

NAME : PREM SAH

PRN : 123B1B234

DIV : D

ASSIGNMENT NO.7

Implement a browser history management system using a stack data structure to track the user's browsing history. The system should support the following functionalities:

- a. Add visited page
- b. Navigate back
- c. View current page
- d. Check if history is empty or not

CODE :

```
#include <iostream>
#include <stack>
#include <string>
using namespace std;

class BrowserHistory {
private:
    stack<string> history;
    string currentPage;

public:
    // Add a visited page
    void visitPage(const string& url) {
        if (!currentPage.empty()) {
            history.push(currentPage); // Push the current page to history
        }
        currentPage = url;
        cout << "Visited: " << currentPage << endl;
    }

    // Navigate back to the previous page
    void goBack() {
        if (!history.empty()) {
            currentPage = history.top(); // Set the previous page as current
            history.pop();
            cout << "Went back to: " << currentPage << endl;
        } else {
            cout << "No previous pages in history!" << endl;
        }
    }

    // View the current page
    void viewCurrentPage() {
        if (currentPage.empty()) {
```

```

        cout << "No page currently open!" << endl;
    } else {
        cout << "Current Page: " << currentPage << endl;
    }
}

// Check if history is empty
bool isHistoryEmpty() {
    return history.empty();
}

};

int main() {
    BrowserHistory browser;

    browser.visitPage("https://example.com");
    browser.visitPage("https://google.com");
    browser.visitPage("https://stackoverflow.com");

    browser.viewCurrentPage(); // Displays: Current Page: https://stackoverflow.com

    browser.goBack();          // Goes back to https://google.com
    browser.viewCurrentPage(); // Displays: Current Page: https://google.com

    browser.goBack();          // Goes back to https://example.com
    browser.viewCurrentPage(); // Displays: Current Page: https://example.com

    if (browser.isHistoryEmpty()) {
        cout << "History is empty!" << endl;
    } else {
        cout << "History is not empty!" << endl;
    }

    return 0;
}

```

Output :

```

Visited: https://example.com
Visited: https://google.com
Visited: https://stackoverflow.com
Current Page: https://stackoverflow.com
Went back to: https://google.com
Current Page: https://google.com
Went back to: https://example.com
Current Page: https://example.com
History is empty!

```

main.cpp

```
1 #include <iostream>
2 #include <stack>
3 #include <string>
4 using namespace std;
5 class BrowserHistory {
6 private:
7     stack<string> history;
8     string currentPage;
9 public:
10    // Add a visited page
11    void visitPage(const string& url) {
12        if (!currentPage.empty()) {
13            history.push(currentPage); // Push the current page to history
14        }
15        currentPage = url;
16        cout << "Visited: " << currentPage << endl;
17    }
18    void goBack() {
19        if (!history.empty()) {
20            currentPage = history.top(); // Set the previous page as current
21            history.pop();
22            cout << "Went back to: " << currentPage << endl;
23        } else {
24            cout << "No previous pages in history!" << endl;
25        }
26    }
27    void viewCurrentPage() {
28        if (currentPage.empty()) {
29            cout << "No page currently open!" << endl;
30        } else {
31            cout << "Current Page: " << currentPage << endl;
32        }
33    }
34    bool isHistoryEmpty() {
35        return history.empty();
36    }
37 };
38 int main() {
39     BrowserHistory browser;
40
41     browser.visitPage("https://example.com");
42     browser.visitPage("https://google.com");
43     browser.visitPage("https://stackoverflow.com");
44
45     browser.viewCurrentPage(); // Displays: Current Page: https://stackoverflow.com
46
47     browser.goBack(); // Goes back to https://google.com
48     browser.viewCurrentPage(); // Displays: Current Page: https://google.com
49
50     browser.goBack(); // Goes back to https://example.com
51     browser.viewCurrentPage(); // Displays: Current Page: https://example.com
52
53     if (browser.isHistoryEmpty()) {
54         cout << "History is empty!" << endl;
55     } else {
56         cout << "History is not empty!" << endl;
57     }
58
59     return 0;
60 }
61
```

Output

```
/tmp/ckdzaCH7cK.o
Visited: https://example.com
Visited: https://google.com
Visited: https://stackoverflow.com
Current Page: https://stackoverflow.com
Went back to: https://google.com
Current Page: https://google.com
Went back to: https://example.com
Current Page: https://example.com
History is empty!
```

=== Code Execution Successful ===