

An Automatic Website Generator (WebGen)

Project 1 Computer Engineering (Final Report)

This Project is done by maknoon.com team

Project team:

*Naser Ali Mohod
Ahmad Abdulah
Ibrahim Ali Saif Al-Jarwan*

Student ID Number:

1239
1208
1224

Project Supervisor:

Prof. David Vernon

Acknowledgements

In life, humans cannot live without other people, and this is for the best, because even if he works hard no one knows everything.

First of all we would like to thank God for giving us the ability to finish and do our best in this project. Special thanks goes to our supervisor Prof. David Vernon for his valuable advice and encouragement throughout the project period.

Finally, we would like to extend our thanks to our families, friends and all the staff who have helped us in one way or another in doing our project.

Abstract

The purpose of this project is to develop software that will automatically convert a text based web specification file into a series of structured web pages, which are navigated using drop-down menus.

This project will help webmasters as well as normal users to build structured web pages with customized page templates in a few seconds.

The user has to edit a text file called menutree.txt to specify the structure of the web site and it has a specific format which must be followed. He can also customize the header (the top part of the page), the default page (it will appear in the body if the user doesn't specify the page's body) and the style of menu by editing the input files located under the webgen_data folder.

This project is written in C++ language in an object oriented programming method.

Table of Contents

Acknowledgements	2
Abstract	3
Chapter 1: Introduction	
1.1 Goal and Objectives	8
1.2 Overview of the project	8
1.3 Overview of the report	8
Chapter 2: System Overview	
2.1 Requirements	
2.1.1 Function	9
2.1.2 Performance & Behavior	9
2.1.3 Constraints	9
2.1.4 Interfaces	9
2.1.5 Reliability	9
2.2 System Specifications	
2.2.1 Objective	10
2.2.2 Design issues	
2.2.2.1 Details of the exact functionality of the system	10
2.2.2.2 Rules governing the content of menutree.txt	12
2.2.2.3 Overview of the Web Generation Process	13
2.2.2.4 Details of the system interface	14
2.2.3 Theoretical Model	14
Chapter 3: Implementation	
3.1 Create menutree class	15
3.2 Process Modeling	16
3.3 Restriction	17
Chapter 4: Testing	
4.1 Verification	18
4.2 Validation	20
Chapter 5: Conclusion	
5.1 Summary	24
5.2 Critical Appraisal	25
5.3 Future work	25

References	25
Appendix A	26

List of illustrations:

Chapter 2

Figure 2.1 Graphical version of the menutree.txt	11
--	----

Chapter 3

Figure 3.2 DFD (0-Level Diagram)	16
Figure 3.1 DFD (Context Diagram)	17

Chapter 4

Figure 4.1 Typical input of menutree.txt	18
Figure 4.2 Typical output of menutree.txt	19
Figure 4.3 Typical output of menutree.txt to show the URLs of web-pages	19
Figure 4.4 menutree.txt when it contains a free line i.e. no menu item (here is line 2)	20
Figure 4.5 The error message of the menutree.txt when it contains a free	20
Figure 4.6 menutree.txt when there is a different of 2 taps between tow sequential lines	20
Figure 4.7 The error message when there is a different of 2 taps between tow sequential lines	21
Figure 4.8 menutree.txt when the first menu item is started with '/'	21
Figure 4.9 The error message when the first menu item is started with '/'	21
Figure 4.10 menutree.txt when the first & the last menu items are links	22
Figure 4.11 The output of the menutree.txt when the last menu item is a link	22

Figure 4.12

The output of the menutree.txt when the first menu item is a link

23

Appendix A

Figure 1

A general look of all folders and files

26

Chapter 1: Introduction

1.1 Objective and goals

The goal of this project is to develop a program that will automatically convert a text based web specification file into a series of structured web pages, which are navigated using drop-down menus. The specification file specifies a hierarchical (tree-like) configuration of the website menu system with non-leaf nodes representing menu entries and leaf nodes representing web pages.

The generator must automatically write the HTML to create the drop-down menu system and create a web page for each appropriate menu item.

1.2 Overview of the project

The target website is configured as a menu hierarchy – or menu tree – with internal nodes on the tree representing menus and leaf nodes representing web pages.

The site is navigated by series of drop-down menus that directly reflect this hierarchical structure. These menus are implemented using JavaScript.

It is designed this way so as to make it easy to extend and add content by growing the menu tree. Thus, to add a page, all a contributor needs to do is specify the menu item and provide the HTML file. To add a new menu, all one needs to do is specify a new leaf in the present tree (i.e. new menu item) and provide the new menu (sub-) tree together with the HTML for all the leaf nodes. The menu tree for the entire website is contained in a file called `menutree.txt`

1.3 Overview of the report

This report is divided into five chapters.

In Chapter one, a general outline to the project will be introduced.

In Chapter two, the system requirements will be shown, which will include information about the system requirements and specifications, functional specifications, inputs, outputs, and the limitations of our program.

Chapter Three, which is divided into the input, and the output, deals with the implementation of the program.

Chapter Four includes different tests that are run on the program to make sure that it works correctly.

Chapter Five includes the conclusion, critical appraisal and future recommendations.

A reference page and appendixes, which include the user manual and the c++ source code of the program, will be attached to the end of the report.

Chapter 2: System Overview

2.1 The Requirements

2.1.1 Function

The objective of this project is to develop a program that will automatically convert a text based web specification file into a series of structured web pages, which are navigated using drop-down menus. The specification file specifies a hierarchical (tree-like) configuration of the website menu system with non-leaf nodes representing menu entries and leaf nodes representing web pages.

The generator must automatically write the HTML to create the drop-down menu system and create a web page for each appropriate menu item.

2.1.2 Performance & Behavior

Just running the program and it will automatically generate the required HTML files.

2.1.3 Constraints

The program will not work on less than Explorer 4 or Netscape 4, because of the limitation of the script of the menu that is written in JavaScript. The style of the menu is defined by the user through the input file menu_array_style.txt

2.1.4 Interfaces

Input files:

./webgen_data/menutree.txt	-contains the definition of the menu hierarchy
./webgen_data/webgen_template.txt	-contains the HTML for the header on each web page
./webgen_data/webgen_default.txt	-contains the HTML for the default 'under construction' page
./webgen_data/menu_array_style.txt	-this defines the look and feel of the menu (that is used by the menu JavaScript)

Output files:

In addition to generating the HTML for all the leaf nodes in the menu tree, WebGen also generates the following files:

./website/target_js/menu_array.js	-contains the JavaScript menu definitions
./website/index.htm	-contains the input HTML for the Home page

The file format will be set out in the specification that follows:

./website/home/Home.htm

This will contain the Home page and will be constructed automatically from:

./website/home/Home.txt

2.1.5 Reliability

The output is very reliable if we use the suitable browser.

2.2 The Specification

2.2.1 Objective

As noted in the Requirement section “The objective of this project is to develop a program that will automatically convert a text based web specification file into a series of structured web pages, which are navigated using drop-down menus. The text file specifies a hierarchical (tree-like) configuration of the website menu system with non-leaf nodes representing menu entries and leaf nodes representing web pages.

The generator must automatically write the HTML to create the drop-down menu system and create a web page for each appropriate menu item.”

2.2.2 Design issues

2.2.2.1 Details of the exact functionality of the system

The target website is configured as a menu hierarchy – or menu tree – with internal nodes on the tree representing menus and leaf nodes representing web pages.

The site is navigated by series of drop-down menus that directly reflect this hierarchical structure. These menus are implemented using JavaScript

It is designed this way so as to make it easy to extend and to add content by growing the menu tree

Thus, to add a page, all a contributor needs to do is to specify the menu item and provide the HTML file

To add a new menu, all one needs to do is specify a new leaf in the present tree (i.e. a new menu item) and provide the new menu (sub-) tree together with the HTML for all the leaf nodes.

The menu tree for the entire website is contained in a file called `menutree.txt`

This file will reside in a specified directory along with any other configuration files. (e.g. `webgen_data`).

Specification of the syntax of `menutree.txt`

This is a specification of the tree by a depth-first traversal, with the depth in the tree being represented by the number of tab characters preceding the node.

Each node is either another menu tree or a leaf node.

Leaf nodes represent the filenames of the web-pages. These filenames have no extension, which is they are the roots of filenames.

For example the following is a partial specification of a target website with only the menu for the Etisalat University menu being fully specified.

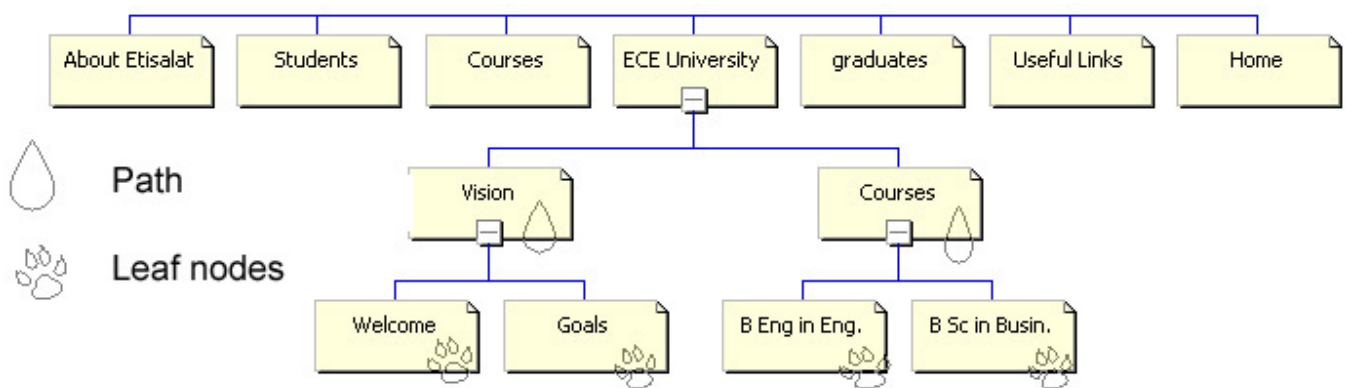


Figure 2.1 Graphical version of menutree.txt

```

Home
About Etisalat
Students
Courses
Etisalat University
    The Etisalat Vision
        MEC/Welcome
        MEC/Goals of the University
    Courses and Degrees
        Courses/B Eng in Engineering
        Courses/B Sc in Business Studies
Graduates
Useful Links
  
```

Semantics of menutree.txt

In the above example, the 'Etisalat University' item invokes a menu with three items:

- The Etisalat Vision
- Courses and Degrees

In turn, these items invoke other menus. In the case of 'Courses and Degrees', it items:

- B Eng in Engineering
- B Sc in Business Studies

All three of these are leaf nodes in the menu tree and, therefore, they correspond to web pages that can be browsed (and not navigation menus). As such some HTML code has to be provided by the user to represent the content of these pages.

The only HTML that a contributor to the target website has to provide is the actual content of the page: the entire HTML associated with the banners and menu-bar, i.e. all the wrapping for the page, is taken care of automatically.

2.2.2.2 Rules governing the content of menutree.txt

To simplify the management of the site, we impose some strict rules on the manner in which you specify the menu sub-tree (i.e. the way you want to extend the menu system) and the way you provide your data. These are as follows:

- Leaf node menu items are represented by any alphanumerical string
 - Spaces are allowed
 - Pathname characters such as '/', '\', '.' are not allowed
 - Upper and lower case characters are encouraged to improve menu readability
 - Example: My New Menu Item
- Each leaf node menu item must have a corresponding text file with the HTML for that page:
 - The filename must be exactly the same as the string describing the menu item, with spaces replaced by underscore character "_", and respecting the case of each character
 - The filename extension must be .txt (e.g. *My_New_Menu_Item.txt*)
 - The file must be placed in subdirectory / folder: the name of this folder should be the name of the author (e.g. MEC or admin)

When specifying the leaf node menu item in the menu sub-tree, prefix the menu item string with the subdirectory /folder name (e.g. *MEC/My_New_Menu_Item*)

- When specifying the level of the menu and menu items in the menu tree, use a tab character for each level descended.

In the example above, you can see immediately, for instance, that the HTML for the 'welcome' and 'Goals' of the 'University' will be held in a folder called 'MEC'.

Similarly, the HTML for the '*B Sc in Business Studies*' will be in sub-directory '*Courses*'

Note well that these files have a .txt extension. This is because they will be automatically copied and inserted into the full web page file, including the wrapping (banners, menu, etc.)

In addition to this menu tree in *./webgen_data/menutree.txt*, the user also provides the following files:

```
./MEC/welcome.txt
./MEC/Goals_of_the_University.txt

./courses/B_Eng_inEngineering.txt
./courses/B Sc_in_Business_Studies.txt
```

2.2.2.3 Overview of the Web Generation Process

WebGen generates a HTML file for every non-URL leaf node in the menu tree. This is done while parsing the `menutree.txt` file, i.e. while performing a depth-first traversal of the menu tree.

The structure of every page is the same and comprises:

1. Header
2. Menu
3. Context Line
4. Body
5. Footer

The HTML for the Header and Menu is taken from

`./webgen_data/webgen_template.txt`

The context line should contain a textual cue about the position of the current page's position in the menu hierarchy (e.g. *'Etisalat University->the Etisalat Vision->Welcome'*).

The HTML for the Body is taken from a file `<menuitem>.txt` where `<menuitem>` stands for the string defining the menu item leaf node with spaces replaced by the underscore character. The case of characters is respected.

If `<menuitem>.txt` does not exist, the body is read from a file containing HTML for an "under construction" page. This is file is

`./webgen_data/webgen_default.txt`

During the traversal, a series of JavaScript menu definition is also created. These form the input for the JavaScript menu functions

The output of the menu definition is placed in a file `./website/target_js/menu_array.js`

This directory also contains the JavaScript itself in `./website/target_js/mmenu.js` (this is not modified)

The menu JavaScript was written by Andy Woolly of Miltonic Solutions; WebGen simply creates the data from (`menu_array.js`) for the script based on the menu tree definition in `menutree.txt`

WebGen uses one other input file (other than `menutree.txt`) in generating `menu_array.js`

`./webgen_data/menu_array_style.txt` -- this defines the look and feel of the menus

There is one other special web page which needs additional processing by WebGen. This is:

`./website/home/Home.htm`

`Home.htm` is the index page of the Target website. Its input is taken in a standard fashion from:

`./website/home/Home.txt`

2.2.2.4 Details of the system interface

As noted in the Requirement section:

Input files:

<code>./webgen_data/menutree.txt</code>	-contains the definition of the menu hierarchy
<code>./webgen_data/webgen_template.txt</code>	-contains the HTML for the header and the menu on each web page
<code>./webgen_data/webgen_default.txt</code>	-contains the HTML for the default 'under construction' page
<code>./webgen_data/menu_array_style.txt</code>	-this defines the look and feel of the menus

Output files:

In addition to generating the HTML for all the leaf nodes in the menu tree, WebGen also generates the following files:

<code>./website/target_js/menu_array.js</code>	-contains the JavaScript menu definitions
<code>./website/home/Home.txt</code>	-contains the input HTML for the Home page

2.2.3 Theoretical Model

JavaScript

JavaScript is used to improve the design, validate forms, and much more. JavaScript was developed by Netscape and is the most popular scripting language on the internet. JavaScript works in all major browsers that are version 3.0 or higher.

What is JavaScript?

- JavaScript was designed to add interactivity to HTML pages
- JavaScript is a scripting language - a scripting language is a lightweight programming language
- A JavaScript is lines of executable computer code
- A JavaScript is usually embedded directly in HTML pages
- JavaScript is an interpreted language (means that scripts execute without preliminary compilation)
- Everyone can use JavaScript without purchasing a license
- JavaScript is supported by all major browsers, like Netscape and Internet Explorer

Chapter 3: Implementation

3.1 Create menutree class that includes:

- **Check member-function that:**
 - Check the syntax of the menutree.txt
 - Displaying a message of errors. In case of
 - An incorrect input structure (e.g. three tabs after a line contains one tab).
 - Using '/', '\', '.' characters.
- **menu member-function that:**
 - Reads the menutree.txt
 - Read the menu items from menutree.txt and store it in a data structure using a flexible and efficient algorithm that is fast and allow unlimited sublevels
 - Extract the menu items from it
 - Return the menu items which are stored in the data structure using a friend function which can be used by other parts of the program.
- **menu_array member-function that creates the JavaScript file (i.e. menu-array.js):**
 - Read style's parameters which are read from menu_array_style.txt
 - The user has just change the content of the existing variables which change the style of the menu (e.g. colors or the fonts used).
 - The variable must be well commented to help the user modify the style of the menu items.
 - Link menu items (i.e. names and links) using a friend function located in tree class which Extract the menu items from the data structure
 - The program will insert the menu items in the menu_array.js taken from the data structure in a way recognized from the JavaScript script.
- **WebGen member-function that:**
 - Extract the leaf nodes and generates the HTML pages that include the header, menu and default 'under construction' style.
 - It contains:
 - A member function that reads webgen_templet.txt & webgen_default.txt
 - A member function that generates folders & files
 - Replace all unwanted characters (e.g. spaces)
 - Generates the HTML files
 - The files must be placed in a subdirectory/folder: the name of this folder should be the name of the author (e.g. MEC or admin).
 - The needed program's files are located in webgen_data subdirectory these files are:

menutree.txt,
webgen_template.txt,
webgen_default.txt,
menu_array_style.txt

- The JavaScript files are located in a subdirectory called target_js these files are: menu_array.js which contains the style and menu items (the program will create this file) and mmenu.js which contains the actual code which generates the menu therefore, we should not touch it.
- A member function which will read the body text files and will update the generated html files (<body>up to </body>) which has the same names.

3.2 Process Modeling

DataFlow Diagram (DFD)

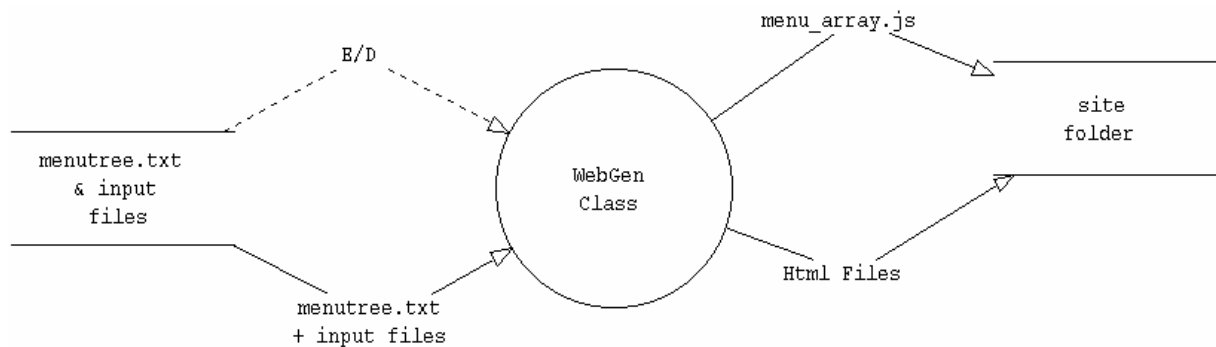


Figure 3.1 DFD (Context Diagram)

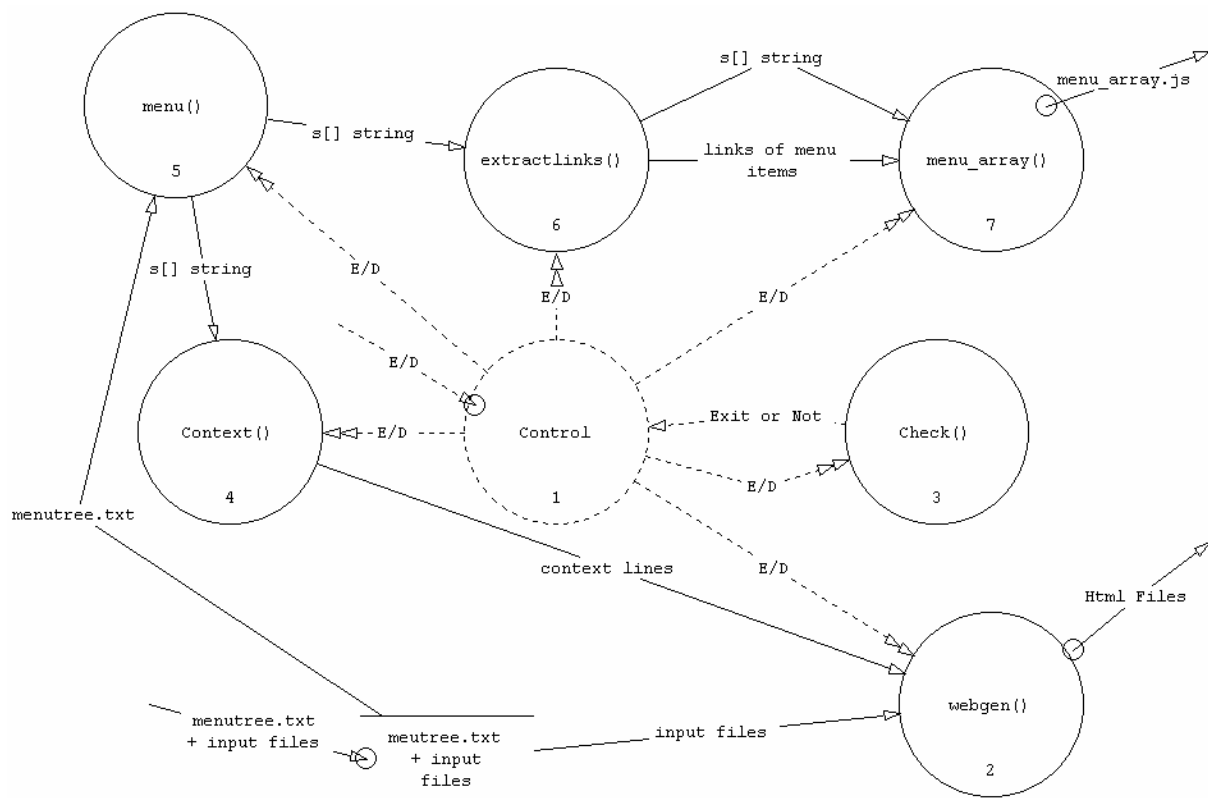


Figure 3.2 DFD (0-Level Diagram)

3.3 Restriction

The program will not work on less than Explorer 4 or Netscape 4, because of the limitation of the script of the menu that is written in JavaScript.

The max number of pages that can be generated at one time is 275 pages unless we use some functions which are used to free memory.

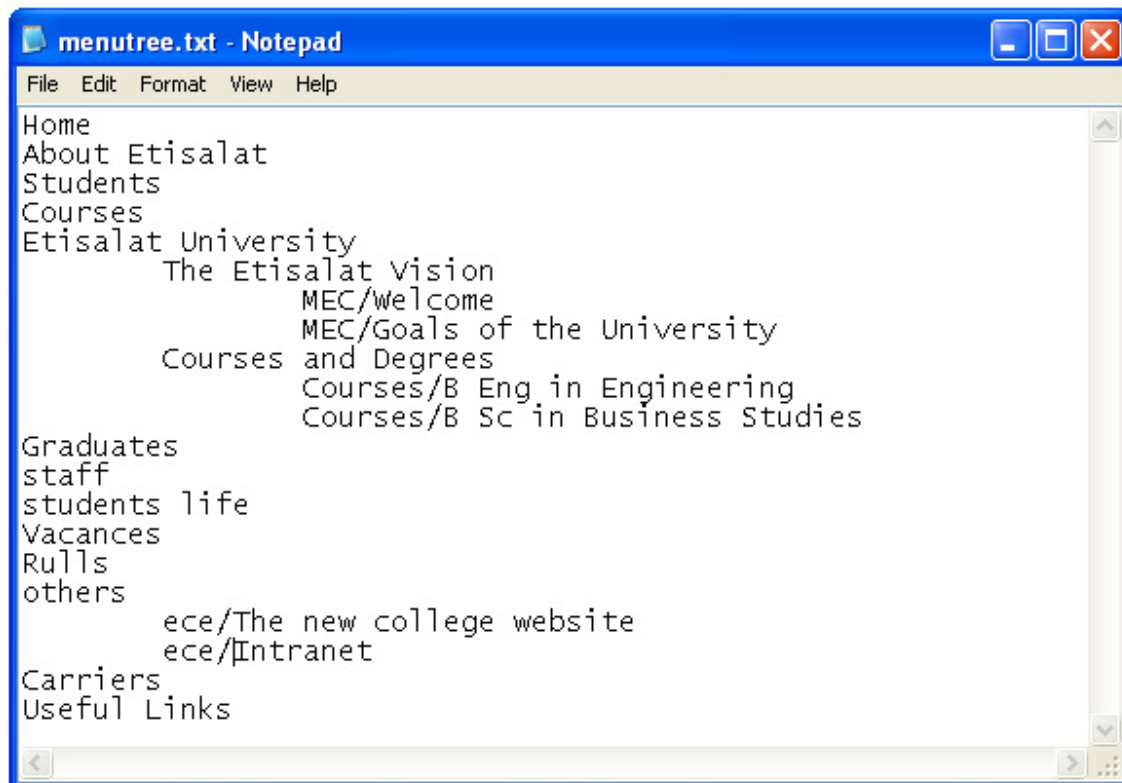
It will not work directly on Linux machines unless we have some changes in the code and that is due to different command used on Linux machines to create folders and that is used on DOS.

Chapter 4: Testing

4.1 Verification

Verification of the implementation can be proved by showing that it satisfies the requirements which are stated in chapter 2. And here is the typical input and its output.

Simple example of a typical menutree.txt file and the output



```
menutree.txt - Notepad
File Edit Format View Help
Home
About Etisalat
Students
Courses
Etisalat University
    The Etisalat Vision
        MEC/Welcome
        MEC/Goals of the University
    Courses and Degrees
        Courses/B Eng in Engineering
        Courses/B Sc in Business Studies
Graduates
staff
students life
Vacances
Rulls
others
    ece/The new college website
    ece/Intranet
Carriers
Useful Links
```

Figure 4.1 Typical input of menutree.txt

N.B All the next menus will be short to explain the different method of inputs

And the output is

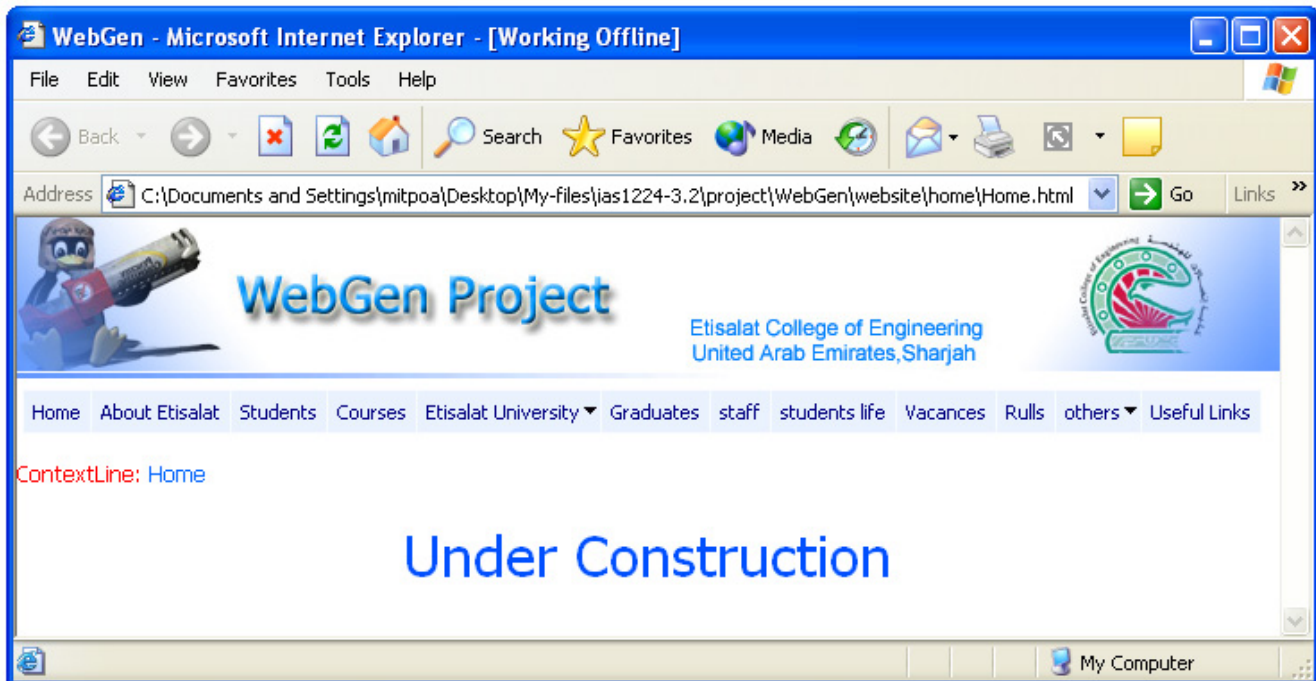


Figure 4.2 Typical output of menutree.txt

Optional feature

Leaf nodes can be URLs of web-pages

E.g. Online courses in the example as shown below

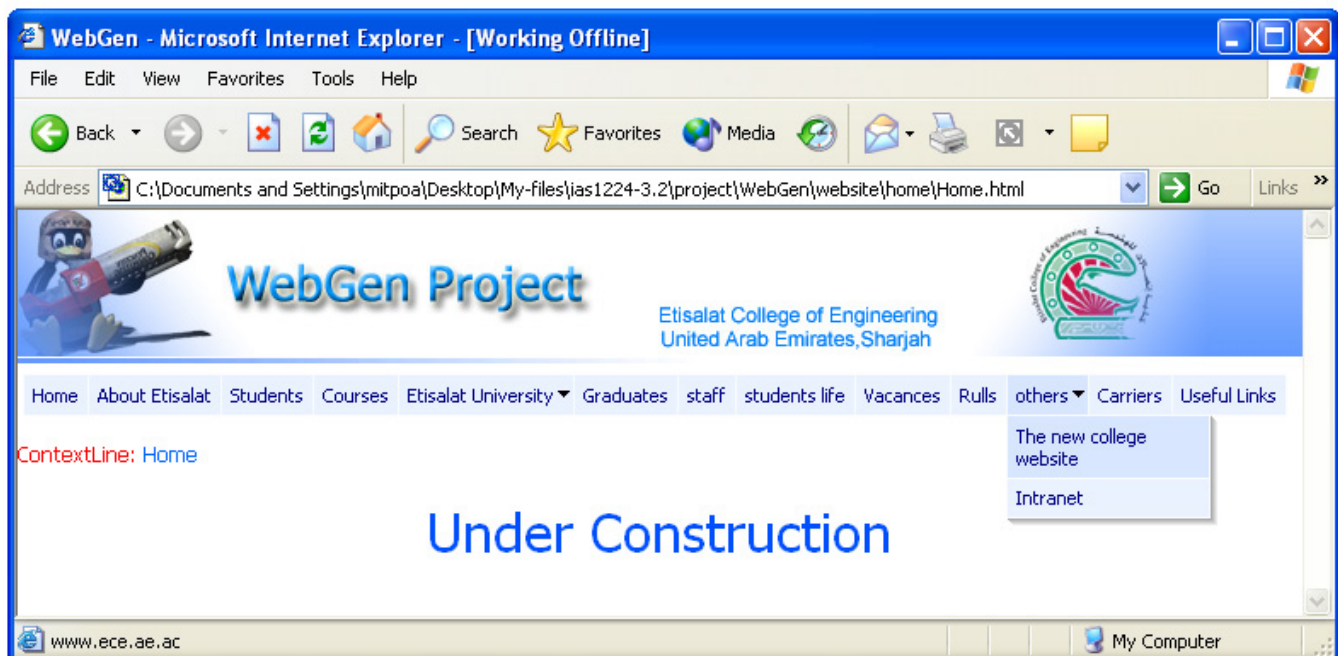


Figure 4.3 Typical output of menutree.txt to show the URLs of web-pages

4.2 Validation

The validation of the implemented system can be proved by showing that it meets the specification stated in chapter 2. The inputs and the outputs satisfy what was stated in the specification.

Different input of menutree.txt files that show the program ability and there outputs

- menutree.txt when it contains a free line (i.e. no menu item)

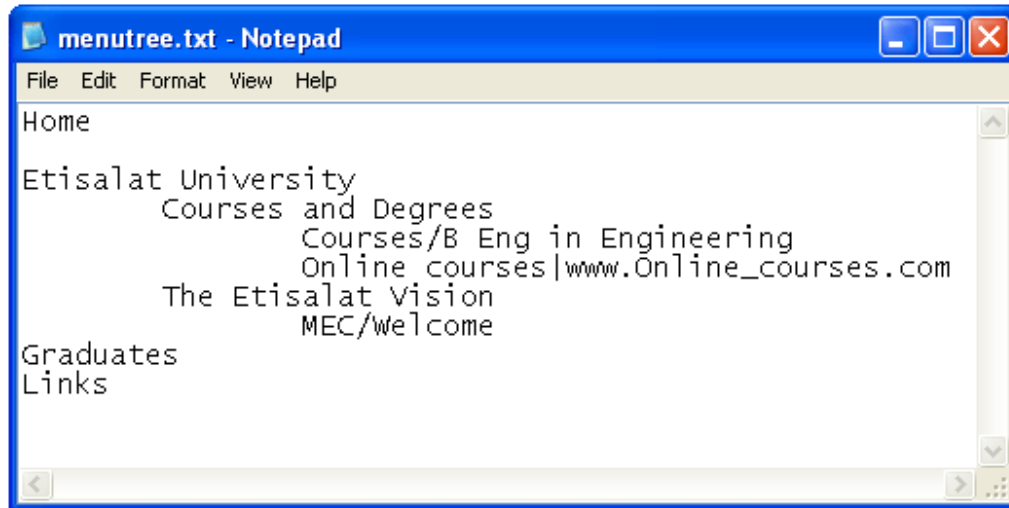


Figure 4.4 menutree.txt when it contains a free line i.e. no menu item (here is line 2)

And the output is

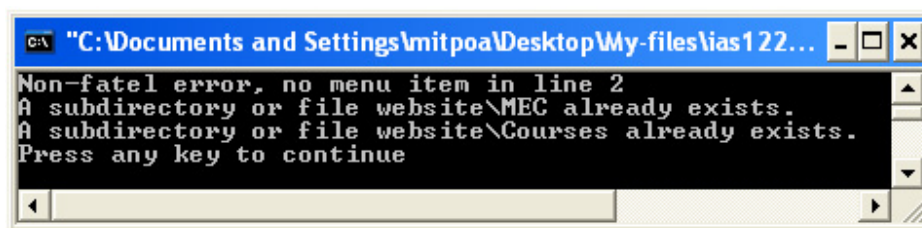


Figure 4.5 the error message of the menutree.txt when it contains a free line

- menutree.txt when there is a different of 2 taps between tow sequential lines

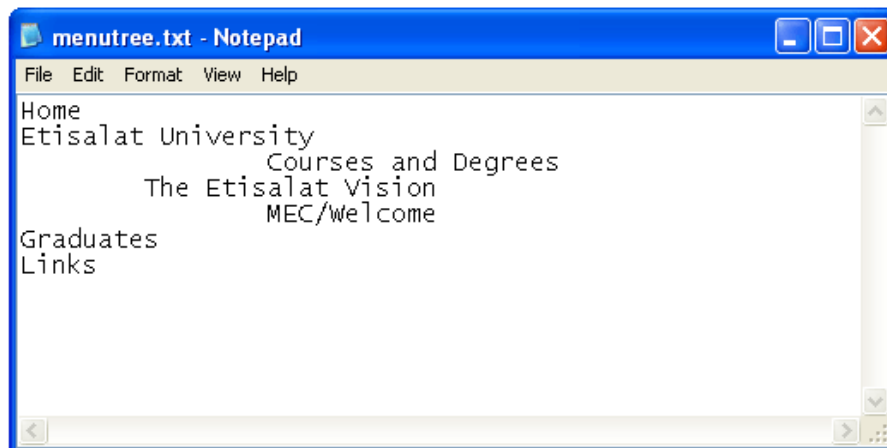


Figure 4.6 menutree.txt when there is a different of 2 taps between tow sequential lines (here they are line 2 & 3)

And the output is

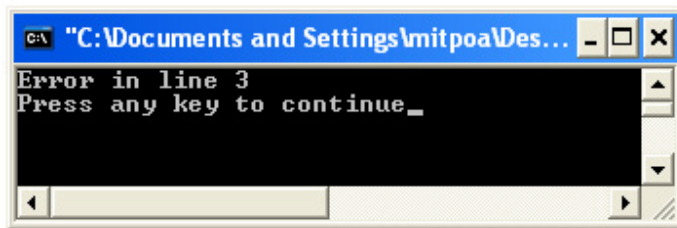


Figure 4.7 the error message of menutree.txt when there is a different of 2 taps between tow sequential lines

- menutree.txt when the first menu item is started with ' / '

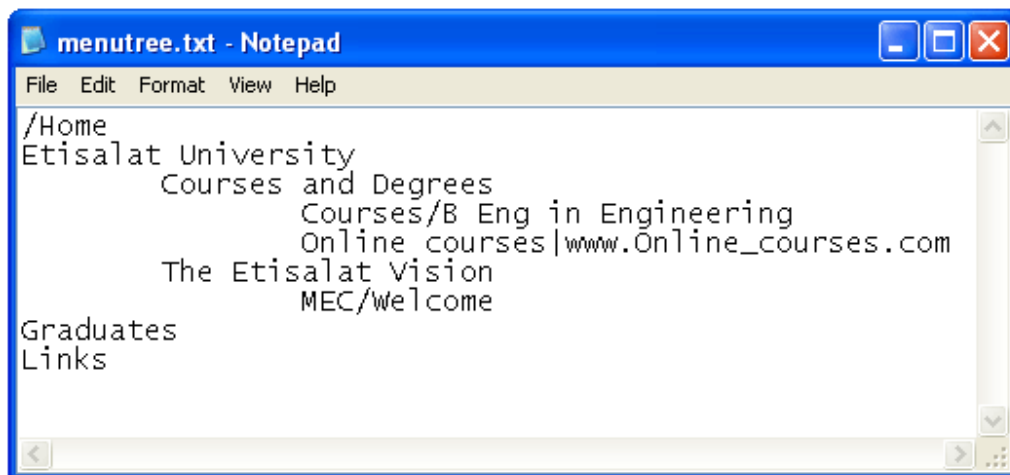


Figure 4.8 menutree.txt when the first menu item is started with ' / ' (Fatal Error due to the JavaScript)

And the output is

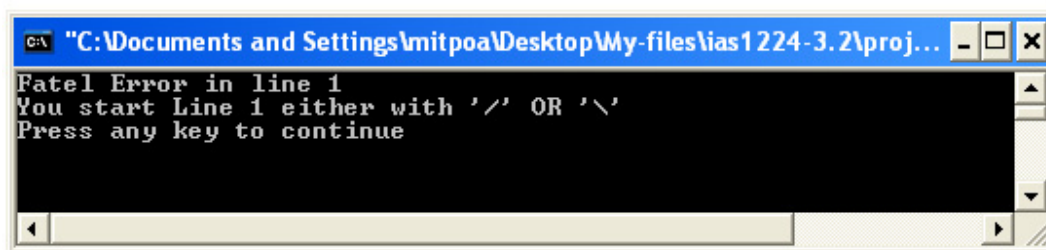


Figure 4.9 the error message of the menutree.txt when the first menu item is started with ' / ' (Fatal Error due to the JavaScript)

- menutree.txt when the first & the last menu items are links

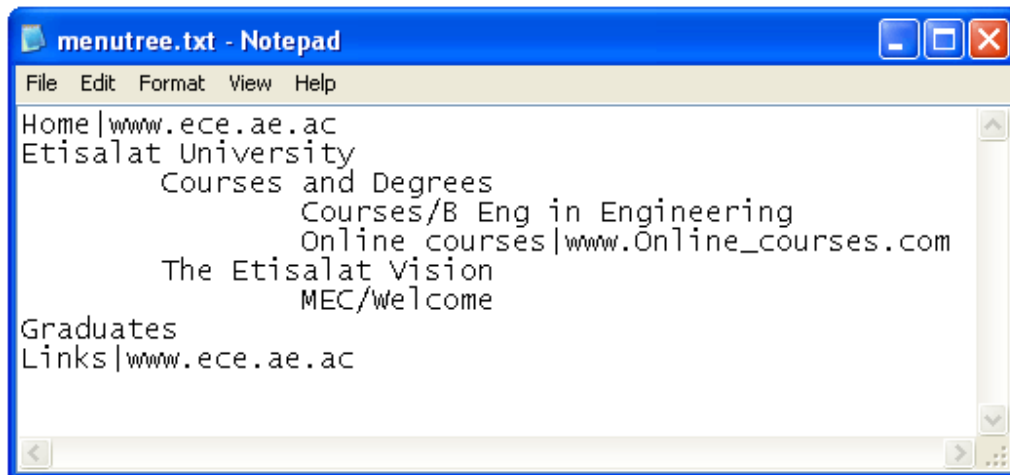


Figure 4.10 menutree.txt when the first & the last menu items are links

And the output is

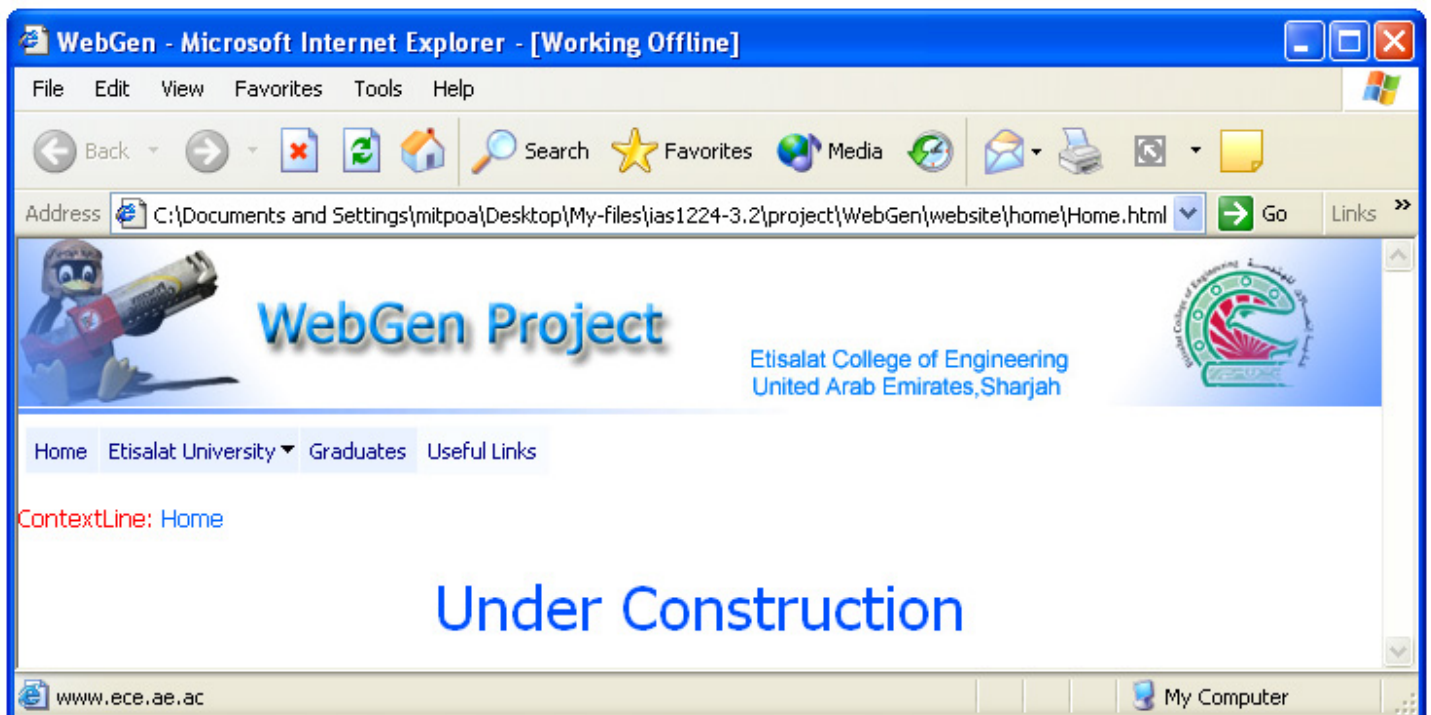


Figure 4.11 the output of the menutree.txt when the first menu item is a link

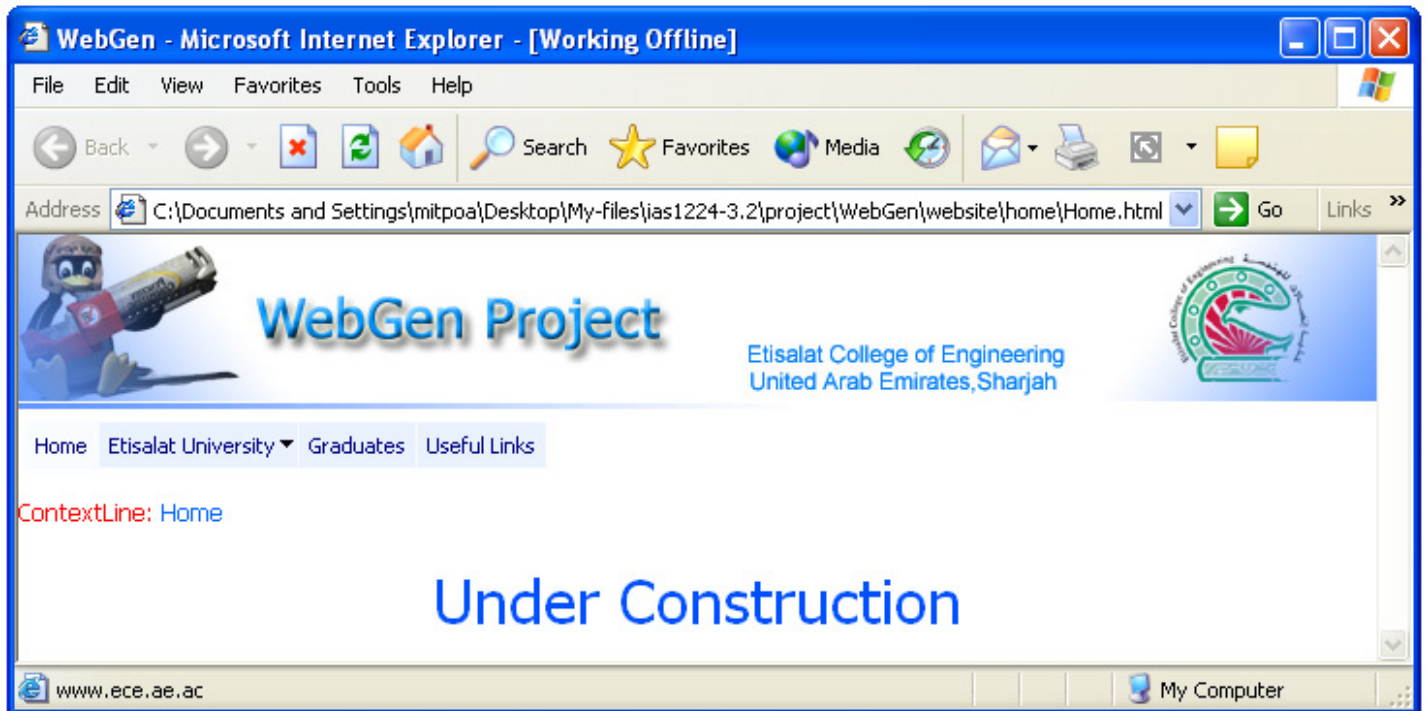


Figure 4.12 the output of the menutree.txt when the last menu item is a link

Chapter 5: Conclusion

5.1 Summary

WebGen software gives webmasters the ability to create large web sites with an easy automated technique.

The aim of this project is to develop software that will automatically convert a text based web specification file into a series of structured web pages, which are navigated using drop-down menus. The text file specifies a hierarchical (tree-like) configuration of the website menu system with non-leaf nodes representing menu entries and leaf nodes representing web pages.

The menu tree for the entire website is contained in a file called `menutree.txt`. This is, in effect, a specification of the tree by a depth-first traversal, with the depth in the tree being represented by the number of tab characters preceding the node.

The only HTML that a contributor to the target website has to provide is the actual content of the page: the entire HTML associated with the banners and menu-bar, i.e. all the wrapping for the page, is taken care of automatically.

There are 4 input files the user should edit located in `webgen_data` folder, these are:

- `menutree.txt`: contains the definition of the menu hierarchy
- `webgen_template.txt`: contains the HTML for the header and the menu on each web
- `webgen_default.txt`: contains the HTML for the default 'under construction'
- `menu_array_style.txt`: this defines the look and feel of the menus

The software will output:

- HTML for all the leaf nodes in the menu
- `menu_array.js`: contains the JavaScript menu definitions
- `Home.txt`: contains the input HTML for the Home page

The user has to edit a text file called `menutree.txt` to specify the structure of the web site and it has a specific format which must be followed. He can also customize the header (the top part of the page), the default page (it will appear in the body of the page if the user didn't specify the page's body) and the style of menu by editing the input files located under the `webgen_data` folder.

The WebGen software is written as a class to be easily plugged in other software as an oriented programming terminology.

The program will not work on less than Explorer 4 or Netscape 4, because of the limitation of the script of the menu that is written in JavaScript.

This project will help webmasters as well as normal users to build structured web pages with customized page templates in a few seconds.

WebGen software is a powerful tool for creating large websites which need an easy and sophisticated automated tool which is provided by WebGen. Imagine how much work you will need if you have to create hundreds of pages. In a normal way you have to create each html file and you have to edit each page to put the header and don't forget the most difficult work which is editing the

JavaScript file to specify the menu items in the menu. However, with WebGen you just specify the structure and it will do much of the work automatically with much less time.

5.2 Critical Appraisal

The program will not work on less than Explorer 4 or Netscape 4, because of the limitation of the script of the menu that is written in JavaScript.

5.3 Future work

This project can be improved as a future work. WebGen can be implemented and improved to be used as a script installed in a server to give the site visitors the ability to create their own site in the hosting site e.g. `www.mysite.com/<any_name>`. He has to provide the structure of his site guided by a wizard and his own page will be created.

The WebGen class can be used in many applications. It can be used in forums to handle the threads and its replies and also in web directories.

6.0 References

*Books

- *C++ How to Program*
By H. M. Deitel & P. J. Deitel
Second Edition

*Journals

- *Guide to Writing a Project Report*
By W. L. Barnes
November 2000

Appendix A

User manual

The manual will consist of instructions on How to generate the HTML files with the specified requirements from the customer

- How to generate the HTML files with the specified requirements from the customer

A contributor needs first to provide the necessary files which are:

1. menutree.txt

It contains the definition of the menu hierarchy. Its description is in the specification sheet

2. webgen_template.txt

It contains the HTML for the header and the menu on each web page. This file can be changed by the user. All headers and footers are the same in all generated html files

3. webgen_default.txt

It contains the HTML for the default 'under construction' page

4. menu_array_style.txt

This defines the look and feel of the menus. The variable must be well commented to modify the style of the menu items easily.

The program will insert the menu items in the menu_array.js taken from the data structure in a way recognized from the JavaScript script.

The file must follow a specific syntax. For example:

```
style1=[                                // style1 is an array of properties
"navy",                                // Mouse Off Font Color
"ccccff",                              // Mouse Off Background Color
"ffebdc",                              // Mouse On Font Color
"ab0082",                              // Mouse On Background Color
"000000",                              // Menu Border Color
9,                                     // Font Size in pixels was 12
"normal",                             // Font Style (italic or normal)
"bold",                               // Font Weight (bold or normal)
"Verdana, Arial",                     // Font Name
4,                                    // Menu Item Padding
"arrow.gif",                          // Sub Menu Image (not needed -> blank)
,                                     // 3D Border & Separator bar
"66ffff",                             // 3D High Color
"000099",                             // 3D Low Color
"Purple",                             // Current Page Item Font Color
"pink",                               // Current Page Item Background Color
"arrowdn.gif",                        // Top Bar image
```

```

"fffff",           // Menu Header Font Color
"000099",          // Menu Header Background Color
]

```

**if any item is not needed leave it blank

These files should be put in a folder called webgen_data which is available already.

In this folder the user can find a program called GenerateMe.exe, so all what you have to do is press it (or run it from the command shell in DOS).

After that the system will generate the required HTML pages with a home page which contains the menu where you can browse these pages by index.htm located in the same level of webgen_data folder but they will be empty (i.e. it displays the header, menu, context line (path) and footer) if the user didn't provide the body of these pages. And in the body you will see the webgen_default 'under construction' page which you provided it to the system.

If the user want to put his own body in these pages he can place them in a *.txt extension in folders created by the user him self with the same names written in the menutree.txt and with the same specified names of files after replacing ' ' with '_' in the names which are written in the menutree.txt

And finally you are of curs want to see your pages so go to the main directory which will contain the whole folders under website folder and you will see there index.htm which allows you to brows your website

A general look of all folders and files is shown in Figure 1.

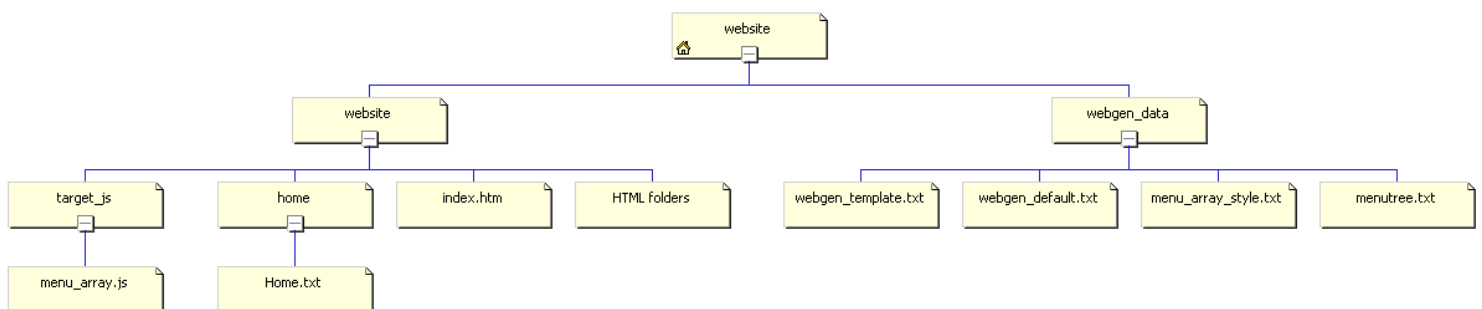


Figure 1 A general look of all folders and files