the document. The scripts come from several sources. They are generated from Dreamweaver or Attain software packages (called “behaviors”), adapted from preexisting scripts on the Web, or developed from tutorials or references. The scripts are designed for 4.0 generation browsers. The software products are set for 4.0 browsers (see Figure 1). The scripts from scratch (see Appendix A) use objects and methods that exist in both the Jscript Developer Network reference and the Netscape JavaScript 1.2 reference guide.

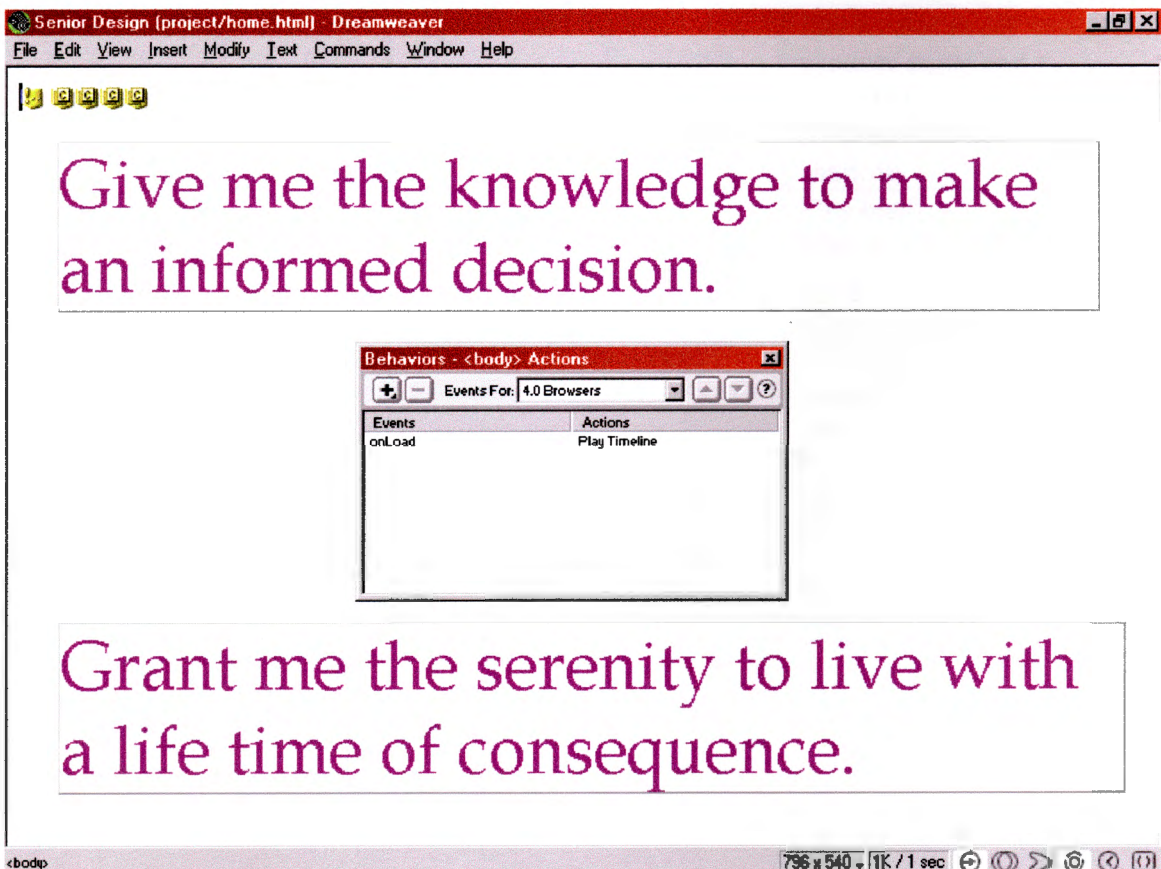


Figure 1. Note the browser setting for the behaviors executed by this page.

In general, the scripts fall into a few categories. One type of script aids document navigation. Another applies a “behavior” to the web page. Macromedia software programs typically generate these. Another type manages document layout. Finally, another type provides general programming assistance. See the table that follows for more detail.

Script Object [Method]	Category	Function
Location	Navigation	This object updates two frames at once.
Window.open(URL, handle, features)	Layout	This method opens a new window. The features include window height and width.
Window.close() or close('handle')	Layout	This method closes a window.
Var winvar	Layout	This declares a variable.
Winvar.moveTo()	Layout	The winvar is any assigned window name and a moveTo() method which positions the newly created window.
Winvar.focus()	Layout	This method brings the Winvar window to the foreground.
Winvar.document.write()	Layout	This method writes HTML or text into another window called Winvar. It could also write to another frame.
Str	General programming	This creates a string object.
Stringvar.Slice()	General programming	This slice() method slices apart a defined string value.
Date()	General programming	This object returns a date and time value.

6.1.4 Visual Basic Contribution

Visual Basic contributes to the initial management of the browser environment.

The primary instance of an Internet Explorer form appears as the housing component for the document. This is desired in order to create an application that opens to a specified

URL. This executable will open the [home.html](#) page that starts the ePrimer regardless of the host PC's configuration. The result is that an autorun.inf file can open the project without relying on the end user to navigate file system commands. In addition, the end user does not have to make any special GUI changes to the Internet Explorer browser. (See Appendix B for more details.)

6.2 Organizational Scheme:

The document is divided into six major topics, a splash page, a cover page, and a "help" section. These are reflected at the bottom of each page in the "tab" frame. The table that follows describes each of these:

Section Name	Description
Splash Page	This page adds a dynamic element to the document at the beginning. A marquee message defines the theme of the project.
Cover Page	This page serves as a functional cover and introduction to the document.
Home	This page welcomes the user to the document and provides help to the user.
Help	This section is always available via the help icon once the user has entered the document. The information contained herein setups environmental elements in order for the document to execute properly and navigation aid for the end user.
Medical	This section provides several medical pages of interest for the end user such as, health risk concerns, biological information, nutritional guidelines, et al.
Journal	This section features two slide shows and a medical journal that of other teens who are sharing their stories with the end user.
Social	This section provides social information to the end user regarding WIC and CHIP assistance.
Legal	This section displays legal information regarding abortion and adoption.
Tools	This section provides resources for the end user such as, a letter generator, a personal calendar, access to on-line chats and web sites, and interactive tests.

6.3 Interface Design/Navigation

There are many considerations for the user interface(UI) of the ePrimer.

However, some simple rules have influenced the overall construction of the UI. They are:

- Know your audience.
- Give the audience what they expect.
- Be consistent.
- Provide essential elements for interaction.

Figure 2 provides an illustration of how these goals are achieved.

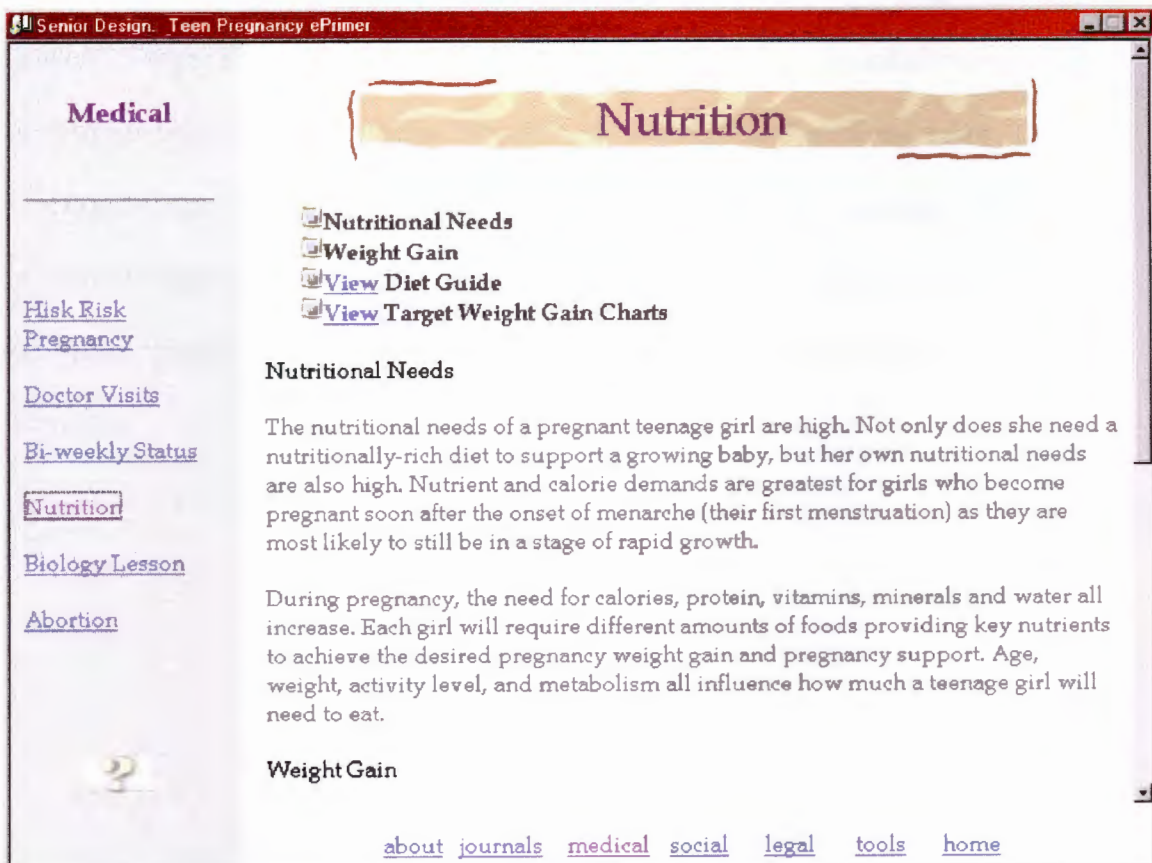


Figure 2. Typical page format and layout

6.3.1 Knowledge of your audience

My knowledge of teen-agers has shaped the design of the UI. Some information about the audience is listed above. In addition, I relied on my experience working with high school students to make generalizations about the group.

The information is distributed in incremental levels. Therefore, the user can go as deep with a topics as they wish. At the top of the page is a banner that gives the heading for the page; in this case, the banner is “Nutrition”. Below this a list of topics that the page covers is displayed; this page talks about “Nutritional Needs,” “Weight Gain,” “Diet Guide,” “Target Weight Gain Charts.” Next, these topics are unpacked more in the body of the page. For “Nutritional Needs” and “Weight Gain” sections, more information is provided. If additional information is needed, there are links that launch an external window. This provides the final layer of information for the user; clicking on the “View” hyperlink next to the “Diet Guide” section launches the following window (Figure 3):

Diet Guidelines - Microsoft Internet Explorer

Pregnancy Diet Guide

Nutrient Rich Food Group	Servings Needed	What equals a serving
Milk & High Calcium Foods	Choose 4 - 5 servings per day	<ul style="list-style-type: none"> • 1 cup milk or yogurt • 2 cups cottage cheese • 1 1/2 oz cheese • 1 cup fortified soy beverage • 1 1/2 cups ice cream • 1 cup calcium-fortified fruit juice
Protein Foods	Choose 2 - 3 servings per day	<ul style="list-style-type: none"> • 3 oz cooked meat, fish or poultry • 2 eggs • 1 cup cooked beans • 4 tablespoons peanut butter
Breads and Grains (whole grains are best)	Choose 6 or more per day	<ul style="list-style-type: none"> • 1 slice bread (1 oz) • 1 small tortilla • 1/2 cup cooked cereal • 3/4 - 1 cup cold cereal • 1/2 cup cooked pasta • 1/3 cup cooked rice • 1/2 English muffin • 1/2 small bagel
		<ul style="list-style-type: none"> • 1 cup raw fruit or vegetables • 1/2 cup cooked vegetable

Figure 3. This diet guide resides in the final layer of information.

Using this incremental scheme, the typical teen-ager seeking information can quickly skim the topics in order to find a subject in which they would like to explore more deeply.

The content of the project is directly related to the end users' needs. The topic headings reflect an overall impression of the materials that this group requires. For instance, this nutritional section contains a relatively large amount of information because of the problem and frequency of the problem that teen mothers have during pregnancy

with their diet. The abortion and adoption sections are comparatively lessened because most pregnant teen keeps their babies.

Another concern specific to this audience of teens that shaped the UI is that this population of teens is under duress. The colors of the text are gradients between black and gray. This softens the contrast. Along with the purple (#663366) and brown shades, these colors are intended to soothe the audience.

6.3.2 User Expectations

Teens today are Internet savvy. The project's HTML format gives the user a familiar UI in which to interact. Items such as links, scroll bars, and frames are conventional devices on the web; therefore, the events that occur in the project are intended not to surprise the user.

Also, the majority of information that teens receive is in brochure format. The layout of each page with the content frame to the side simulates a brochure. Therefore, the ePrimer is inline with other information that teens view.

6.3.3 Consistency

Important to staying consistent with other materials teens view is the function of being consistent with the UI throughout the project. Each page has the look of the Nutrition page (Figure 2.): the text color, size, font, the banner always appears, the links have a uniform color, and extra windows have a purple background. Three frames manage the overall appearance. The layout is illustrated below (Figure 4):

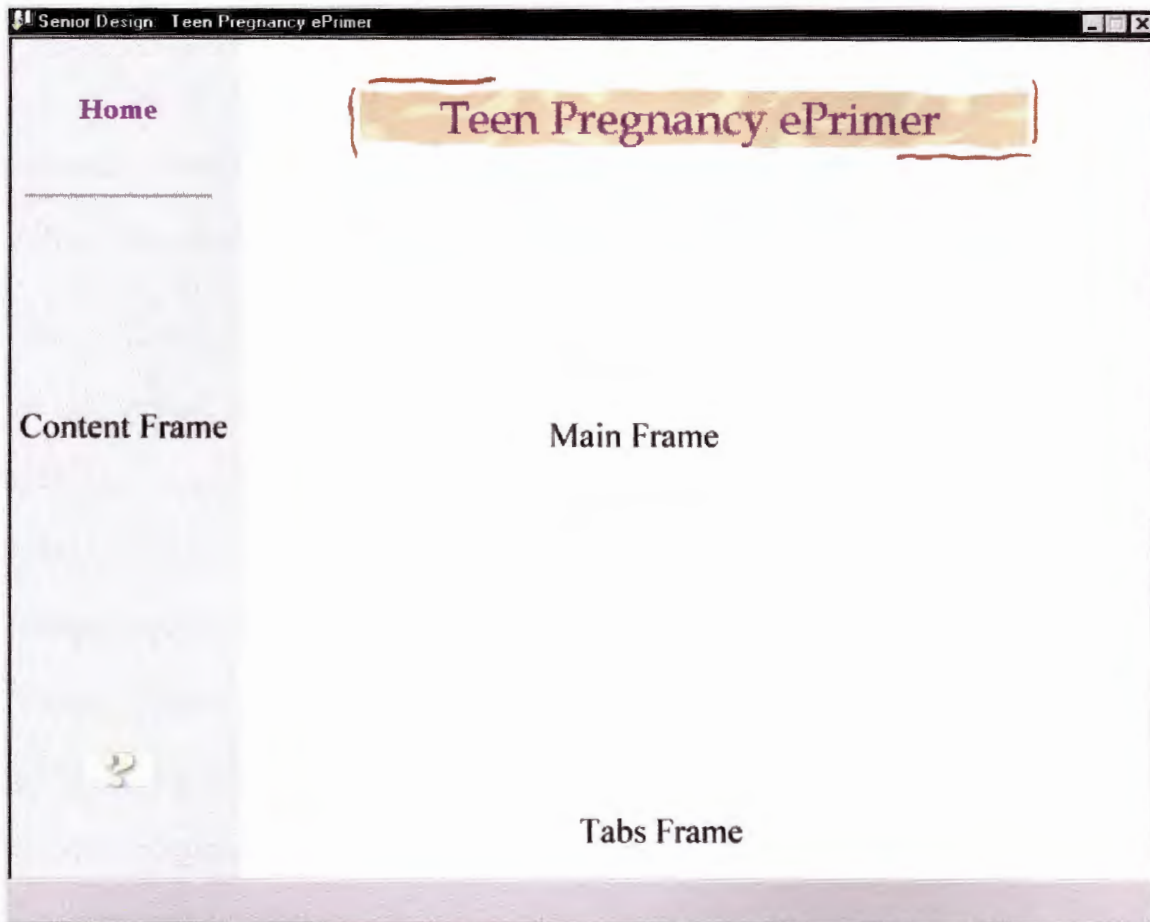
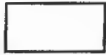


Figure 4. Frame layout for the bulk of the project.

The global links located in the “Tabs Frame” take the user to the home page for each section. The local links always affect local objects subordinate to the global topic. The help icon is consistently located in the same location of the “Content Frame”. The majority of content is displayed in the “Main Frame.” Any additional information (such as the “Diet Guide” (Figure 3.)) launches a new instance of a window. These pages are to be discarded after the end user retrieves the information.

6.3.4 Essential Elements of Interactivity

Web users are conditioned to seek common devices in order to interact with the web page. Keeping this in mind, the project contains essential elements of interactivity such as, help, global navigation, location, and easy readability. The help is accessible by the  icon. This gives the user the familiar question mark sign in order to launch the help section. The global navigation at the bottom of the page is always present. The banner and content page header combine to specify the location for the user. The banner displays the local topic, while the content header shows the main topic heading. In the example page, this is the “Nutrition” section contained within the “Medical” topic. Finally, the text is a serif style font sized appropriately for young eyes (12 point). The specific choice is “Book Antiqua” which is more informal and fun^L than Times New Roman. Links appear as text as well. However, these are consistently displayed in color and underlined as follows:

Link	This is the a general link
Link	This color denotes a link that is active.
Link	This color link signifies a previously visited link.
Link	This is the default link on a purple background.
Link	This is the active link on a purple background.

6.4 Interactive Resources

The ePrimer extends interactive resources to the end user that move this electronic packet beyond the typical paper source. The previously mentioned HTML format and scripts create dynamic effects on the page and construct print outs for the teen. Also, with the help of Dreamweaver Attain Knowledge Objects, the project includes tests that

reinforce the major themes of the ePrimer. Another interactive element is the collection of Internet resources that provide communication via secure chat rooms.

6.4.1 Dynamic Effects

The ePrimer touts visual effects that occur as a page loads or when a user interacts with a page object (such as a target link object). The initial “splash” screen is a timed procedure that scrolls a message on the page and hide/shows the object in turn^M. The second sequential page is the cover page. This page contains scroll over events that hide/show a layer of information and a simple rollover that casts a shadow over the selected text graphic. View Figures 5 and 6 to see the how the mouseover event effects the initial page.



Figure 5. The initial cover page after it loads.



Figure 6. Note the rollover effect on the “Journals” graphic and the shown layer of text to the lower left of the picture.

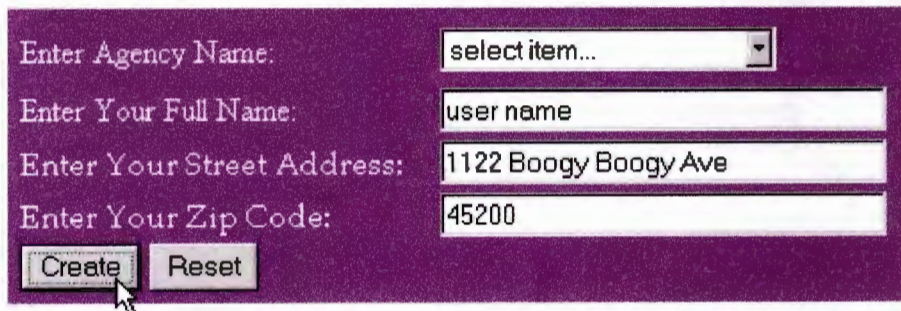
6.4.2 Print Outs

The ePrimer creates two types of printouts for the teen. One type is a generic letter requesting help from a specific agency. The user selects the agency from a list and enters some address information through a form (Figure 7). This material is then parsed out into a letter format and the user is asked to confirm the print action (Figure 8). In addition, the script adds the agency’s address and the date to the letter without input from the teen.

The other printout is a personal calendar^N. This also asks the user to input information via a form and returns a table of information with the option to print.

Create Mailing Form

1. Complete the following form.
2. When finished, click the "Create" button.
3. To print the letter, click OK on the confirmation window.



Enter Agency Name:	<input type="text" value="select item..."/>
Enter Your Full Name:	<input type="text" value="user name"/>
Enter Your Street Address:	<input type="text" value="1122 Boogy Boogy Ave"/>
Enter Your Zip Code:	<input type="text" value="45200"/>

Figure 7. The user enters data into this create mailings form.

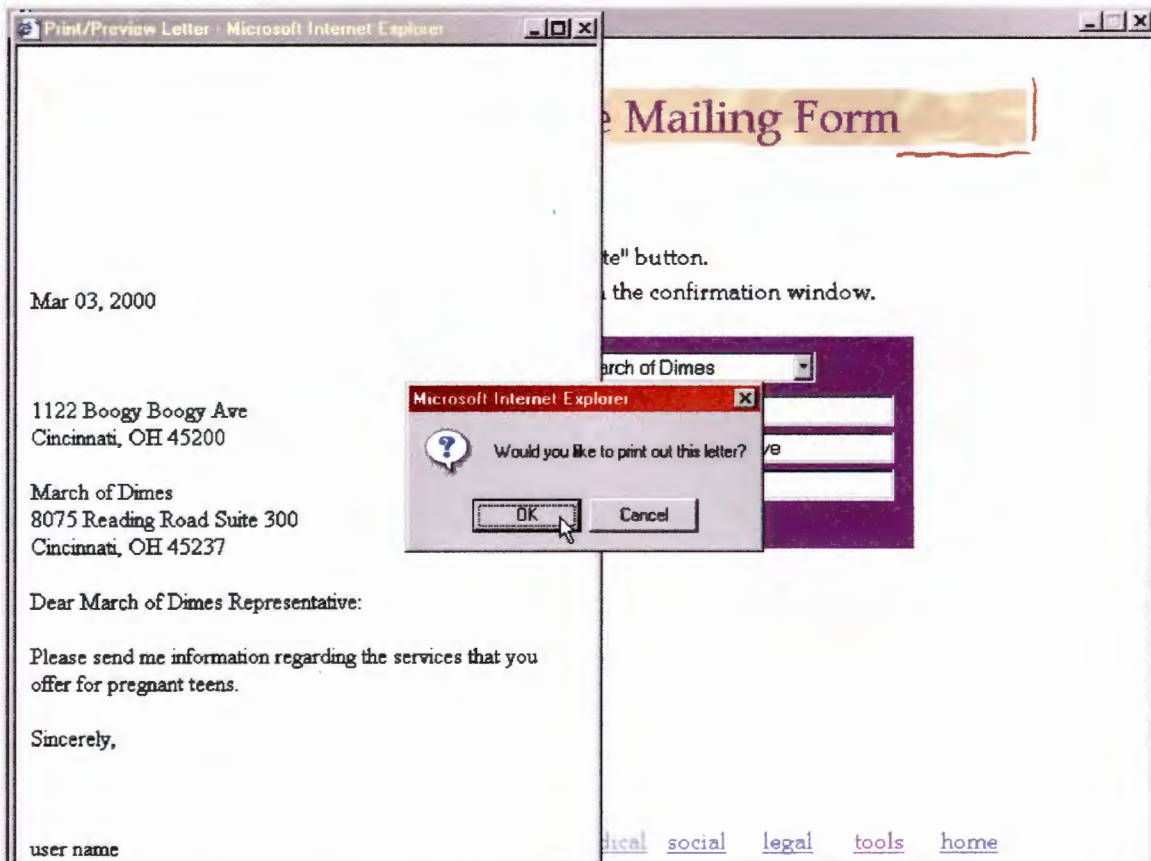


Figure 8. Note the formatted letter, the date object, and the user data.

6.4.3 Interactive Tests

These test questions reinforce learning for the end user. They are didactic in nature and are not created in order to judge the individual's performance. Dreamweaver Attain provides authoring tools that create test objects using JavaScript. This project features two instances of these objects. One simulates an image map that sends an alert message when the onclick event happens (Figure 9). The other group interacts via option buttons on a form that return messages designed to encourage or affirm the tester (Figure 10).

Ultra Sound Tester

- ☐ Make a guess as to what baby part is in the box
- ☐ Next, click on the box to view the answer.



Figure 9. By clicking on the image, a message box appears to the user.

Self Test

Answer any or all of the following questions. There is no grade, so do not be afraid to try.

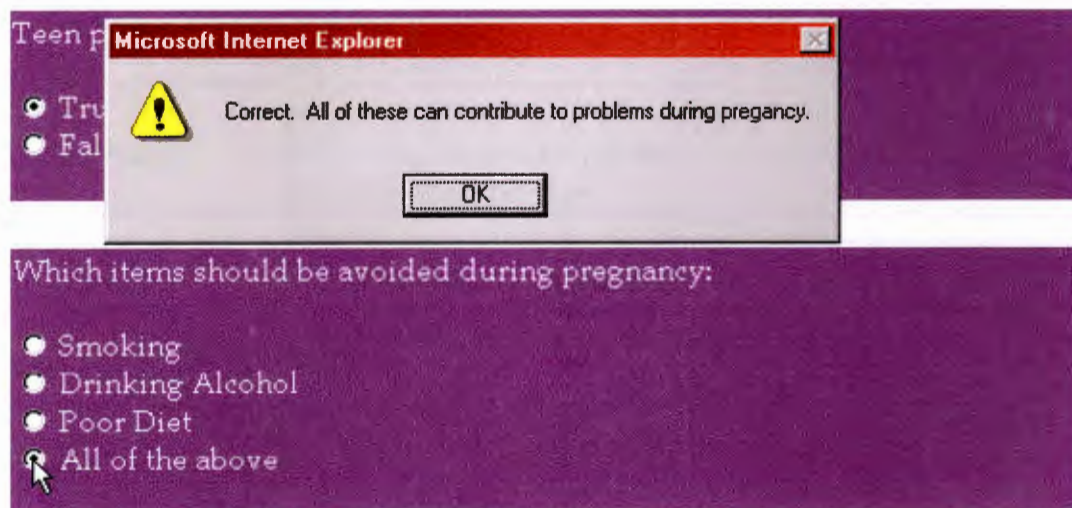


Figure 10. Each answer sends an alert message. The message extends the answer.

6.4.4 Interactive Resources

The project also includes a way for teens to gain more information or chat with other teens and health professionals using a select list of links. The list is far from comprehensive^o. Too many links would be too much information for the intended audience. However, the teen launches a new window in order to visit an Internet site.

The resources are:

Baby Center	This site contains several resources for expecting moms. Check out the Baby Namer Tool. Sign up for free baby products.
Teen Pregnancy	This site is devoted to helping pregnant teens.
Teen Wire	This site connects teens from all over together to discuss tough sex-related issues. Participate in a secure online chat or post questions to health professionals.
Yahoo! Health	This site contains links to several resources for expecting moms and teens. Contact health professionals on-line.
i wanna know	Similar to Teen Wire. More on-line, secure chat with other teens and professionals.
Pregnancy Helpline	This site represents the National Helpline for pregnant teens. Find valuable resources here for obtaining help.

7. Conclusions and Recommendations

The Teen Pregnancy ePrimer meets the objectives as defined above in the “Description of the Solution.” The project consists of an electronic resource, interactive in function, that is user specific.

The document serves merely as a prototype in which the development can now go forward. The following are suggestions for subsequent steps to the Senior Design project:

- Actual user interaction has not been tested.
- The content needs to be clearly written at the cognitive level of school-disoriented teen-agers.
- The scripts could include more error checking and debugging.

Another major hurdle for developing the project is copyright concerns. Many of the photos and content of the project is taken from previously published and copyrighted materials. No permissions for these materials exist.

In addition, the ePrimer is designed to be extensible as a web site. In order for this to happen, a business partner must be willing to sponsor the project. This means they could fund and maintain the web site. Also, they would be responsible for seeking materials in which they own or have permission to certain copyrights.

Appendix A

This is the JavaScript code I wrote for the Create Mailings Form and the Personal Calendar Form. Note that in the latter script I only copied below my contribution to a pre-existing script located at the Mayo Clinic Health Site.

Create Mailings Form script:

This script resides in two locations. Most of the script is on the mailjs.htm page and an onLoad function resides on the target window "letter.htm." The latter portion is called whenever the letter gets loaded. Herein lies a bug. I am unaware of a way to manage this process that takes place by the browser. By refreshing the document or by repeatedly recreating the form, one can overcome this problem. However, this is not an acceptable practice for future releases. The code is listed below:

This part resides on the mailjs.htm page:

```
// This function checks to see if the data was entered and seeks the correct address for each agency
```

```
function getAddress(optagency,name) {  
    var agencyname=document.mailform.optagency.value;  
    var whoname=document.mailform.name.value;  
    var whoaddress=document.mailform.address.value;  
    var whozip=document.mailform.zip.value;
```

```
// Checking for selected agency  
    if (agencyname=="") {  
        alert('Please select an agency from the list');  
        return;  
    }  
}
```

```
// Checking for entered name  
    if (whoname=="") {  
        alert('Please enter your full name');  
        return;  
    }  
}
```

```
//Switch assigns an address to the corresponding agency name  
    switch (agencyname) {  
        case "March of Dimes":
```

```

        var address="8075 Reading Road Suite 300";
        var zip="45237";
        break;
    case "Adoption Option":
        var address="PO Box 492327";
        var zip="45242";
        break;
    case "Adoption Connection":
        var address="11223 Cornell Park Drive";
        var zip="45242";
        break;
    case "Catholic Social Services":
        var address="100 East Eighth Street";
        var zip="45202";
        break;
    case "Planned Parenthood":
        var address="2134 Auburn Avenue";
        var zip="45219";
        break;
    case "University Hospital":
        var address="234 Goodman Street";
        var zip="45267-0753";
        break;
    }
    writeletter(agencyname,address,zip,whoname,whozip,whoaddress);
}

// This function writes the letter
function writeletter(agencyname,address,zip,whoname,whozip,whoaddress) {
// These statements parse the date to mmm/dd/yyyy format
    var date = Date();
    var datestr = new String(date);
    var month = datestr.slice(3,7);
    var day = datestr.slice(8,10);
    var year = datestr.slice(20,24);

    var myWin = window.open("letter.htm","letter","width=400,height=570");
    myWin.moveTo(5,5); myWin.document.write("<title>Print/Preview
Letter</title>");
    myWin.document.write("<Br>");
    myWin.document.write("<Br>");
    myWin.document.write("<Br>");
    myWin.document.write("<Br>");
    myWin.document.write("<Br>");
    myWin.document.write("<Br>");
    myWin.document.write("<Br>");
    myWin.document.write("<Br>");

```

```

myWin.document.write("<Br>");
myWin.document.write(month + " " + day + ", " + year);
myWin.document.write("<Br>");
myWin.document.write("<Br>");
myWin.document.write("<Br>");
myWin.document.write("<Br>");
myWin.document.write(whoaddress);
myWin.document.write("<Br>");
myWin.document.write("Cincinnati, OH " + whozip);
myWin.document.write("<Br>");
myWin.document.write("<Br>");
myWin.document.write(agencyname);
myWin.document.write("<Br>");
myWin.document.write(address);
myWin.document.write("<Br>");
myWin.document.write("Cincinnati, OH " + zip);
myWin.document.write("<Br>");
myWin.document.write("<Br>");
myWin.document.write("Dear " + agencyname, " Representative.");
myWin.document.write("<Br>");
myWin.document.write("<Br>");
myWin.document.write("Please send me information regarding the
services that you offer for pregnant teens.");
myWin.document.write("<Br>");
myWin.document.write("<Br>");
myWin.document.write("Sincerely,");
myWin.document.write("<Br>");
myWin.document.write("<Br>");
myWin.document.write("<Br>");
myWin.document.write("<Br>");
myWin.document.write("<Br>");
myWin.document.write("<Br>");
myWin.document.write(whoname);

myWin.document.close();

```

```

}

```

This is the script that resides on the "letter.htm" page:

```

function printPage(page) {
if(confirm('Would you like to print out this letter?')) {
    print(page);
    return;
} else {
    window.close();
}
}

```

Personal Calendar Script:

This script is similar to the aforementioned script because they both create a new window and prompt the user to print out the page. The Personal Calendar Script is different in content and form. The content is a date object that is derived from the date array variables. I extended the functionality of the original script in this matter. The form of the text is a table instead of a letter. The code for this script is (note that the original sections are italicized):

```
// general purpose function to see if an input value has been entered at all
function isEmpty(inputStr) {
    if (inputStr == "" || inputStr == null) {
        return true
    }
    return false
}

// general purpose function to see if a suspected numeric input
// is a positive integer
function isNumber(inputStr) {
    for (var i = 0; i < inputStr.length; i++) {
        var oneChar = inputStr.charAt(i)
        if (oneChar < "0" || oneChar > "9") {
            return false
        }
    }
    return true
}

// function to determine if value is in acceptable range for this application
function inRangeDay(inputStr) {
    num = parseInt(inputStr)
    if (num < 1 || num > 31) {
        return false
    }
    return true
}

// function to determine if value is in acceptable range for this application
function inRangeYear(inputStr) {
    num = parseInt(inputStr)
    if (num < 1999 || num > 2100) {
        return false
    }
    return true
}
```

```

}
// Master value validator routine for day
function isValidDay(inputStr) {
    if (isEmpty(inputStr)) {
        alert("Please enter the number of the day into the day field before clicking the
Calculate button.")
        return false
    } else {
        if (!isNumber(inputStr)) {
            alert("Please make sure the day is a number only.")
            return false
        } else {
            if (!inRangeDay(inputStr)) {
                alert("Please enter a valid day.")
                return false
            }
        }
    }
    return true
}
// Master value validator routine for year
function isValidYear(inputStr) {
    if (isEmpty(inputStr)) {
        alert("Please enter the year into the year field before clicking the Calculate
button.")
        return false
    } else {
        if (!isNumber(inputStr)) {
            alert("Please make sure the year is a number only.")
            return false
        } else {
            if (!inRangeYear(inputStr)) {
                alert("Please enter a valid year.")
                return false
            }
        }
    }
    return true
}
function makeArray(n) {
    this.length = n
    for (var i=1; i <= n; i++)
        this[i] = null
    return this
}
var maxday = new makeArray(12)

```

```

maxday[1] = 31
maxday[2] = 28
maxday[3] = 31
maxday[4] = 30
maxday[5] = 31
maxday[6] = 30
maxday[7] = 31
maxday[8] = 31
maxday[9] = 30
maxday[10] = 31
maxday[11] = 30
maxday[12] = 31
var monthname = new makeArray(12)
monthname[1] = "January"
monthname[2] = "February"
monthname[3] = "March"
monthname[4] = "April"
monthname[5] = "May"
monthname[6] = "June"
monthname[7] = "July"
monthname[8] = "August"
monthname[9] = "September"
monthname[10] = "October"
monthname[11] = "November"
monthname[12] = "December"
var adddays = new makeArray(7)
adddays[1] = 14
adddays[2] = 36
adddays[3] = 60
adddays[4] = 84
adddays[5] = 108
adddays[6] = 132
adddays[7] = 156
adddays[8] = 180
adddays[9] = 204
adddays[10] = 228
adddays[11] = 252
adddays[12] = 280
// Calculate the date string
function calcNewDate(month,day,year,adddays) {

    newday = eval(day) + adddays
    newmonth = month + 1
    newyear = eval(year)
    var max
    for (var i = 0; i < 12; i++) {

```

```

if (newmonth == 2 && (newyear % 4) == 0) {
    max = 29
} else
    max = maxday[newmonth]
if (newday > max) {
    newday = newday - max
    newmonth = newmonth + 1
    if (newmonth > 12) {
        newyear = newyear + 1
        newmonth = 1
    }
}
else
    break
}
var datestring = monthname[newmonth] + " " + newday + ", " + newyear
return datestring
}
// Get the date entered and calculate the rest of the dates
function calc(form) {
    day = form.day.value
    year = form.year.value
    monthnum = form.month.selectedIndex
    var winform = window.open('calendar.htm', '', 'width=500,height=500');
    if (isValidDay(day)) {
        if (isValidYear(year)){
// Create new page with calendar print out
    winform.document.write("<title>Preview/Print Calendar</title>");
winform.document.write("<h3 align='center'>Personal Calendar</h3>");
    winform.document.write("<table align='center' border='1'width='95%'
bordercolor='#c3c3c3'>");
    winform.document.write("<tr>");
    winform.document.write("<td>");
    winform.document.write("Conception likely occurred around: ");
    winform.document.write("</td>");
    winform.document.write("<td align='center'><b>");
    winform.document.write(calcNewDate(monthnum,day,year,adddays[1]));
    winform.document.write("</td></b>");
    winform.document.write("</tr>");
    winform.document.write("<tr>");
    winform.document.write("<td>");
    winform.document.write("Plan to see doctor around: ");
    winform.document.write("</td>");
    winform.document.write("<td align='center'><b>");
    winform.document.write(calcNewDate(monthnum,day,year,adddays[2]));
    winform.document.write("</td></b>");

```

```

winform.document.write("</tr>");
winform.document.write("<tr>");
winform.document.write("<td>");
winform.document.write("Plan to see doctor around: ");
winform.document.write("</td>");
winform.document.write("<td align='center'><b>");
winform.document.write(calcNewDate(monthnum,day,year,adddays[3]));
winform.document.write("</td></b>");
winform.document.write("</tr>");
winform.document.write("<tr>");
winform.document.write("<td>");
winform.document.write("Plan to see doctor around:<br>Approximate end of
first term");
winform.document.write("</td>");
winform.document.write("<td align='center'><b>");
winform.document.write(calcNewDate(monthnum,day,year,adddays[4]));
winform.document.write("</td></b>");
winform.document.write("</tr>");
winform.document.write("<tr>");
winform.document.write("<td>");
winform.document.write("Plan to see doctor around: <br>Schedule Ultra
Sound");
winform.document.write("</td>");
winform.document.write("<td align='center'><b>");
winform.document.write(calcNewDate(monthnum,day,year,adddays[5]));
winform.document.write("</td></b>");
winform.document.write("</tr>");
winform.document.write("<tr>");
winform.document.write("<td>");
winform.document.write("Plan to see doctor around: ");
winform.document.write("</td>");
winform.document.write("<td align='center'><b>");
winform.document.write(calcNewDate(monthnum,day,year,adddays[6]));
winform.document.write("</td></b>");
winform.document.write("</tr>");
winform.document.write("<tr>");
winform.document.write("<td>");
winform.document.write("Plan to see doctor around: ");
winform.document.write("</td>");
winform.document.write("<td align='center'><b>");
winform.document.write(calcNewDate(monthnum,day,year,adddays[7]));
winform.document.write("</td></b>");
winform.document.write("</tr>");
winform.document.write("<tr>");
winform.document.write("<td>");

```

```

winform.document.write("Plan to see doctor around: <br>Approximate end of
second term");
winform.document.write("</td>");
winform.document.write("<td align='center'><b>");
winform.document.write(calcNewDate(monthnum,day,year,adddays[8]));
winform.document.write("</td></b>");
winform.document.write("</tr>");
winform.document.write("<tr>");
winform.document.write("<td>");
winform.document.write("Plan to see doctor around: <br>Attend
Parenting/Childbirth Classes");
winform.document.write("</td>");
winform.document.write("<td align='center'><b>");
winform.document.write(calcNewDate(monthnum,day,year,adddays[9]));
winform.document.write("</td></b>");
winform.document.write("</tr>");
winform.document.write("<tr>");
winform.document.write("<td>");
winform.document.write("Plan to see doctor around: ");
winform.document.write("</td>");
winform.document.write("<td align='center'><b>");
winform.document.write(calcNewDate(monthnum,day,year,adddays[10]));
winform.document.write("</td></b>");
winform.document.write("</tr>");
winform.document.write("<tr>");
winform.document.write("<td>");
winform.document.write("Plan to see doctor around: <br> Find baby doctor");
winform.document.write("</td>");
winform.document.write("<td align='center'><b>");
winform.document.write(calcNewDate(monthnum,day,year,adddays[11]));
winform.document.write("</td></b>");
winform.document.write("</tr>");
winform.document.write("<tr>");
winform.document.write("<td>");
winform.document.write("Your due date is: ");
winform.document.write("</td>");
winform.document.write("<td align='center'><b>");
winform.document.write(calcNewDate(monthnum,day,year,adddays[12]));
winform.document.write("</td></b>");
winform.document.write("</tr>");
winform.document.write("</table>");
winform.document.close();
}
}
}

```

Appendix B

Visual Basic Contribution:

This form is actually a ActiveX component. Most properties of the form are defined via the properties dialog box. The code for the VB project is:

```
'Define the variables:
Public StartingAddress As String
Dim apploc As String
Dim locpath As String

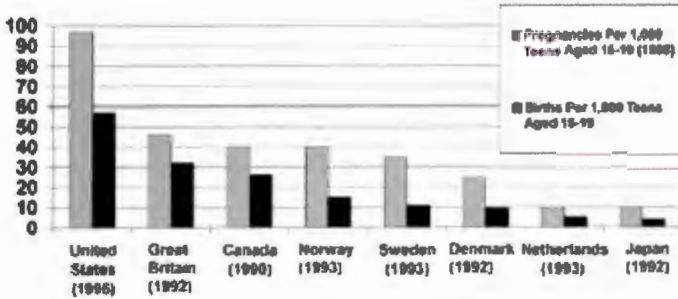
'Find CD-ROM drive, then load form
Private Sub Form_Load()
    On Error Resume Next
    apploc = App.Path
    locpath = RTrim(apploc)
    StartingAddress = "file:\\\" & locpath & "\\project\home.html"
'Navigate to starting page.
    brwWebBrowser.Navigate StartingAddress
End Sub
```

The brwWebBrowser is the ActiveX component that provides the functionality for a browser. The Navigate method opens the specified URL. AppPath returns the appropriate drive letter that contains the CD-ROM drive. This string is then trimmed to expose the drive letter. Finally, this letter is concatenated with the URL for the primary HTML file for the document.

Appendix C

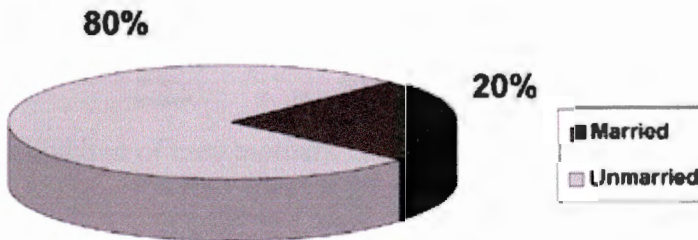
These charts are taken from the www.teenpregnancy.org website. They contain information relevant to the user audience.

Additional Statistics and Charts



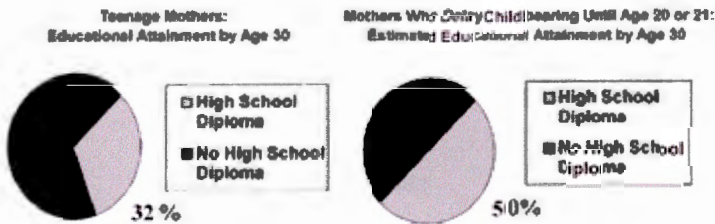
Teen pregnancy rates are higher in the United States than in any other industrialized country.

Whatever Happened to Childhood, NCPTP



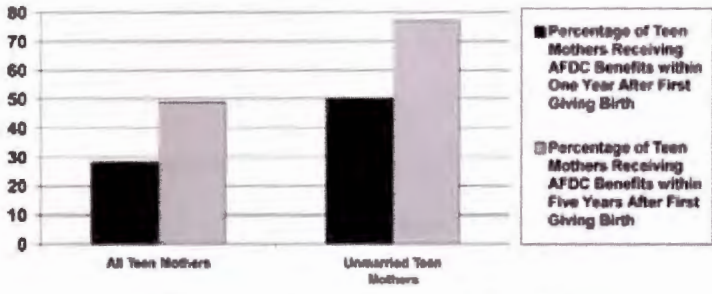
Most pregnant teens are unmarried.

Whatever Happened to Childhood, NCPTP

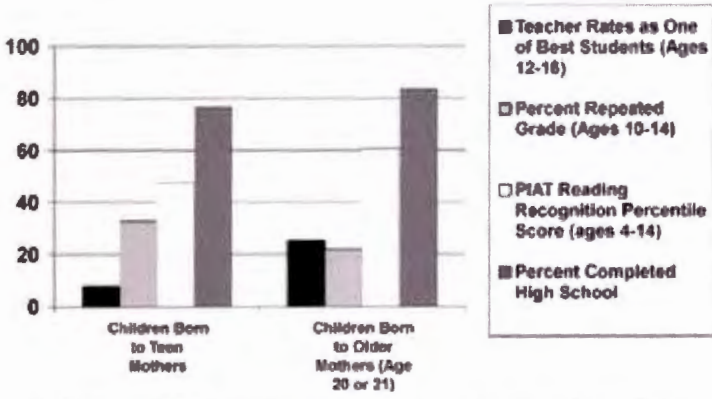


Teen mothers are less likely to complete high school.

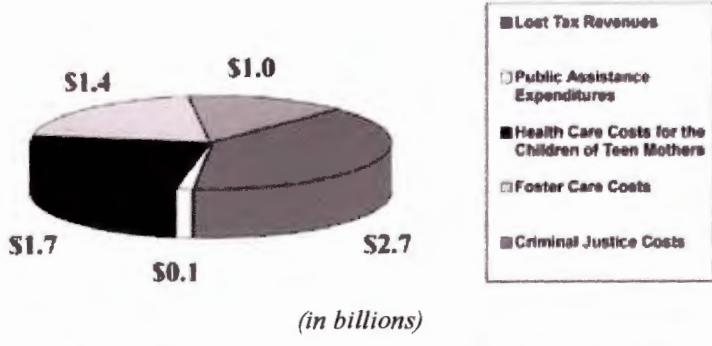
Whatever Happened to Childhood, NCPTP



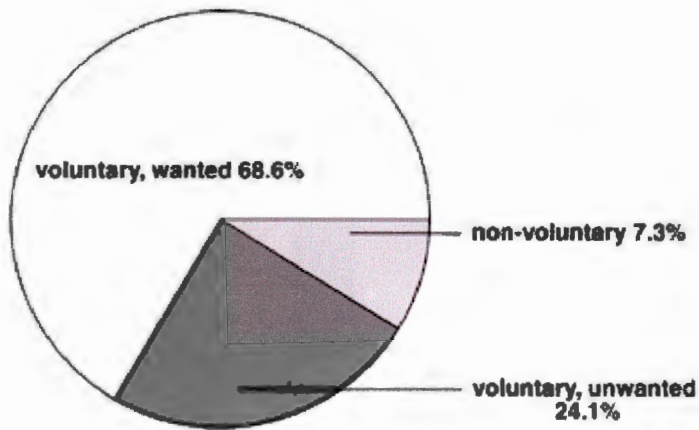
Economic future of teen moms
Whatever Happened to Childhood, NCPTP



The children of teen mothers have poorer school performance.
Whatever Happened to Childhood, NCPTP

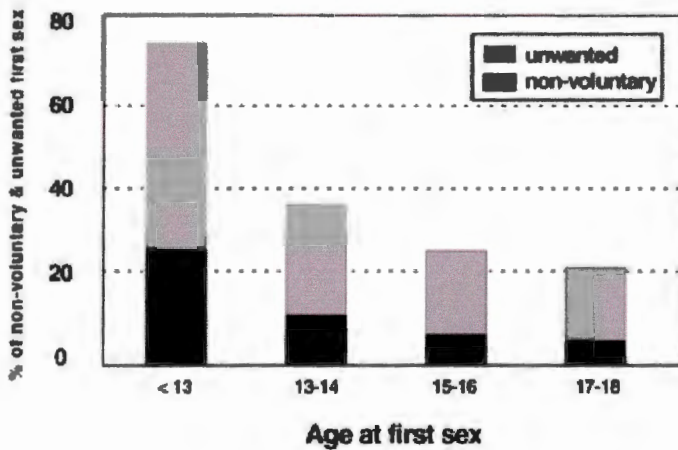


Teen childbearing costs taxpayers \$7 billion a year.
Whatever Happened to Childhood, NCPTP



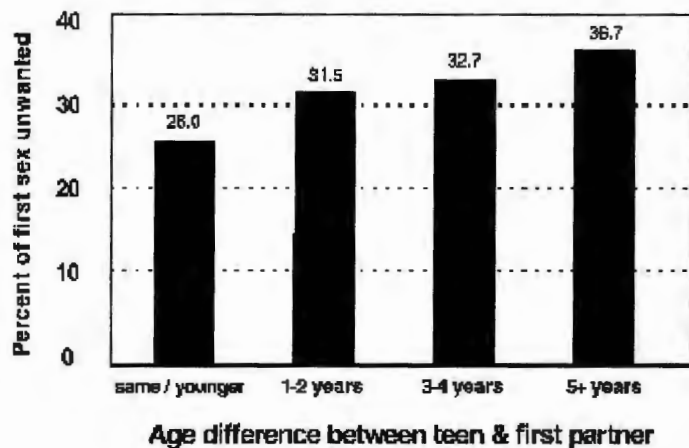
Of girls 15-19, 7% said their first sexual experience was not voluntary, including those that said that they were raped.

Not Just For Girls, NCPTP



Teen girls who have sex at younger ages are more likely to say that sex was either forced or unwanted.

Not Just For Girls, NCPTP



The greater the age difference between girls and their partners, the more likely that first sexual encounter was unwanted by the girl.

Not Just For Girls, NCPTP

References:^P

- Brown, Al. "Use of Storyboards as a Visual Guide for Powerful Presentations". *Presentations*. August, 1997. 33-36.
- Goldberg, Ron. *Multimedia Producers Bible*. Fort City, CA: IDG Books Worldwide Inc., 1996.
- Hinkin, Sue. "Not Just Another Pretty Face: 10 Tips for the Most Effective Use to Type". *Presentations*. January, 1995. 34ff.
- Parker, Roger. "What Color is Your Presentation?". *Presentations*. March, 1995. 32ff.
- Weinschenk, Susan. *GUI Design Essentials*. New York: Wiley Computer., 1997. 159-236.
- Zielinski, Dave. "Match the Medium to the Message". *Presentations*. October, 1997. 37ff.

Notes

^A This site <http://www.agi-usa.org/> contains several documents assessing teen pregnancy.

^B Several of the on-line sites dealing with teen pregnancy and public policy refer to this information. Also, the information specialist at the library in Planned Parenthood confirmed the Institute as authoritative.

^C These sites were extensive and authoritative. Each is geared to a particular audience. The first listed, Teen Pregnancy, is an excellent site targeted for teens and adults seeking quick information about the topic. The Planned Parenthood site is graphically well crafted and designed for mature discussions. They have an additional site specifically for teens at www.teenwire.org. Finally, the Mayo Health Institute has an on-line site that gives medically accurate information on various help topics. This site is not targeted for teens.

^D Rosefeld, L. and Morville, P. *Information Architecture for the World Wide Web*. New York: O'Reilly Assoc., 1998.

^E Levine, R. "Guide to Web Style." [Http://www.sun.com/styleguide/](http://www.sun.com/styleguide/). 1995.

^F Goldberg, Ron. *Multimedia Producers Bible*. Foster City, CA: IDG Books Worldwide, 1996.

^G Visit the Webmonkey at www.webmonkey.com. The tutorials cover a myriad of topics from technical programming to graphic design.

^H Each tool came with an online tutorial respective to its features.

^I *U.S. Teenage Pregnancy Statistics*, New York: Alan Guttmacher Institute, May, 1996; this reference, as well as other relevant documents, can be viewed on-line at <http://www.agi-usa.org/>.

^J The "typical" criteria are based upon my personal visits to three facilities. The clinic for teens at University Hospital and Planned Parenthood clinic provide computers with Microsoft OS, printers, and Internet access.

^K The World Wide Web Consortium defines some standards for the web. The HTML standard is published at their website <http://www.wc3.org/>. The quick reference I used for the tags is "A Bare Bones Guide to HTML". A link to this reference can be found at <http://webach.com/barebones/>.

^L I realize the subjectivity of this statement. However, this criteria speaks again to the audience.

^M This cannot be captured in a single screen shot. However, Figure 1 above does give a glimpse of the page. View the "home.htm" page of the project itself for the full effect.

^N This is a common feature to Internet sites dealing with pregnancy. However, none of the calendars that I encountered provide a prompt or even a friendly option for a print out.

^O Type "teen pregnancy" into a search engine and sit back. This list covers the major topics of the ePrimer and provides secure chat and reliable health information for the teen.

^P These references represent some of the materials that contributed to the project. Others are listed in the Notes section of this document.