

An Automatic Website Generator (Progress proposal)

Names:

ID's:

Naser Ali Mohod

1239

Ahmad Abdulah

1208

Ibrahim Ali Saif Al-Jarwan

1224

Project Supervisor:

Prof. David Vernon

1. 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 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.

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.

1.4. Input Files:

./webgen_data/menutree.txt	-contains the definition of the menu hierarchy
./webgen_data/webgen_template.txt	-contains the HTML for the header and the menu 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 menus

1.5. Output Files:

- HTML for all the leaf nodes in the menu tree.
- ./website/target_js/menu_array.js
 - Contains the JavaScript menu definitions.
- ./website/home/Home.txt
 - Contains the input HTML for the Home page.

1.6. Reliability

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

2. The Specification:

2.1. Objective

- To develop a program that will automatically convert a text based web specification file into a series of structured web pages.

2.2. 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.
- to add a page, all a contributor needs to do is to specify the menu item and provide the HTML file
- The menu tree for the entire website is contained in a file called menutree.txt

- Specification of the syntax of menutree.txt
 - 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.
- Rules governing the content of menutree.txt
 - 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.
 - 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 level of the menu and menu items in the menu tree, use a tab character for each level descended.

2. Tasks:

2.1. tree class:

- Check the syntax of the menutree.txt (*Done by Naser / week 1-2*)
Displaying a message of errors 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.

2.2. webgen class:

- Extract the leaf nodes and generates the HTML pages that include the header, menu and default 'under construction' style.
 - Create a MenuStyle function that creates a JavaScript file (i.e. menu-array.js):
 - Insert the menu items in JavaScript file (Done by Ibrahim)
 - Insert the style's parameters which are read from menu_array_style.txt in the JavaScript file (*done by Ahmed / week 1-2*)
 - A member function that forms the context line. (Done by Ahmad)

- A member function that reads webgen_templet.txt & webgen_default.txt (*done by Ahmad / week 3*)
- A member function that generates folders & files (Done by Ahmad)
 - Replace all unwanted characters (e.g. spaces).
 - 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.

3. Progress:

3.1. Tree class:

- Checking the syntax (80 %)
 - Checking for putting a tab after an item (20%)
- Reading menutree.txt and storing it in a data structure (100 %)
- Extracting the menu items (100 %)

3.2. webgen class:

- Forming the context line (0%)
- Inserting the menu items in JavaScript file (100 %)
- Insert the style's parameters in JavaScript file (0 %)
- Reading webgen_template.txt & webgen_default.txt (100%)
- Generates folders & files (95 %)
 - Putting html files under its author folder (5%)

Task	Status	Problems	Possible solutions
Checking the syntax of menutree.txt	Partial (80%)	-	-
Reading & Extracting menu items	Completed on time	Inefficiency	Converting 3D array into 2D array
Generating context line	Completed on time	-	-
Reading style's parameters	Not started	-	-
Generating HTML files	Partial (80%)	-	-
Generating "menu_array.js"	Completed on time	-	-

4. Time schedule

See the attached Time-schedule bar