Assignment 8

Roll No 71 T14Batch

Aim : WAP to implement concept of REPL in commandline.

LO Mapped : LO 6

Theory :

The node:repl module exports the repl.REPLServer class. While running, instances of repl.REPLServer will accept individual lines of user input, evaluate those according to a user-defined evaluation function, then output the result. Input and output may be from stdin and stdout, respectively, or may be connected to any Node.js stream.

Instances of repl.REPLServer support automatic completion of inputs, completion preview, simplistic Emacs-style line editing, multi-line inputs, ZSH-like reverse-i-search, ZSH-like substring-based history search, ANSI-styled output, saving and restoring current REPL session state, error recovery, and customizable evaluation functions. Terminals that do not support ANSI styles and Emacs-style line editing automatically fall back to a limited feature set.

Commands and special keys

.break: When in the process of inputting a multi-line expression, enter the .break command (or press Ctrl+C) to abort further input or processing of that expression.

.clear: Resets the REPL context to an empty object and clears any multi-line expression being input.

.**exit:** Close the I/O stream, causing the REPL to exit.

.**help**: Show this list of special commands.

.save: Save the current REPL session to a file: > .save ./file/to/save.js **.load**:

Load a file into the current REPL session. > .load ./file/to/load.js .**editor**:

Enter editor mode (Ctrl+D to finish, Ctrl+C to cancel).

By default, all instances of repl.REPLServer use an evaluation function that evaluates JavaScript expressions and provides access to Node.js built-in modules. This default behavior can be overridden by passing in an alternative evaluation function when the repl.REPLServer instance is created.

Conclusion :

Overviewed REPL in node.js and demonstrated various methods related to it.

LO6 was achieved.

Created a calculator in for commandline REPL.