



International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijirccce.com

Vol. 5, Issue 10, October 2017

PHP Syntactic Analyser and Comparer with Document Comparison

Manoj Sabnis¹, Nidhi Shah², Sukhada Khatu³, Rucha Guntoorkar⁴, Riya Lalwani⁵

Associate Professor, Department of Information Technology, Vivekanand Institute of Technology, Mumbai, India¹

Student, Department of Information Technology, Vivekanand Institute of Technology, Mumbai, India²

Student, Department of Information Technology, Vivekanand Institute of Technology, Mumbai, India³

Student, Department of Information Technology, Vivekanand Institute of Technology, Mumbai, India⁴

Student, Department of Information Technology, Vivekanand Institute of Technology, Mumbai, India⁵

ABSTRACT: This paper describes application that compares two text documents and code files side by side. It also provides hub for PHP compilation and comparison. It checks assignments based on keywords provided by professor.

KEYWORDS: lexer : Performs lexical analysis;

Semantic Analysis : Process of relating syntactic structures, from the levels of phrases, clauses, sentences and paragraphs to the level of the writing as a whole, to their language-independent meanings.

I. INTRODUCTION

In a company, usually there is a team working on same project. Real time applications have more than 1000 lines of code in each file and most of the files are used by more than 2 team mates. To see which changes are made by another team mate be it a text or code, we have to scan through the whole file to view the changes. Also, many times we want to compare the backup files and updated file, here if we sit and scan and compare both the files manually it would be very tedious job.

In our project we are going to suggest a solution for this by making an application which can solve above stated problem. Our domain of project is Application development.

The application will work as follows: First it will load two files which we want to compare side by side. It will check the extension to inspect whether the files are in same format. If it is a text file then it will only go for comparison and then it will highlight the changes. For a code file it will compare as well check for lexical, syntactic and semantic errors, then it will highlight the changes. For assignment checking it will take keywords and assignment as input and check for keywords in assignment, depending on number of matches found; assignment can be accepted or rejected.

II. LITERATURE SURVEY

Various tools that are available and solve this problem to some extent are:

- *Code Compare:*

Code Compare is a free tool designed to compare and merge differing files and folders. It provides with functions such as:

- A. Text Comparison and Merging.
- B. Structural Source Code Comparison.



International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

- **WinMerge:**
WinMerge is an Open Source differencing and merging tool for Windows. It provides with functions such as:
 - A. Compares both folders and files, presenting differences in a visual text format that is easy to understand and handle.
 - B. Determines what has changed between project versions, and then merging changes between versions.
- **Beyond Comparer:**
Beyond Compare is a paid data comparison utility. It provides with functions such as:
 - A. Compares files, the program is capable of doing side-by-side comparison of directories, FTP and SFTP directories, Dropbox directories, Amazon S3 directories, and archives.
- **Diff(command):**
In computing, the diff utility is a data comparison tool that calculates and displays the differences between two files. Diff is line-oriented rather than character-oriented.

III. PROPOSED METHODOLOGY AND DISCUSSION

Our System consist of three different modules:

1. Mere Comparison

In mere comparison we first choose two text files which we have to compare and application checks the extensions of both the files, then it compares both the files side by side. After comparison, it highlights and shows the differences in the box below for better understanding.

2. PHP Comparison

In PHP comparison two options are provided, first comparison and second compilation. In comparison, it will compare two files and highlight the difference. In compiler it will show lexical, syntactic and semantic error.

3. Assignment Checking

Based on keywords provided by the professor, assignment will be checked. If the matched keywords are greater than threshold number of keywords then assignment is accepted or rejected.

Algorithm:

- **PHP Compiler**
 - 1) Start
 - 2) Take .php file as input.
 - 3) Lexer: Convert the php content into tokens according to the lexical grammar defined in lexer file.
 - a. If token is not defined generate exception and display it to user.
 - b. Token that will not be used for parsing can be skipped.
 - c. Rest of the tokens will be sent to parser.
 - 4) Syntax analysis: Recover the structure described for syntax by using series of tokens from lexer.If the structure is not as per definition then it will be highlighted in program.
 - 5) Semantic analysis: Ensure program has well defined meaning i.e check for semantics of program. If the semantic of program is not as per definition then error will be displayed in the box below.
 - 6) Stop.
- **Working of Lexer**
 - 1) The source code is passed through a lexer, which converts the source code to a set of tokens.
 - 2) Tokens for PHP can be defined by representing keywords and operators in it. Names of function, identifier, variables, statements, class, function is defined by regular expression.



International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

- 3) A lexer has a unique name for each keyword.
 - 4) Lexical analyser builds a symbol table for the given PHP code. This is a table of all the identifiers (variable names, procedures, and constants) used in the program.
 - 5) When an identifier is first recognized by the analyser, it is inserted into the symbol table, along with information about its type, where it is to be stored, and so forth. This information is used in subsequent passes of the compiler.
- *Working of Syntax Analysis*
 - 1) Grammar rules are defined for PHP which correspond to EBNF notation.
 - 2) Parser for PHP follows LL grammar that is a top-down parser for a subset of context-free languages. It parses the input from Left to right, performing Leftmost derivation of the sentence.
 - 3) Grammar is observed for all the structural rules of a PHP to make logical sense to the computer.
 - 4) The parser reads in the tokens generated by the lexical analyzer and compares them to the set of grammar rules defined.
 - 5) If the program follows the rules of the PHP, then it is syntactically correct.
 - 6) When the parser encounters a mistake, highlights them in program and tries to continue.
 - 7) If the program is grammatically correct, the parser will call for semantic routines else it will show the message for termination.
 - *Working of Semantic Routines*
 - 1) Check to make sure that each series of tokens will be understood by the computer.
 - 2) Takes a series of tokens, called a production, and checks to see if it makes sense.
 - *Assignment Checking*
 - 1) Start.
 - 2) Take keywords for assignment from professor/user.
 - 3) Take threshold value for acceptance of assignment.
 - 4) Upload assignment to be checked.
 - 5) Find keywords in assignment.
 - 6) Calculate number of keywords found in assignment.
 - 7) If the number of keywords found in the assignment is greater to the threshold defined then assignment is accepted else it will be rejected.
 - 8) Stop.

IV. EXPERIMENTAL RESULTS

User Interface:

1) Home Page

It consists of three options for comparison, compilation and assignment checking. On selecting the option it will redirect to corresponding window.

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijirccce.com

Vol. 5, Issue 10, October 2017

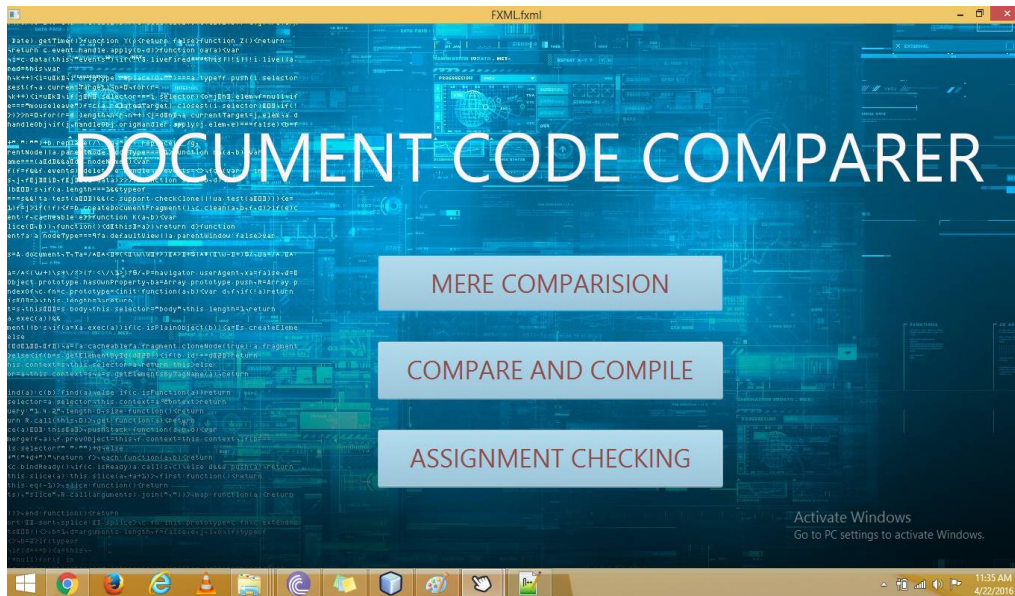


Figure 1- Home Page UI

- 2) *Mere Comparison*: Differences in both files will be highlighted. The changes will also be given in the message box.

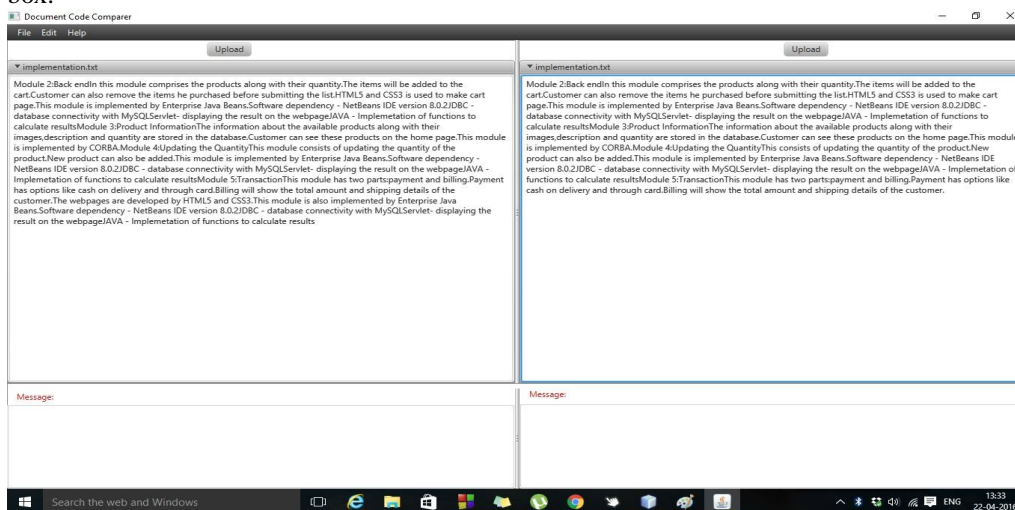


Figure 2- Document comparison with highlighted text

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijirccce.com

Vol. 5, Issue 10, October 2017

3) *PHP Compiler*: Syntax errors are highlighted. Semantic errors are shown in pop-up window.



Figure 3- Upload File

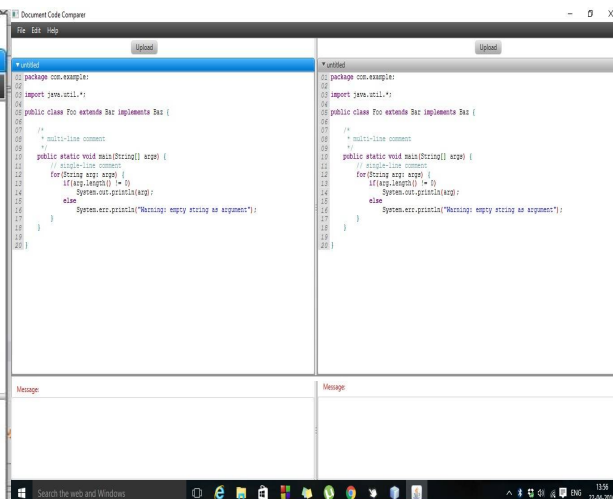


Figure 4- Compile and compare PHP code

4) *Assignment Checking*: Assignment accepted or rejected is displayed.

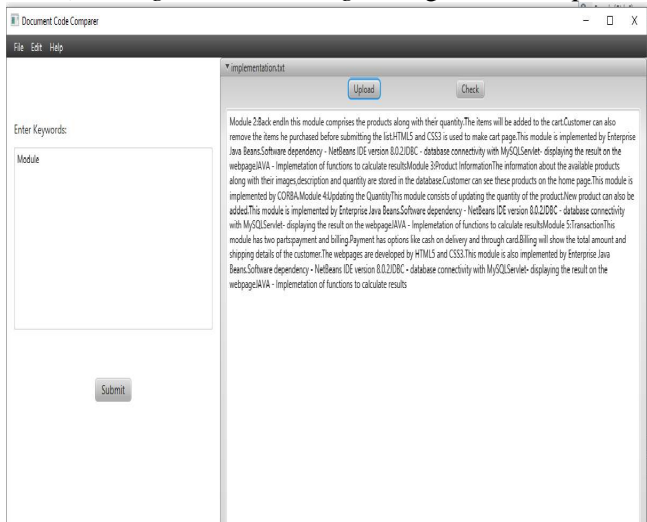


Figure 5- Assignment Upload

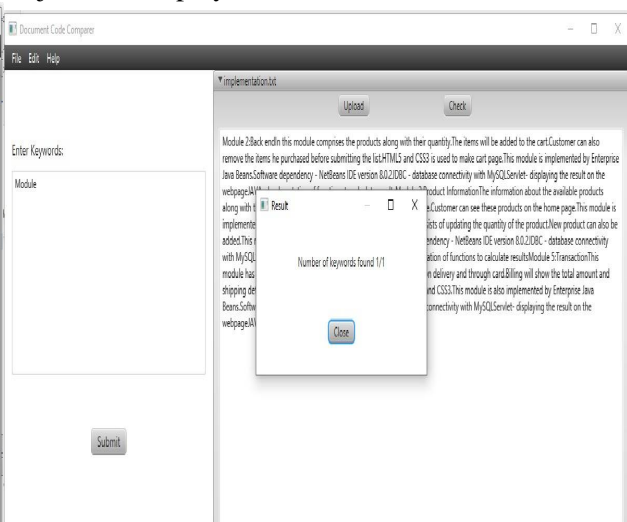


Figure 6- Output of assignment check

Future Scope:

- 1) Providing navigation in two files through changes shown.
- 2) Improving PHP module by checking for MySQL lexical and Semantic errors.
- 3) Develop full compiler which will have IR Generation, IR Optimization, Code Generation, Optimization.
- 4) For assignment checking, add synonyms for keywords.



ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijirccce.com

Vol. 5, Issue 10, October 2017

V. CONCLUSION

Document Code Comparer gives provision of compilation and comparison of PHP code in one application. After comparing two files and editing the changes the user will get the improvised version of the file. The compiled PHP code will be free of syntactic, lexical and semantic error. The assignment checking will be done on the basis of keywords and the final result will be displayed with the help of keyword matching.

REFERENCES

1. <https://www.cs.tcd.ie/publications/tech-reports/reports.07/TCD-CS-2007-47.pdf>
2. <https://platform.netbeans.org/tutorials/nbm-javacc-parser.html>
3. <https://javacc.java.net/doc/docindex.html>
4. <https://class.coursera.org/compilers/lecture/preview>