PIMPRI CHINCHWAD EDUCATION TRUST's.

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

(An Autonomous Institute)

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

Name: Sonawane Prachi Mahendra. **PRN:** 124B2B018

Department : Computer Engineering **Division:** B

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

Date: 25/09/24

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:

  1. Add visited page
  2. Navigate back
  3. View current page
  4. Check if history is empty or not Assume no upper bound on number of pages visited

- **Source Code:**

```
#include <iostream> #include

<string>

class SimpleBrowserHistory {

private:

    std::string back[100];

    std::string forward[100]; int

    backIndex = -1;
```

```
    int forwardIndex = -1;

    std::string currentPage;

public:
        void addVisitedPage(const std::string& page) {

        if (!currentPage.empty()) {

            back[++backIndex] = currentPage;
        }
        currentPage = page;
        forwardIndex = -1;
    }
        std::string navigateBack() {
        if (backIndex >= 0) {
            forward[++forwardIndex] = currentPage;
            currentPage = back[backIndex--];
            return currentPage;
        }
        return "No previous page.";
    }
        std::string navigateForward() {

        if (forwardIndex >= 0) {
            back[++backIndex] = currentPage;
            currentPage = forward[forwardIndex--];
```

```
            return currentPage;
        }
        return "No forward page.";
    }
    std::string viewCurrentPage() const {
        return currentPage.empty() ? "No current page." : currentPage;
    }
    bool isHistoryEmpty() const {
        return backIndex == -1 && forwardIndex == -1 && currentPage.empty();
    }
};
void displayMenu() {
    std::cout << "\nBrowser History Management System\n"; std::cout <<
    "1. Add Visited Page\n";
    std::cout << "2. Navigate Back\n"; std::cout << "3.
    Navigate Forward\n"; std::cout << "4. View
    Current Page\n";
    std::cout << "5. Check if History is Empty\n"; std::cout
    << "6. Exit\n";
    std::cout << "Enter your choice: ";
}
    int main() { SimpleBrowserHistory
    browser; int choice;
```

```
    std::string page;
    do {
        displayMenu(); std::cin >>
        choice;

            switch (choice) {

            case 1:
                std::cout << "Enter page URL: "; std::cin
                >> page; browser.addVisitedPage(page);
                break;
            case 2:
                std::cout << "Navigating back to: " << browser.navigateBack() << std::endl; break;
            case 3:
 std::endl;
std::cout << "Navigating forward to: " << browser.navigateForward() <<
break;
            case 4:
                std::cout << "Current Page: " << browser.viewCurrentPage() << std::endl; break;
            case 5:
 std::cout << "Is history empty? " << (browser.isHistoryEmpty() ? "Yes" : "No") << std::endl;
            break; case 6:
                std::cout << "Exiting...\n"; break;
            default:
```

```
            std::cout << "Invalid choice. Please try again.\n"; break; }

    } while (choice != 6);

    return 0;

}
```

- **Screen shots of Output:**

1.

```
Output
/tmp/76bvRBGiFX.o

Browser History Management System
1. Add Visited Page
2. Navigate Back
3. Navigate Forward
4. View Current Page
5. Check if History is Empty
6. Exit
Enter your choice: 1
Enter page URL: hello

Browser History Management System3
1. Add Visited Page
2. Navigate Back
3. Navigate Forward
4. View Current Page
5. Check if History is Empty
6. Exit
Enter your choice: 1
Enter page URL: hii
```

```
Output
Browser History Management System
1. Add Visited Page
2. Navigate Back
3. Navigate Forward
4. View Current Page
5. Check if History is Empty
6. Exit
Enter your choice: 4
Current Page: hii

Browser History Management System
1. Add Visited Page
2. Navigate Back
3. Navigate Forward
4. View Current Page
5. Check if History is Empty
6. Exit
Enter your choice: 2
Navigating back to: hello
```

```
Browser History Management System
1. Add Visited Page
2. Navigate Back
3. Navigate Forward
4. View Current Page
5. Check if History is Empty
6. Exit
Enter your choice:
3
Navigating forward to: hii

Browser History Management System
1. Add Visited Page
2. Navigate Back
3. Navigate Forward
4. View Current Page
5. Check if History is Empty
6. Exit
Enter your choice: 6\6
Exiting...
```

- **Conclusion:**

  Hence, we studied about the stack data structure and its operations.