

**An Automatic Website Generator
(Requirements & Specification)
(Initial proposal)**

Names:

ID's:

Naser Ali Mohod

1239

Ahmad Abdulah

1208

Ibrahim Ali Saif Al-Jarwan

1224

Project Supervisor:

Prof. David Vernon

An Automatic Website Generator (Requirements & Specification)

1. The Requirements

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.

1.2 Performance & Behavior

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

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`

1.4 Interfaces

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 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 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:

<code>./website/target_js/menu_array.js</code>	-contains the JavaScript menu definitions
<code>./website/index.htm</code>	-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`

1.5 Reliability

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

2. The Specification

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 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*).

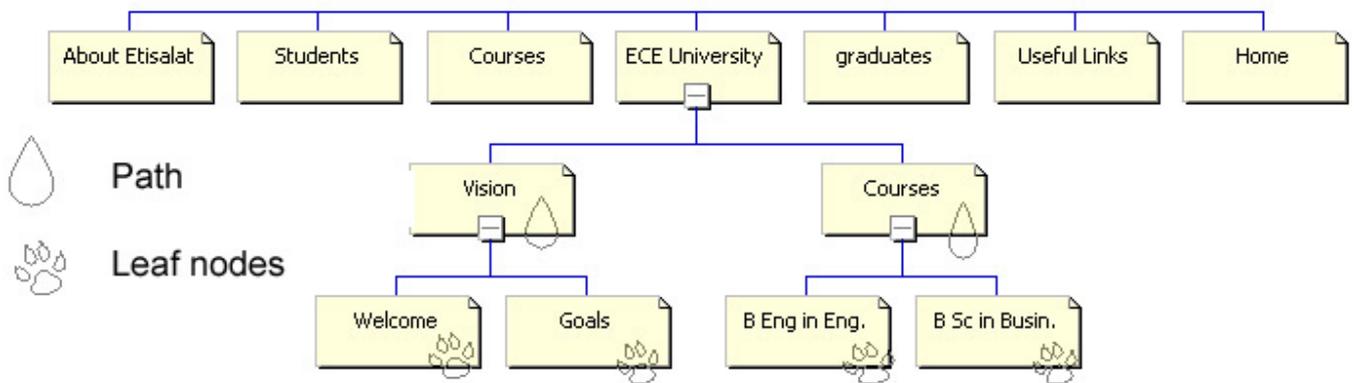
Specification of the syntax of `menutree.txt`

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.



```

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.

2.3 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
```

2.4 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
```

2.5 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

2.6 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. It's description is in the specification sheet
2. `webgen_template.txt`
It contains the HTML for the header and the menu on each web page
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 file must follow a specific syntax. For example:

```
"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
"ffffff", // 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.

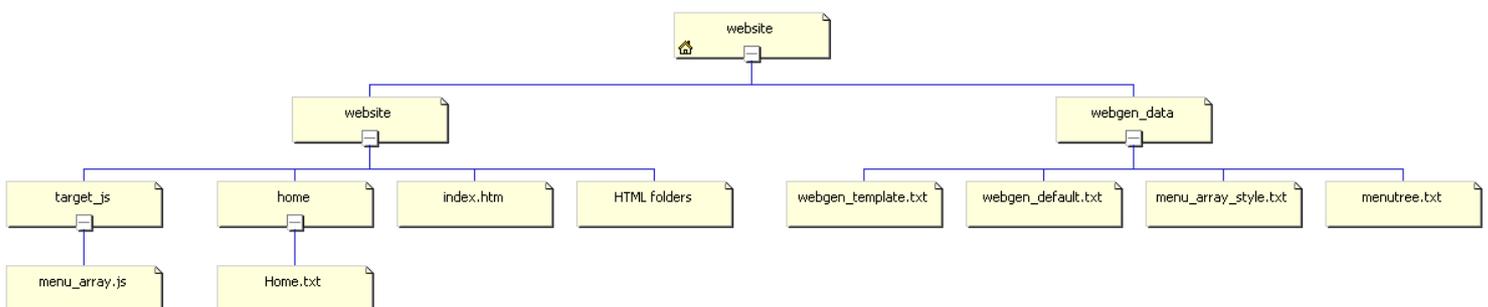
In this folder the user can find a program called GenerateMe.exe (for Linux operating system it is only GenerateMe).so all what you have to do is press it (or run it from the command shell in DOS or Linux environment).

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



2.7 Tasks:

- **Create a tree class that:**
 - Check the syntax of the menutree.txt (*Done by Naser / week 1-2*)
 - Displaying a message of errors. (*Done by Naser / week 1-2*) in case of:
 - An incorrect input structure (e.g. three tabs after a line contains one tab).
 - Using '/', '\', '.' characters.
 - Reads the menutree.txt (*Done by Ibrahim / week 1-2*)
 - 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 (*Done by Ibrahim / week 1-2*)
 - 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.

- **Create a MenuStyle function that creates a JavaScript file (i.e. menu-array.js):**
 - Read style's parameters which are read from menu_array_style.txt (*done by Ahmed / week 1-2*)
 - 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 (*done by Ahmed / week 3*)
 - 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.

- **Create a WebGen class 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 (*done by Naser / week 3*)
 - A member function that generates folders & files
 - Replace all unwanted characters (e.g. spaces) (*Done by Ibrahim / week 4*)
 - Generates the HTML files (*Done by Ahmed / week 4-6*)
 - 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. (*Done by Ibrahim / week 5-6*).

****All these tasks will start after the 1st major exam.**