BLM19202E/ BLM22212E / SEN22212E Data Structures Project #2
MINI DESKTOP SEARCH ENGINE

PROJECT REPORT

Project Name: SupremoSearch

Project Owners: İzzettin Furkan Özmen 2121251024
& Cihan Yılmaz 2121251036

BST Class

Features and Methods:

- The `BST` (Binary Search Tree) class is used for indexing words within documents by their frequencies. Each node (`Node`) contains a word and a linked list (`DocumentFrequencyNode`) of document names and their frequencies associated with that word.

- search(String word): Searches for the specified word in the BST and returns the document names and frequency information in a `MyLinkedList<String>`. If the word is not found, it adds "No occurrences found." to the list.

- insert(String word, String documentName): Inserts the given word into the BST. If the word already exists, it updates the frequency for the specified document. This method is recursive.

- traverse(String type, HashSet<String> ignoreWords): Traverses the tree in the specified order (Preorder, Inorder, Postorder) and returns a list of words excluding those in the ignoreWords set.

- searchRec(Node node, String word): This is a special helper method for the search operation. It recursively traverses the tree to find the node containing a specific word.

- insertRec(Node node, String word, String documentName):
This is a special helper method for the insertion operation. It recursively traverses the tree to add the new node to the appropriate position.

- inOrderHelper(Node node, MyLinkedlist<String> result, HashSet<String> ignoreWords): This is a special helper method for the inorder traversal type. It traverses the tree in left-root-right order and disregards the specified words.

- preOrderHelper(Node node, MyLinkedList<String> result, HashSet<String> ignoreWords): This is a special helper method for the preorder traversal type. It traverses the tree in root-left-right order and disregards the specified words.

- postOrderHelper(Node node, MyLinkedList<String> result, HashSet<String> ignoreWords): This is a special helper method for the postorder traversal type. It traverses the tree in left-right-root order and disregards the specified words.

- frequenciesToString(DocumentFrequencyNode frequencies):
This is a helper method that converts frequencies of a document into a string.
It creates a list of document names and their frequencies.

Node Class

Features and Methods:
- The `Node` class defines a node for the BST. Each node contains a word, left and right children (other `Node` objects), and a linked list of `DocumentFrequencyNode`.

- addOrUpdateFrequency(String documentName): Updates the frequency for a given document name or adds a new frequency node if it doesn't exist.

DocumentFrequencyNode Class

Features and Methods:
- Each `DocumentFrequencyNode` object holds a document name, frequency value, and a reference to the next node. This class is used to store the number of occurrences of a word in a specific document.

MyLinkedList Class

Features and Methods:
- A general-purpose linked list class. This list is used to store results from the BST and other data.

- add(T data): Adds a new node to the list.

- remove(T data): Removes the specified data from the list.

- printList(): Prints the list to the console.

- contains(T data): Checks if the specified data is in the list.

- join(String delimiter): Returns the list as a string joined by the specified delimiter.

DocumentFile Class

Features and Methods:
- Processes text or HTML documents from a specified file path. It extracts words from the document and adds them to the BST, excluding those in the `ignoreWords` list.

- processTextDocument(BST bst, HashSet<String> ignoreWords): Reads and processes a text document.

- processHtmlDocument(BST bst, HashSet<String> ignoreWords): Reads and processes an HTML document.

Each class and method effectively integrates data structures and document processing methods, setting up a structured approach for handling word indexing in documents. If you have any more questions about the functionality or need further details, feel free to ask.

MainMenu Jframe

Features and Methods:

- The `MainMenu` class is a JFrame derivative GUI component that represents the main menu of the application. It contains user interface components and provides functions for file operations, word search, visualization, and ignore list management.

- btnAddFileActionPerformed(java.awt.event.ActionEvent evt): Opens a file chooser with multi-file selection enabled when the "Add File" button is clicked and adds the selected files to the `selectedFiles` list.

- displaySelectedFileContents(): Visualizes the content of the selected file. It performs a traverse operation on the BST and displays the results in the `txtareaBST`.

- readIgnoreList(): Reads and returns the ignore list as a HashSet. This list is used to exclude certain words from processing.

- comboBoxFileActionPerformed(java.awt.event.ActionEvent evt) and **comboFixActionPerformed(java.awt.event.ActionEvent evt):** Visualizes relevant content when the file selection or sorting method is changed.

- btnIgnoreListActionPerformed(java.awt.event.ActionEvent evt): Allows the user to select an ignore list and adds the selected words to the `txtareaIgnore`.

- btnClearIgnoreActionPerformed(java.awt.event.ActionEvent evt): Clears the ignore list and regenerates and visualizes the BST.

- checkBoxAllActionPerformed(java.awt.event.ActionEvent evt): Shows the BST for all added files together or returns to visualizing a single file.

- txtSearchWordKeyPressed(java.awt.event.KeyEvent evt): Performs a word search and displays the results in the `txtareaSearchedFiles`.

These two classes enable file operations and visualization features, allowing the user to interactively process documents. Each method responds to user interactions, enhancing the application's functionality.

FileChooserFrame Jframe

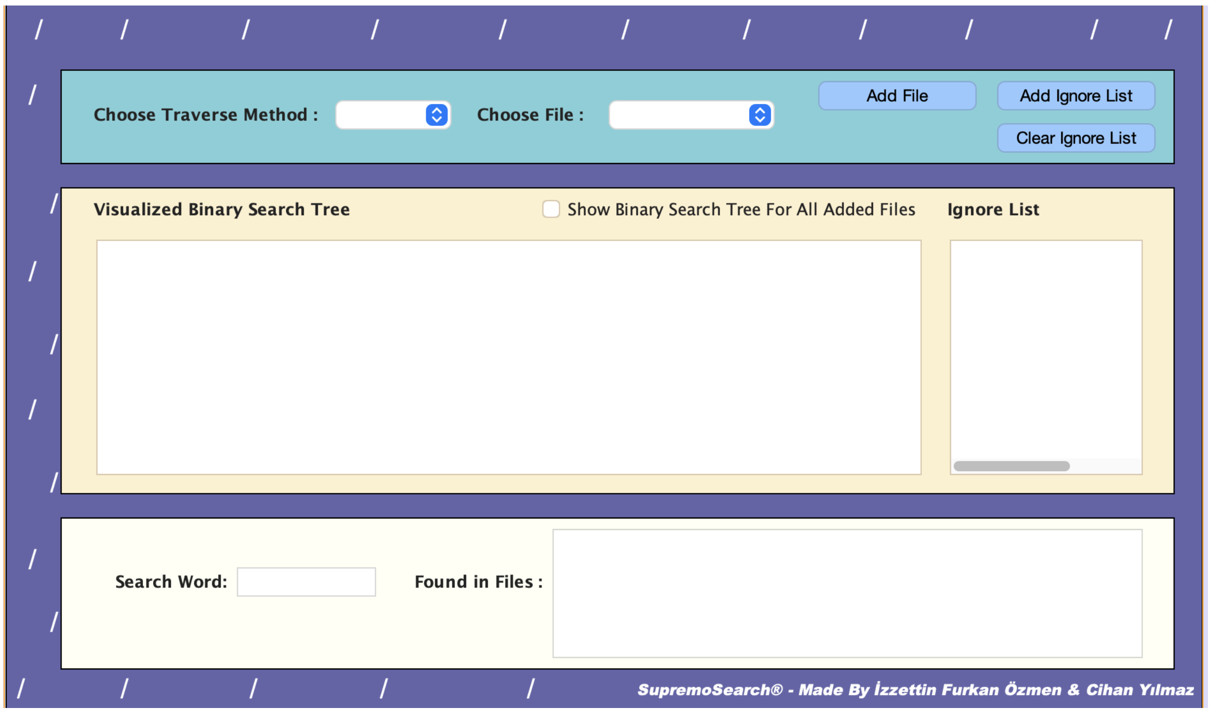
Features and Methods:

- The `FileChooserFrame` class is a JFrame derivative GUI component that allows the user to choose files. It references the main menu to process the selected files.

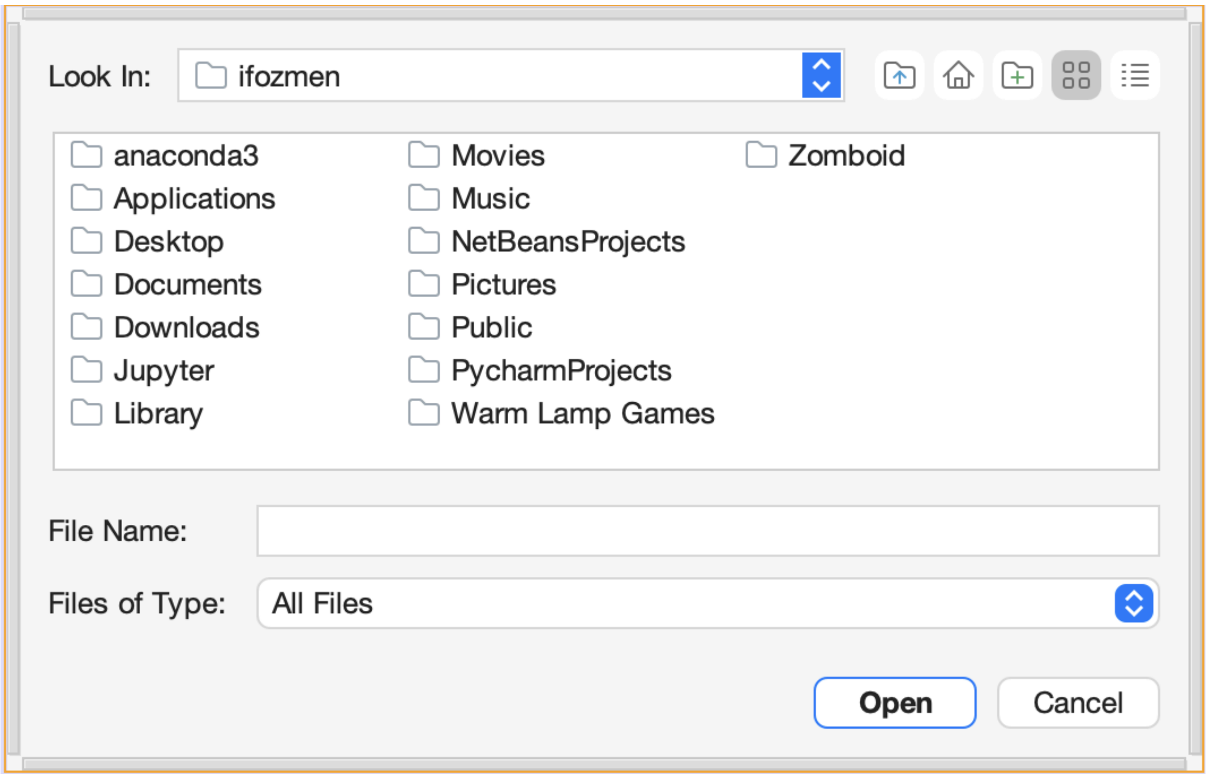
- jFileChooser1ActionPerformed(java.awt.event.ActionEvent evt): Triggered after an operation with the file chooser. This method is activated when the user selects or cancels file selection. If a file is selected, it checks if the file has been added before, adds it to the main menu if not, and creates a new `DocumentFile` and `BST`.

- isFileAdded(File file): Checks if a file has already been added to the ComboBox by its name. Returns `true` if the file name already exists.

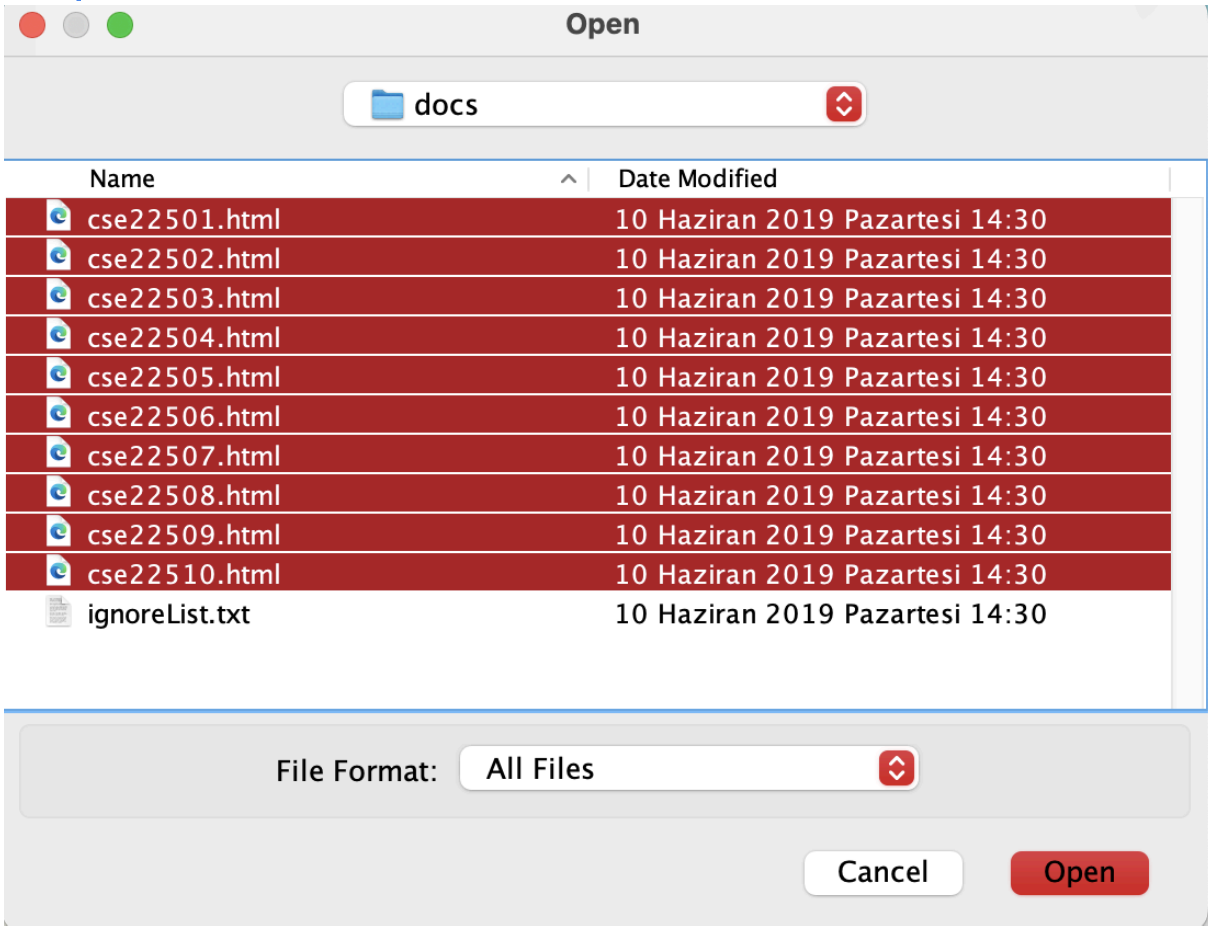
Program Interface


Choose Traverse Method :
Choose File :
Add File
Add Ignore List
Clear Ignore List
Visualized Binary Search Tree
Show Binary Search Tree For All Added Files
Ignore List
Search Word:
Found in Files :
SupremoSearch® - Made By İzzettin Furkan Özmen & Cihan Yılmaz

File Chooser Screen:

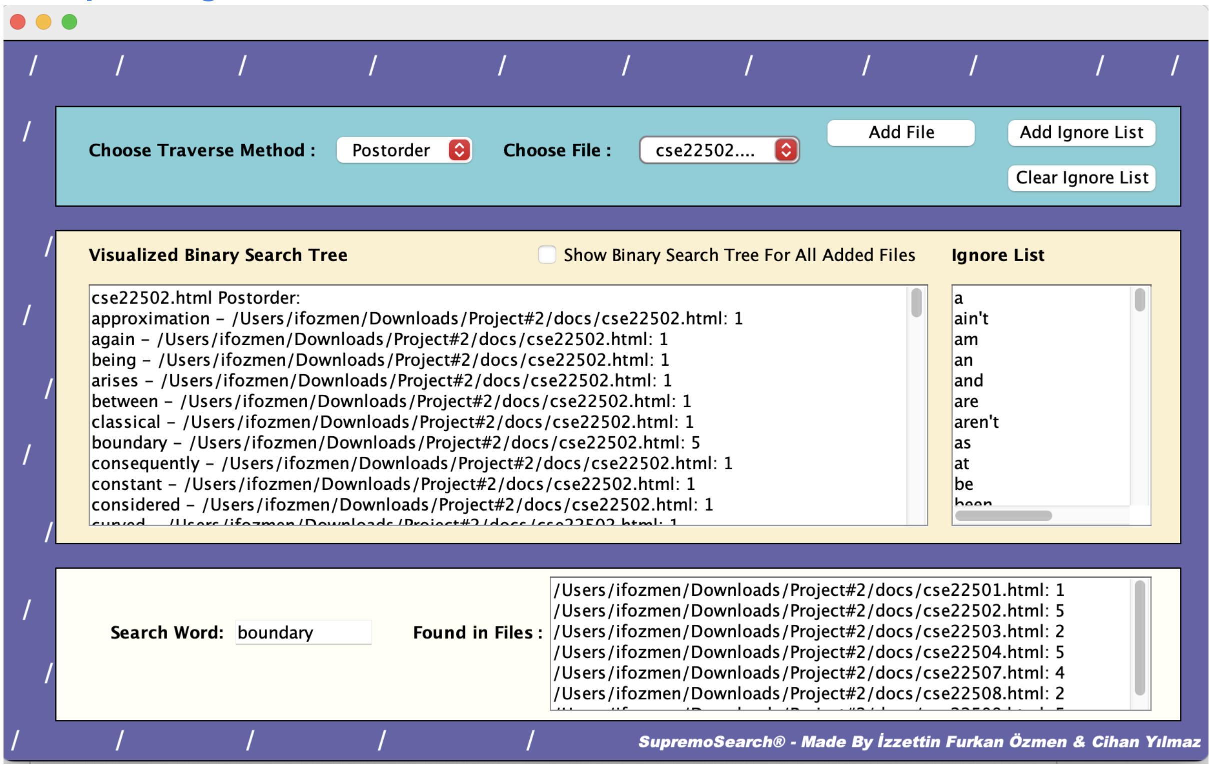


Multiple Document Choice Is Available:



Name	Date Modified
cse22501.html	10 Haziran 2019 Pazartesi 14:30
cse22502.html	10 Haziran 2019 Pazartesi 14:30
cse22503.html	10 Haziran 2019 Pazartesi 14:30
cse22504.html	10 Haziran 2019 Pazartesi 14:30
cse22505.html	10 Haziran 2019 Pazartesi 14:30
cse22506.html	10 Haziran 2019 Pazartesi 14:30
cse22507.html	10 Haziran 2019 Pazartesi 14:30
cse22508.html	10 Haziran 2019 Pazartesi 14:30
cse22509.html	10 Haziran 2019 Pazartesi 14:30
cse22510.html	10 Haziran 2019 Pazartesi 14:30
ignoreList.txt	10 Haziran 2019 Pazartesi 14:30

Example Usage:



Choose Traverse Method : Postorder

Choose File : cse22502....

Visualized Binary Search Tree

```
cse22502.html Postorder:
approximation – /Users/ifozmen/Downloads/Project#2/docs/cse22502.html: 1
again – /Users/ifozmen/Downloads/Project#2/docs/cse22502.html: 1
being – /Users/ifozmen/Downloads/Project#2/docs/cse22502.html: 1
arises – /Users/ifozmen/Downloads/Project#2/docs/cse22502.html: 1
between – /Users/ifozmen/Downloads/Project#2/docs/cse22502.html: 1
classical – /Users/ifozmen/Downloads/Project#2/docs/cse22502.html: 1
boundary – /Users/ifozmen/Downloads/Project#2/docs/cse22502.html: 5
consequently – /Users/ifozmen/Downloads/Project#2/docs/cse22502.html: 1
constant – /Users/ifozmen/Downloads/Project#2/docs/cse22502.html: 1
considered – /Users/ifozmen/Downloads/Project#2/docs/cse22502.html: 1
```

Ignore List

```
a
ain't
am
an
and
are
aren't
as
at
be
```

Search Word: boundary

Found in Files :

```
/Users/ifozmen/Downloads/Project#2/docs/cse22501.html: 1
/Users/ifozmen/Downloads/Project#2/docs/cse22502.html: 5
/Users/ifozmen/Downloads/Project#2/docs/cse22503.html: 2
/Users/ifozmen/Downloads/Project#2/docs/cse22504.html: 5
/Users/ifozmen/Downloads/Project#2/docs/cse22507.html: 4
/Users/ifozmen/Downloads/Project#2/docs/cse22508.html: 2
```

SupremoSearch® - Made By İzzettin Furkan Özmen & Cihan Yılmaz