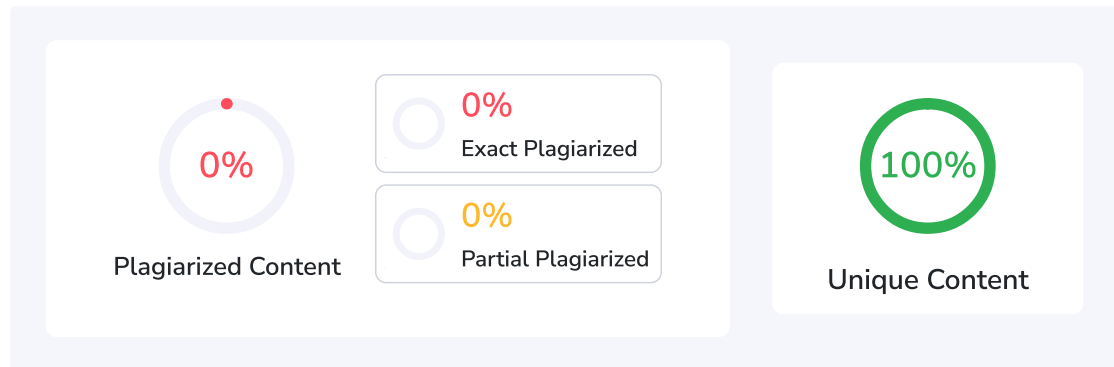


Plagiarism Scan Report By SmallSEOTools

Report Generated on: Oct 29,2024



Total Words: 659

Total Characters: 4062

Plagiarized Sentences: 0

Unique Sentences: 3 (100%)

Content Checked for Plagiarism

```
#include
#include
using namespace std;

class Email
{
private:
string sender;
string receiver;
string subject;
string body;

public:
// Default constructor
Email() {}

// Parameterized constructor
Email(string s, string r, string sub, string b)
{
sender = s;
receiver = r;
subject = sub;
body = b;
}

// Getter methods to access private data
string getSender() {
return sender;
}
string getReceiver() {
return receiver;
}
string getSubject() {
return subject;
}
string getBody() {
```

```

return body;
}

// Display function to print email details
void display()
{
cout << "From: " << sender << "\nTo: " << receiver
<< "\nSubject: " << subject << "\nBody: " << body << endl;
}
};

class EmailQueue
{
private:
static const int SIZE = 100; // Maximum size of the queue
Email queue[SIZE]; // Array to store emails
int front; // Points to the first element in the queue
int rear; // Points to the last element in the queue
int count; // To keep track of number of emails in the queue

public:
// Constructor initializes the queue
EmailQueue()
{
front = 0;
rear = -1;
count = 0; // Initialize count to 0 as the queue is empty initially
}

// Check if the queue is full
bool isFull()
{
return count == SIZE; // Queue is full if count reaches SIZE
}

// Check if the queue is empty
bool isEmpty()
{
return count == 0; // Queue is empty if count is 0
}

// Add an email to the queue
void enqueue(Email email)
{
if (isFull())
{
cout << "Queue is full. Cannot add more emails." << endl;
}
else
{
rear = (rear + 1) % SIZE; // Move rear forward in a circular manner
queue[rear] = email; // Add the email at the new rear position
count++; // Increase the email count
cout << "Email added to the queue successfully!" << endl;
}
}

// Remove and process an email from the queue
void dequeue()

```

```

{
if (isEmpty())
{
cout << "Queue is empty. No emails to process." << endl;
}
else
{
cout << "Processing email:\n";
queue[front].display(); // Display the email being processed

front = (front + 1) % SIZE; // Move front forward in a circular manner
count--; // Decrease the email count
}
}

// Display all emails in the queue
void displayAllEmails()
{
if (isEmpty())
{
cout << "No emails in the queue." << endl;
}
else
{
cout << "Emails in the queue:\n";
for (int i = 0; i < count; i++)
{
int index = (front + i) % SIZE;
cout << (i + 1) << ". ";
queue[index].display();
}
}
}

// Function to display the menu
void displayMenu()
{
cout << "\nMenu:\n";
cout << "1. Add an Email (Enqueue)\n";
cout << "2. Process an Email (Dequeue)\n";
cout << "3. Show All Emails\n";
cout << "4. Exit\n";
cout << "Enter your choice: ";
}

int main()
{
EmailQueue emailQueue; // Create an EmailQueue object
int choice; // To store user input

while (true) // Infinite loop to keep the menu active
{
displayMenu(); // Display the menu
cin >> choice;

switch (choice)
{
case 1:

```

```

{
string sender, receiver, subject, body;

// Taking email details from the user
cout << "Enter sender email: ";
cin >> sender;
cout << "Enter receiver email: ";
cin >> receiver;
cin.ignore(); // Clear input buffer before taking multi-line input
cout << "Enter subject: ";
getline(cin, subject);
cout << "Enter body: ";
getline(cin, body);

// Creating an Email object
Email email(sender, receiver, subject, body);
emailQueue.enqueue(email); // Adding email to the queue
break;
}
case 2:
emailQueue.dequeue(); // Process and remove the first email in the queue
break;

case 3:
emailQueue.displayAllEmails(); // Show all emails in the queue
break;

case 4:
cout << "Exiting the program." << endl;
return 0; // Exit the program

default:
cout << "Invalid choice. Please try again." << endl;
}
}

return 0;
}

```



No Plagiarism Found