PIMPRI CHINCHWAD EDUCATION TRUST's.

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

(An Autonomous Institute)

S.Y. B. TECH **Year:** 2024 – 25 **Semester:** I

Name: Abishek Joshi **PRN: 123B1B150**

Department: Computer Engineering **Division:** C (C1)

Course: Data Structures Laboratory **Course Code:** BCE23PC02

Date:

Assignment – 7

- **Aim:**

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
Assume no upper bound on number of pages visited

- **Source Code :**

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

class History {
   string sites[100];
   string nextSites[100];
   int top1;
   int top2;
public:
   History() {
      top1 = -1;
      top2 = -1;
   }
```

```
void visitedPage(string link) {
    if (top1 < 100 - 1) {
        sites[++top1] = link;
    } else {
        cout << "History is full. Cannot add more pages." << endl;
    }
}
void navigateBack() {
    if (top1 < 0) {
        cout << "No pages to navigate back." << endl;
        return;
    }
    nextSites[++top2] = sites[top1];
    cout << "Navigated back to: " << sites[--top1] << endl;
    cout<<"--------------------"<<endl;
}
void navigateForward() {
    if (top2 < 0) {
        cout << "No pages to navigate forward." << endl;
        return;
    }
    sites[++top1] = nextSites[top2--];
    cout << "Navigated forward to: " << sites[top1] << endl;
    cout<<"--------------------"<<endl;
}

void viewCurrentPage() {
    if (top1 >= 0) {
        cout << "Current Page: " << sites[top1] << endl;
    } else {
        cout << "No current page." << endl;
    }
    cout<<"--------------------"<<endl;
}
```

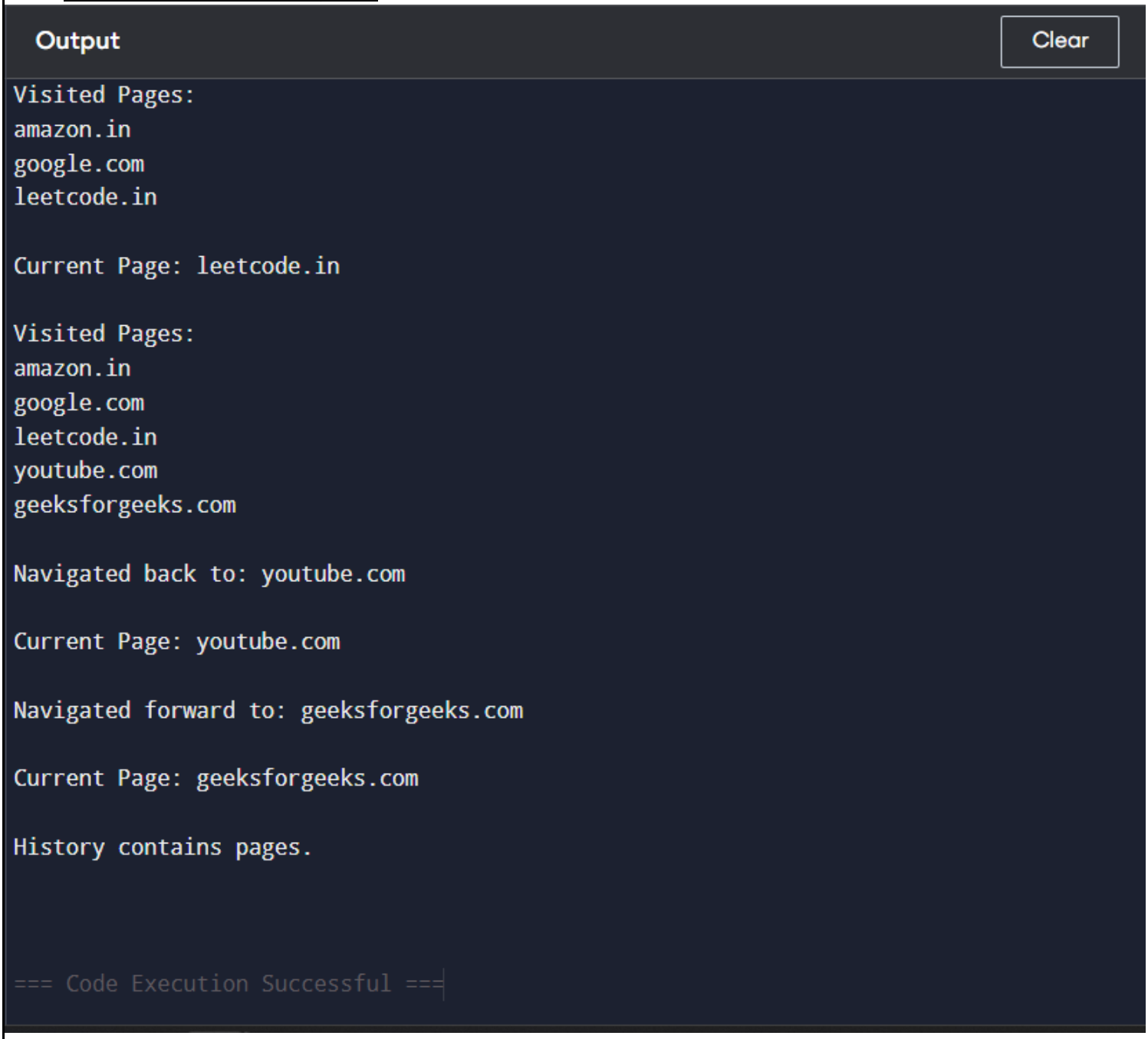
```
    void checkHistory() {
        if (top1 < 0) {
            cout << "History is empty." << endl;

        } else {
            cout << "History contains pages." << endl;
        }
        cout<<"------------------"<<endl;
    }

    void printHistory() {
        cout << "Visited Pages:" << endl;
        for (int i = 0; i <= top1; i++) {
            cout << sites[i] << endl;
        }
        cout<<"------------------"<<endl;
    }
};

int main() {
    History h1;
    h1.visitedPage("amazon.in");
    h1.visitedPage("google.com");
    h1.visitedPage("leetcode.in");
    h1.printHistory();
    h1.viewCurrentPage();
    h1.visitedPage("youtube.com");
    h1.visitedPage("geeksforgeeks.com");
    h1.printHistory();
    h1.navigateBack();
    h1.viewCurrentPage();
    h1.navigateForward();
    h1.viewCurrentPage();
    h1.checkHistory();
}
```

- **Screen Shot of Output :**



- **Conclusion:**

  Hence, we studied about application of arrays to demonstrate stack operations for adding pages and navigating history to implement browser history management system effectively.