SAVEETHA
SCHOOL OF ENGINEERING

Name of the Student : .....

Register Number : .....

Department : .....

Semestor : .....

Subject : .....

LABORATORY RECORD NOTE BOOK

SAVEETHA
INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES
(Declared as Deemed to be University under Section 3 of UGC Act 1956)

Saveetha Nagar, Thandalam, Chennai - 602 105,
Tamil Nadu, India. Phone : +91 44 6672 6672
Website : www.saveetha.com,
Email : principal.sse@saveetha.com

SAVEETHA
SCHOOL OF ENGINEERING

Department Of .....

LABORATORY RECORD NOTE BOOK

20 - 20

This is certify that this is a bonafide record of that work done by

Mr. / Ms .....Register Number .....

of the year ..... B.E / B.Tech., Department of .....

in the ..... Laboratory in the ..... Semester

University Examination held on .....

Staff in - Charge

Head of the Department

Internal Examiner

External Examiner

SAVEETHA
INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES
(Declared as Deemed to be University under Section 3 of UGC Act 1956)

Saveetha Nagar, Thandalam, Chennai - 602 105,
Tamil Nadu, India. Phone : +91 44 6672 6672
Website : www.saveetha.com,
Email : principal.sse@saveetha.com

SAVEETHA SCHOOL OF ENGINEERING
SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES
COMPUTER SCIENCE AND ENGINEERING PROGRAMME

Engineer to Excel

CSA38 – SOFTWARE QUALITY ASSURANCE AND MANAGEMENT

LIST OF LAB EXPERIMENTS

EXP. NO.	NAME OF THE EXPERIMENT
1	**Identification of Functional Requirements**
1 -A	Online Voting is simple, attractive, and easy to use. It reduces manual efforts and the bulk of information can be handled easily. The system should provide concrete security features likecreating users and assigning privileges to users of the system. System should be capable of keeping track of all the detailed descriptions of the client and the whole details of services offered by the client organization. Document suitable problem description with functional requirements specification for Online Voting System.
1 -B	Library Management System gives us complete information about the library and the daily transactions done in a Library. We need to maintain the record of books and retrieve the details of books available in the library which mainly focuses on basic operations in a library like adding new member, new books, and up new information, searching books and members and facility to borrow and return books. It features a familiar and well thought-out, an attractive user interface, combined with strong searching, insertion, and reporting capabilities. The report generation facility of library system helps to get a good idea of which are borrowed by the members, makes users possible to generate hard copy. Document suitable problem description with functional requirements specification for Library Management System.
1 -C	Online Shopping system aims to develop online shopping for customers with the goal that it is very easy to shop for your loved things from an extensive number of online shopping sites available on the web. With the help of this, we can carry out online shopping from our homes. Here is no compelling reason to go to the crowded stores or shopping centers during festival seasons. Upon successful login, the customers can purchase a wide range of things such as mobiles, books, apparel, jewelry, infant care, gifts, tools, etc. can be dispatched using an online shopping system. Document suitable problem descriptions with functional requirements specifications for the Online Shopping system.
2	**Identification of Non Functional Requirements**
2 -A	Online Voting System

2 -B	Library Management System
2 -C	Online Shopping System
3	UML Diagram for Online Voting System
4	UML Diagram for Library Management System
5	UML Diagram for Online Shopping System
6	UML Diagram for Food Ordering System
7	UML Diagram for Railway Reservation System
8	UML Diagram for Airport Management System
9	Using Raptor – Draw the flowchart to find whether the given number is prime or not
10	Draw and validate the flowchart to compute the quotient and remainder
11	Using Raptor – Draw the flowchart to find whether the given input is a palindrome or not
12	Draw and validate the flowchart to count and print vowels in given word.
13	Draw and validate the flowchart to calculate Fibonacci series.
14	Using Raptor – Draw the flowchart to display the positions of numbers
15	Using Raptor – Draw and validate the flowchart to swap two characters.
16	Using Raptor – Draw the flowchart to check whether the given number is an Armstrong Number or not.
17	Draw and validate the flowchart to display the prime numbers from 1 – 100
18	Draw and validate the flowchart to swap two numbers
19	Using Raptor – Draw the flowchart to display the length of the string
20	Automation Testing using Selenium tool for Student login of their Institution (Ex – ARMS)
21	Automation Testing using Selenium tool for Online Food Application.

22	Automation Testing using Selenium tool for Ticket Reservation System (IRCTC)
23	Automation Testing using Selenium tool for Shopping Domain
24	Automation Testing using Selenium tool for Opening any Search Engine (EX-google.com, yahoo.com)
25	Automation Testing using Selenium tool for any Social Media Platform
26	Using Junit Testing Check whether the given string is printed or not
27	Using Junit Testing Check Whether the given string is getting Reversed or not
28	Using Junit Testing Check Whether the given Username is Valid or not
29	Using Junit Testing Check Whether the Calculated simple interest is valid or not
30	Using Junit Testing check whether the given number is palindrome or not

Software Requirements: Umbrello, Raptor, Eclipse (Junit, Selenium packages for Testing), java JDK-1.8

SAVEETHA
INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES
(Declared as Deemed to be University under Section 3 of UGC Act 1956)

SAVEETHA SCHOOL OF ENGINEERING
DEPARTMENT OF CSE

COURSE CODE/NAME: CSA3821 - Software Quality Assurance and Management for Process Analysis

COURSE FACULTY: Dr. J. Velmurgan

STUDENT NAME: I. SELMEYA

REG NO : 192121137

EXP NO: 1A FUNCTIONAL REQUIREMENTS FOR ONLINE VOTING SYSTEM

AIM:

To perform analysis of the given problem statement for listing the functional requirements of the online voting system

PROBLEM STATEMENT:

i) Current offline voting system status:

Today, only about 1 percent of the population votes at polling places on hand counted paper ballots, but this figure is misleading. There are many elections conducted on optical mark-sense ballots that are actually hand counted, and many jurisdictions that use lever voting machines process absentee ballots by hand.

ii) New technology:

Today, an increasing fraction of the direct-recording electronic voting machines on the market include provisions to network all of the voting machines in one polling place. This allows each machine to store vote totals in the memory of the others, and at the close of the polls, it allows a single report for the entire precinct to be created instead of one report for each machine.

ONLINE VOTING SYSTEM:

Online voting systems protect the integrity of the vote by preventing voters from being able to vote multiple times. As a digital platform, they eliminate the need to gather in-person, cast votes using paper orby any other means. Online voting is a digital platform which allows the groups to conduct voting securely, High quality systems balance ballot security, accessibility, and overall requirements.

FUNCTIONAL REQUIREMENTS:

- **Admin:** This use case is the starting point for any interaction with the information system. It is a general use case specialized for organizers, i.e. "Authorize organizer" and users i.e. "Authorize user".

- **Manage Election Districts:** Create, view and modify different sets of election districts for one or more election procedures.
- **User-Interface:** The system shall provide an easy-to-use user-interface. Also, it shall not disadvantage any candidate while displaying the choices (e.g., by requiring the user to scroll down to see the last few choices).
- **Manage electronic units:** Import, insert, view and modify electors for one or more election procedures.
- **Support for Disabled Voters:** The system shall cater to the needs of physically challenged voters (e.g. blind voters).
- **Eligibility:** Only authorized voters, who are registered, should be able to vote.
- **Voter Confirmation:** The voter shall be able to confirm clearly how his vote is being cast, and shall be given a chance to modify his vote before he commits it.
- **Provisional Ballots:** The voter shall be able to vote with a provisional (electronic) ballot if he has some registration problems, which could be counted if verified by the authorities later.

Documentation and Assurance: The design, implementation, and testing procedures must be well documented so that the voter-confidence in the election process is ensured.

RESULT:

The given problem statement is analysed and various functional requirements are listed.

EXP NO: 1B FUNCTIONAL REQUIREMENTS FOR LIBRARY MANAGEMENT SYSTEM

AIM:

To perform analysis of the given problem statement for listing the functional requirements

PROBLEM STATEMENT: Several problems such as inadequacy of infrastructure and hardware, problems in the software, lack of trained and skilled staff and lack of funds.

LIBRARY INFORMATION SYSTEM:

The library management system software helps in reducing operational costs. Managing a library manually is labor intensive and an immense amount of paperwork is involved. An automated system reduces the need for manpower and stationary. This leads to lower operational costs.

The system saves time for both the user and the librarian. With just a click the user can search for the books available in the library. The librarian can answer queries with ease regarding the availability of books. Adding, removing or editing the database is a simple process. Adding new members or cancelling existing memberships can be done with ease.

FUNCTIONAL REQUIREMENTS:

Administrator must be able to:

- Provide the information regarding books.
- Search for the required books from database.
- Add new book to the database.
- Update the number of books in database.
- Enter date of issued book in Database.
- Information of returned books.

Administrator Menu:

- Admin should be able to insert, modify and delete books.
- Can accept or reject a new user according to the library policy.
- Can get the information of any member who has borrowed a book.
- Add and edit book categories and arrange books by categories.
- Can record books returned by users.

List of books:

- Includes the bunch of books accordingly with series of department

User Requirements:

- Allow user to view quick reports like book issues/ returned etc. in particular time.
- Stock verification in different criteria or according to different categories.

New user Registration:

Any member of the institute who wishes to avail the facilities of the library has to register himself with the library information system.

Search book:

The system must be able to search the database based on select search type, must be able to filter book based on keyword entered and must be able to show the filtered book in table view.

Issue books and return books:

The system must be able to enter issue information in database, must be able to update number of books, must be able to search if book is available or not before issuing books and should be able to enter issue and return date information

Event addition:

System should be able to add detailed information about events and display information on notice board available in the homepage of site.

Renew the book:

If the time exceeds beyond the time limit of the management then one can be able renew the book for next few days so that it avoids paying of extra payment.

RESULT:

The given problem statement is analysed and various functional requirements are listed.

EXP NO: 1C FUNCTIONAL REQUIREMENTS FOR ONLINE SHOPPING SYSTEM

AIM:

To perform analysis of the given problem statement for listing the functional requirements

PROBLEM STATEMENT:

Offline shopping system: It includes the customer effort to shop the items and it makes the customers inconvenience, inefficiency.

Online shopping system: The functional and non-functional requirements were mentioned below to assure the benefit of online shopping system.

ONLINE SHOPPING SYSTEM:

Online shopping is a form of electronic commerce which allows consumers to directly buy goods or services from a seller over the Internet using a web browser or a mobile app.

FUNCTIONAL REQUIREMENTS:

- **Registration:**

  Every parson can view the store but only the members can checkout it become control by administrator. For being member of website, the customer needs to register for membership.

- **User Profile:**

  The User Profile page is area that given user maintains his /her own information. The user can only change his/her password required. The user can browse and search the item and add to shopping cart and can checkout and also the user can edit profile that include change shipping address and details. For other information to be changed, the super admin and admin privilege is required.

- **Item Selection:**

  Customer must be able to view his/her orders history. Each customer must be able to view placed order status. Customer should be able to validate and confirm their invoice

- **Browsing Through item:**

  The customer will be given a web interface which allows easy browsing through the different rooms. Basically the items in the shop are well Organized and well-presented so that a user can find his enquired item easily.

- **Login :**

  Only customer can login to the system with user id and password. The guest can registered or sign up to the system.

- **User Administration:**

This includes registration processing, user deactivation, user suspicious and required password change.

- **Payment:**

The customers choose the payment option when he was placed a order. The payment is handled by the bank and the payment which could be done either by using credit card and debit card or cash on delivery.

- **Online help:**

When the customers are needs a help the system will provide the online help system.

- **Administrative**

Administrative must be able to add/modify customer. System must be able to accept orders from the customer via email or online form. System must generate pro forma and customer invoice against an order, system must be able to view order history of a customer. System must beable to validate and manage customers order against payments.

RESULT:

The given problem statement is analysed and various functional requirements are listed

EXP NO: 2A NON-FUNCTIONAL REQUIREMENTS FOR ONLINE VOTING SYSTEM

AIM:

To perform analysis of the given problem statement for listing the non-functional requirements

PROBLEM STATEMENT:

i) Current offline voting system status:

Today, only about 1 percent of the population votes at polling places on hand counted paper ballots, but this figure is misleading. There are many elections conducted on optical mark-sense ballots that are actually hand counted, and many jurisdictions that use lever voting machines process absentee ballots by hand.

ii) New technology:

Today, an increasing fraction of the direct-recording electronic voting machines on the market include provisions to network all of the voting machines in one polling place. This allows each machine to store vote totals in the memory of the others, and at the close of the polls, it allows a single report for the entire precinct to be created instead of one report for each machine.

ONLINE VOTING SYSTEM:

Online voting systems protect the integrity of the vote by preventing voters from being able to vote multiple times. As a digital platform, they eliminate the need to gather in-person, cast votes using paper or by any other means. Online voting is a digital platform which allows the groups to conduct voting securely, High quality systems balance ballot security, accessibility, and overall requirements.

NON-FUNCTIONAL REQUIREMENTS:

Voter Authenticity:

Ensure that the voter must identify himself (with respect to the registration database) to be entitled to vote. If voting other than at his home precinct, the voter may be asked to show some legal identification document.

Accuracy: The system shall record and count all the votes and shall do so correctly.

Uniqueness: No voter should be able to vote more than once.

Audit ability: It should be possible to verify that all votes have been correctly accounted for in the final election tally, and there should be reliable and demonstrably authentic election records, in terms of physical, permanent audit trail (which should not reveal the user's identity in any manner).

Voter Anonymity:

Ensure that votes must not be associated with voter identity.

System Integrity:

Ensure that the system cannot be re-configured during operation.

Data Integrity:

Ensure that each vote is recorded as intended and cannot be tampered with in any manner, once recorded (i.e., votes should not be modified, forged or deleted without detection).

Reliability:

Election systems should work robustly, without loss of any votes, even in the face of numerous failures, including failures of voting machines and total loss of network communication. The system shall bedeveloped in a manner that ensures there is no malicious code or bugs.

Availability:

Ensure that system is protected against accidental and malicious denial of service attacks. Also, setup redundant communication paths so that availability is ensured.

System Disclosability:

The core of the system, especially the vote-casting equipment, shall be open source, so that it can allow external inspection and auditing.

Simplicity:

The system shall be designed to be extremely simple, as complexity is the enemy of security.

Testing and Certification:

The system should be tested by experts with respect to all of the security considerations, so that election officials have the confidence that the system meets the necessary criteria.

System Accountability:

Ensure that system operations are logged and audited.

Personnel Integrity:

Those developing and operating the voting system should have unquestionable records of behavior.

Operator Authentication and Control:

Ensure that those operating and administering the system are authenticated and have strictly controlled functional access on the system.

Distribution of Authority:

The administrative authority shall not rest with a single entity. The authority shall be distributed among multiple administrators, who are known not to collude among themselves (e.g., different political parties).

RESULT:

The given problem statement is analysed and various non-functional requirements are listed.

EXP NO: 2B NON-FUNCTIONAL REQUIREMENTS FOR LIBRARY MANAGEMENT SYSTEM

AIM:

To perform analysis of the given problem statement for listing the non-functional requirements

PROBLEM STATEMENT:

LIBRARY INFORMATION SYSTEM:

As the size and capacity of the institute is increasing with the time, it has been proposed to develop a Library Information System (LIS) for the benefit of students and employees of the institute. LIS will enable the members to borrow a book (or return it) with ease while sitting at his desk/chamber. The system also enables a member to extend the date of his borrowing if no other booking for that particular book has been made. For the library staff, this system aids them to easily handle day-to-day book transactions. The librarian, who has administrative privileges and complete control over the system, can enter a new record into the system when a new book has been purchased, or remove a record in case any book is taken off the shelf. Any non-member is free to use this system to browse/search books online. However, issuing or returning books is restricted to valid users (members) of LIS only. The final deliverable would a web application (using the recent HTML 5), which should run only within the institute LAN. Although this reduces security risk of the software to a large extent, care should be taken no confidential information (eg, passwords) is stored in plain text.

NON-FUNCTIONAL REQUIREMENTS:

EFFICIENCY REQUIREMENT:

When a library management system will be implemented librarian and user will easily access library as searching and book transaction will be very faster

PERFORMANCE REQUIREMENT:

The performance of the system should be fast and accurate and the system should be able to handle large amount of data. Thus it should accommodate high number of books and users without any fault.

RELIABILITY REQUIREMENT:

The system should accurately perform member registration, member validation, report generation, book transaction and search.

USABILITY REQUIREMENT:

The system is designed for a user friendly environment so that student and staff of library can perform the various tasks easily and in an effective way.

SAFETY REQUIREMENT:

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup so that the database is not lost. Proper UPS/inverter

facility should be there in case of power supply failure.

IMPLEMENTATION

In implementing whole system it uses html in front end with php as server side scripting language which will be used for database connectivity and the backend ie. the database part is developed using SQL.

DELIVERY:

The whole system is expected to be delivered in six months of time with a weekly evaluation by the project guide.

SECURITY REQUIREMENT:

System should use secured database, normal users can just read information but they cannot edit or modify anything except their personal and some other information. Proper user authentication should be provided

RESULT:

The given problem statement is analyzed and various functional requirements are listed.

EXP NO: 2C NON-FUNCTIONAL REQUIREMENTS FOR ONLINE SHOPPING SYSTEM

AIM:

To perform analysis of the given problem statement for listing the functional requirements

PROBLEM STATEMENT:

ONLINE SHOPPING SYSTEM:

In day to day life, we will need to buy lots of goods or products from a shop. It may be food items, electronic items, house hold items etc. Now days, it is really hard to get some time to go out and get them byourselves due to busy life style or lots of works. In order to solve this, B2C E-Commerce websites have beenstarted. Using these websites, we can buy goods or products online just by visiting the website and ordering the item online by making payments online. This existing system of buying goods has several disadvantages.It requires lots of time to travel to the particular shop to buy the goods. Since everyone is leading busy life now a day, time means a lot to everyone. Also there are expenses for travelling from house to shop. More over the shop from where we would like to buy something may not be open 24*7*365. Hence we have to adjust our time with the shopkeeper's time or vendor's time. In order to overcome these, we have e- commerce solution, ie. one place where we can get all required goods/products online. The proposed systemhelps in building a website to buy, sell products or goods online using internet connection. Purchasing of goods online, user can choose different products based on categories, online payments, delivery services andhence covering the disadvantages of the existing system and making the buying easier and helping the vendors to reach wider market

NON-FUNCTIONAL REQUIREMENTS:

Maintainability:

- The system should be developed in such a way that changes can be made easily, whether for bug fixes or to add new functionality.
- The system should be easy enough to maintain that someone else could do it with a manual and a few hours training.

Portability:

- The system should be portable to various operating environments.
- Should the current hosting become too restricting for the system, the system must be portable enough to be moved over to a new server with minimal downtime.

Integrity:

- The system should be able to protect and preserve transactions.

Manageability:

- The system should be developed in such a way that it can be easily reused, deployed and tested.

Usability:

- The user interface of the system should be very user friendly.

- It should not take more than 120 seconds for a new user to register for an account.
- It should not take more than 90 seconds for a registered user to place an order.

Security:

- There needs to be clearly defined roles of the users. These roles are 'customer' and 'administrator'. Each person that goes to the system's website will be required to register if they want to do more than just read / browse site content.
- A secure server will be required to ensure confidentiality of customer's credit card and other details.

Simplicity:

- The system shall be designed to be extremely simple, as complexity is the enemy of security.

System Accountability:

- Ensure that system operations are logged and audited.

Personnel Integrity:

- Those developing and operating the voting system should have unquestionable records of behavior.

RESULT:

The given problem statement is analysed and various non-functional requirements are listed.

EXP No: 3 UML DIAGRAM FOR ONLINE VOTING SYSTEM

AIM:
To draw UML diagram for Online voting system using Umbrello tool

PROBLEM STATEMENT:

Develop a scenario based modeling and draw the appropriate use case diagrams for the below given "Online Voting System". Internet has led to discussion of e-democracy and online voting. Many peoples think that the internet could replace representative democracy, enabling everyone to vote on everything and anything by online voting.Online voting could reduce cost and make voting more convenient. This type of voting can be done for e-democracy, or it may be used for finalizing a solution, if many alternatives are present. Online voting make's use of authentication, hence it needs security, and the system must be able to address obtaining, marking, delivering and counting ballots via computer. Advantage of online voting is it could increase voter turnout because of convenience, and it helps to reduce fraud voting.

USE CASE SCENARIOS:

The voter scenarios are:

- The voter login to the system.
- The voter can begin to vote at any time.
- We have only 1 change to change a vote
- Caste vote
- View online help

Admin scenario arc: -

- Admin can login to the system
- Admin has an activities like update and delete or add new candidate
- File can give the results of the elections
- He can filter the results
- He has the permissions like modified the voter profile etc

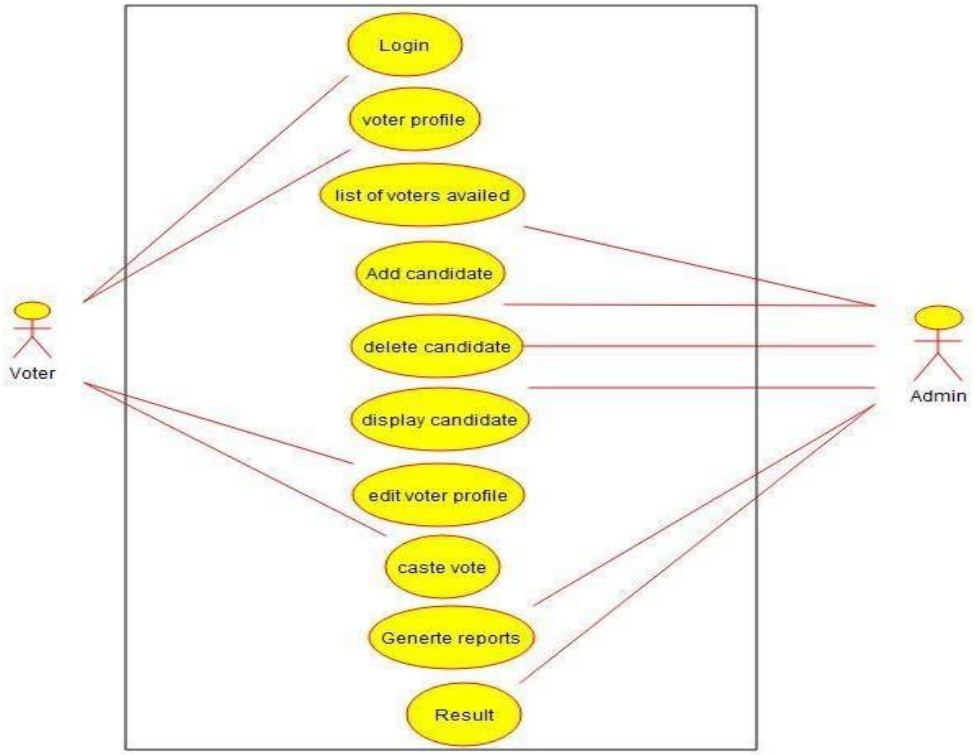
Login: -Both the admin and valid voter can login into the system. The admin should provide a password and login ID to the voters before the election starts

Voter profile: -The voter can seen their profile but changes not be done by the voter only admin can change the voter profile

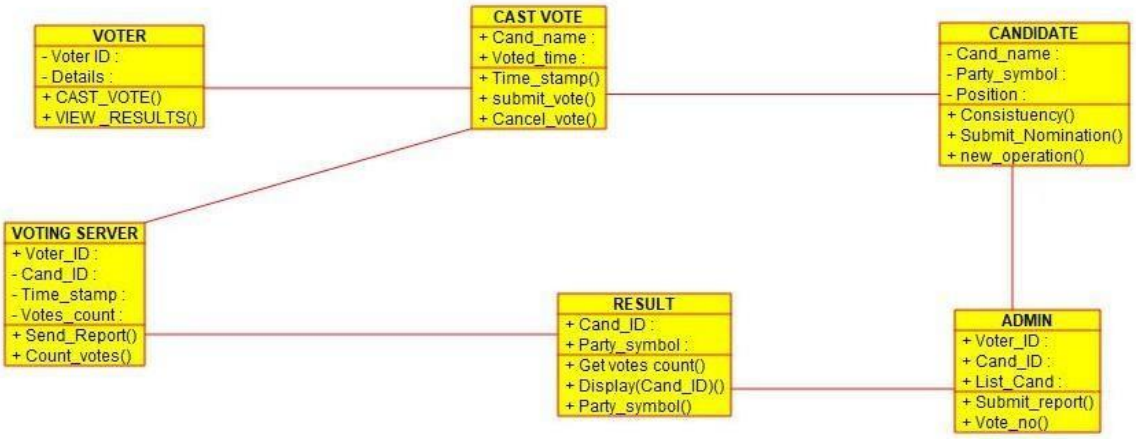
Add candidate:-The admin can have the permissions like to add the candidate or update the details of the candidate

Caste vote: -The voter can use the vote on the day of polling can be due to their respective candidate

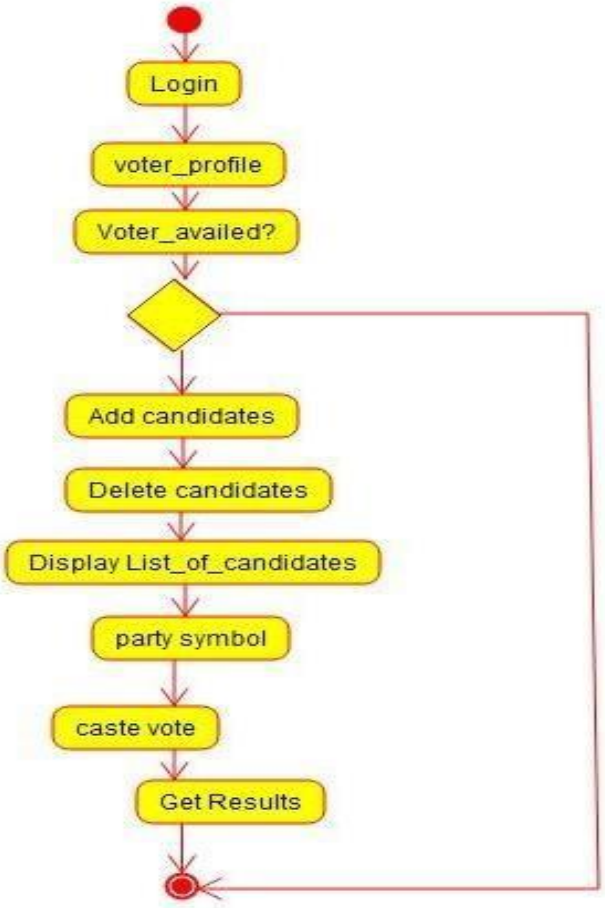
USE CASE DIAGRAM:



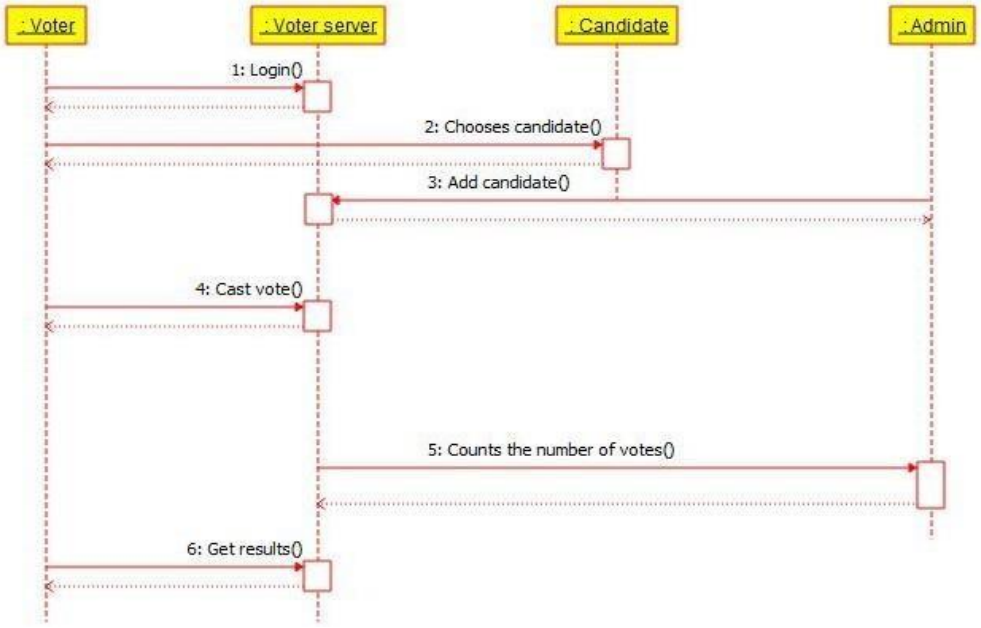
CLASS DIAGRAM:



ACTIVITY DIAGRAM:



SEQUENCE DIAGRAM:



RESULT:

Thus the use-case diagram for online voting system is drawn and verified successfully.

EXP NO: 4 UML DIAGRAM FOR LIBRARY MANAGEMENT SYSTEM

AIM:

To draw UML diagrams for Library management system using Umbrello tool.

INTRODUCTION:

In the Unified Modeling Language, a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or external systems
- Goals that it helps those entities (known as actors) achieve
- The scope of your system

Common components include:

- **Actors** - the users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.
- **System** - a specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.
- **Goals** - the end result of most use cases. A successful diagram should describe the activities and variants used to reach the goal.

USE CASE DIAGRAM OBJECTS

- Actor
- Use case
- System
- Package

The objects are further explained below.

Actor:

Actor in a use case diagram is **any entity that performs a role** in one given system. This could be a person, organization or an external system and usually drawn like skeleton shown below.

Use Case:

A use case **represents a function or an action within the system**. Its drawn as an oval and named with the function.

System:

System is used to **define the scope of the use case** and drawn as a rectangle. This an optional element but useful when your visualizing large systems. For example you can create all the use cases and then use the system object to define the scope covered by your project. Or you can even use it to show the different areas covered in different releases.

Package:

Package is another optional element that is extremely useful in complex diagrams. Similar to **class diagrams**, packages are **used to group together use cases**. They are drawn like the image shown below.

USE CASE SCENARIOS:

LOGIN:

To interact with the system, LMS will validate its registration with this system.

The actors involved are

- Administrator
- Librarian
- member

View user details:

1. To see the details of the registered user & the books currently borrowed from the library.

2. Member can involve.

3.User must be logged into the system.

View books:

1. To display the details, when a member, guest or administrator want to see the details on the available books.

2. The Actors involved in step are Administrator, guest and member.

Reserve Books:

1. User can reserve a book by inputting the relevant details and the librarian can also reserve a book for a member

Search books:

1.Member or guest can search for a particular book in the book library by book name or category or author name.

Issue books:

1. This use case can describe the process of issuing a certain book for a member by librarian.

2. Get the member ID and book ID before issue a book.

3. Check the availability.

Return books:

1. This use case describes the process of return a book.

2. If return book is late member should be paid fine.

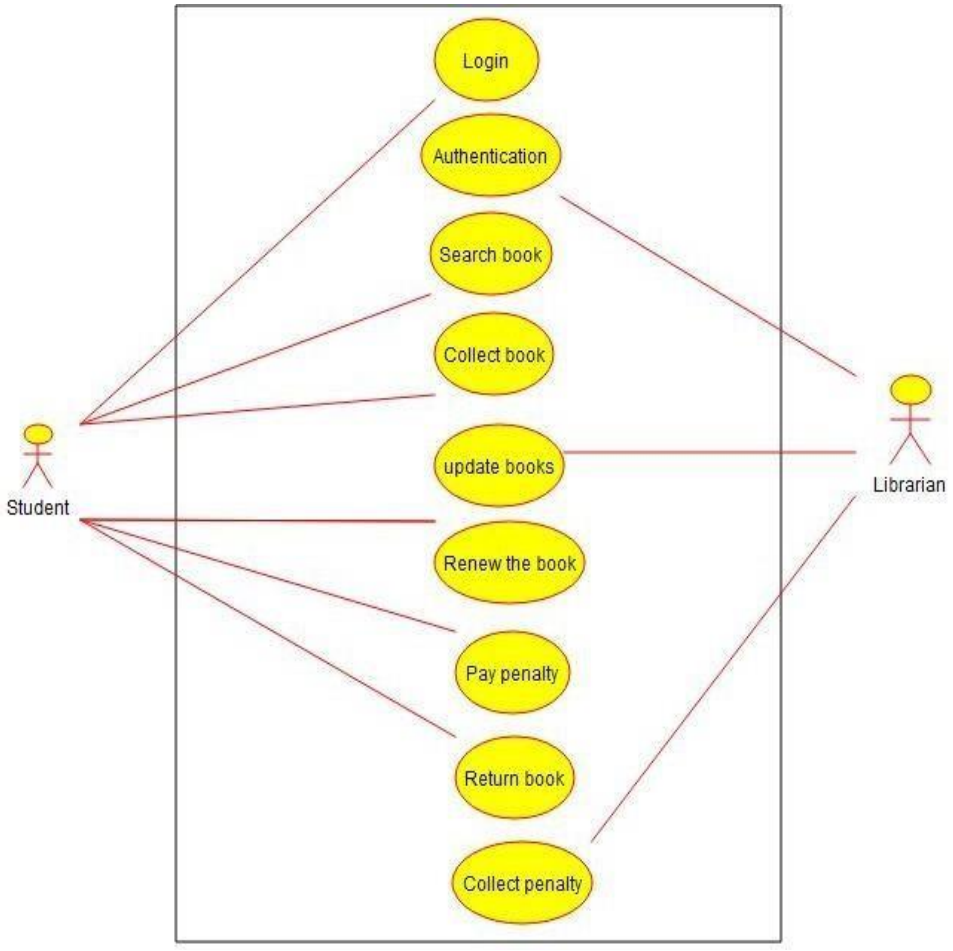
View Members:

To display the details, when a member, guest or administrator wants to see the details of the registered user.

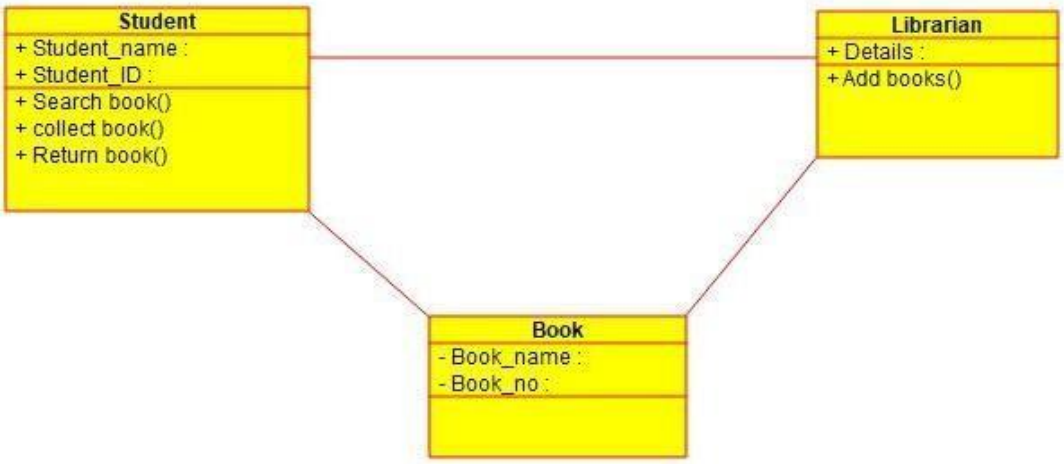
Add /remove members:

Only administrator is allowed to add or remove a member from library data base .to remove a member, member should request to leave the library.

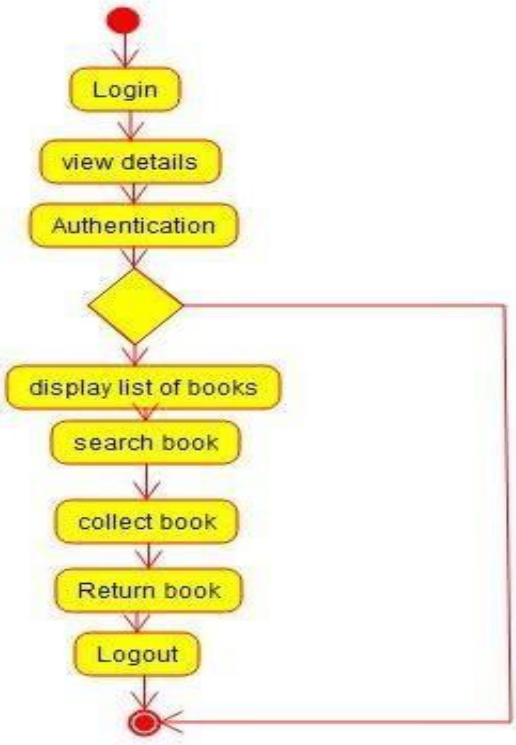
USE CASE DIAGRAM:



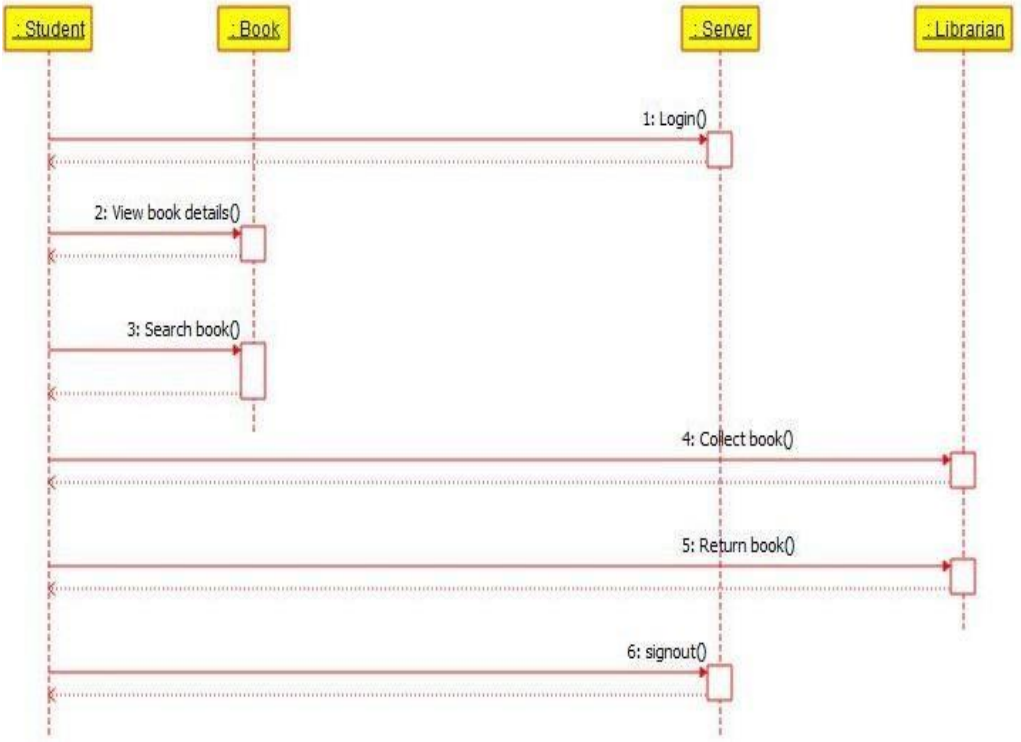
CLASS DIAGRAM:



ACTIVITY DIAGRAM:



SEQUENCE DIAGRAM:



RESULT:

Thus the use-case diagram for online voting system is drawn and verified successfully.

EXP NO: 5 **UML DIAGRAM FOR ONLINR SHOPPING SYSTEM**

AIM:
To draw UML diagram for Online voting system using Umbrello tool.

PROBLEM STATEMENT:

Online shopping system:

In day to day life, we will need to buy lots of goods or products from a shop. It may be food items, electronic items, house hold items etc etc. Now a days, it is really hard to get some time to go out and get them by ourselves due to busy life style or lots of works. In order to solve this, B2C E-Commerce websites have been started. Using these websites, we can buy goods or products online just by visiting the website and ordering the item online by making payments online. This existing system of buying goods has several disadvantages. It requires lots of time to travel to the particular shop to buy the goods. Since everyone is leading busy life now a days, time means a lot to everyone. Also there are expenses for travelling from house to shop. More over the shop from where we would like to buy something may not be open 24*7*365. Hence we have to adjust our time with the shopkeeper's time or vendor's time. In order to overcome these, we have e-commerce solution, ie. one place where we can get all required goods/products online. The proposed system helps in building a website to buy, sell products or goods online using internet connection. Purchasing of goods online, user can choose different products based on categories, online payments , delivery services and hence covering the disadvantages of the existing system and making the buying easier and helping the vendors to reach wider market.draw a use case diagram for given scenario.

Login/Register:

- The user is a visitor, who visits the site and buy something from our website by making an online payment (or) otherwise he can register.
- Both customer and admin have different task in online system.

Admin:

Admin is a responsible person to run the whole system. Admin can add delete, update all information, orderinformation, item information.

Admin scenarios:

- Login module
- Manage items
- Process a order

- Cancel the order
- Process return
- Online help

Customer scenarios:

- Registration for new customer
- Login for existing customers
- Place a order
- Browser the item
- Change the password
- Cancel the order
- Make payment

Courier service scenarios are:

- The courier service has to provide by the deliver the orders to the customers.
- And also return items for the customer’s ad send to the back to sales marketing.

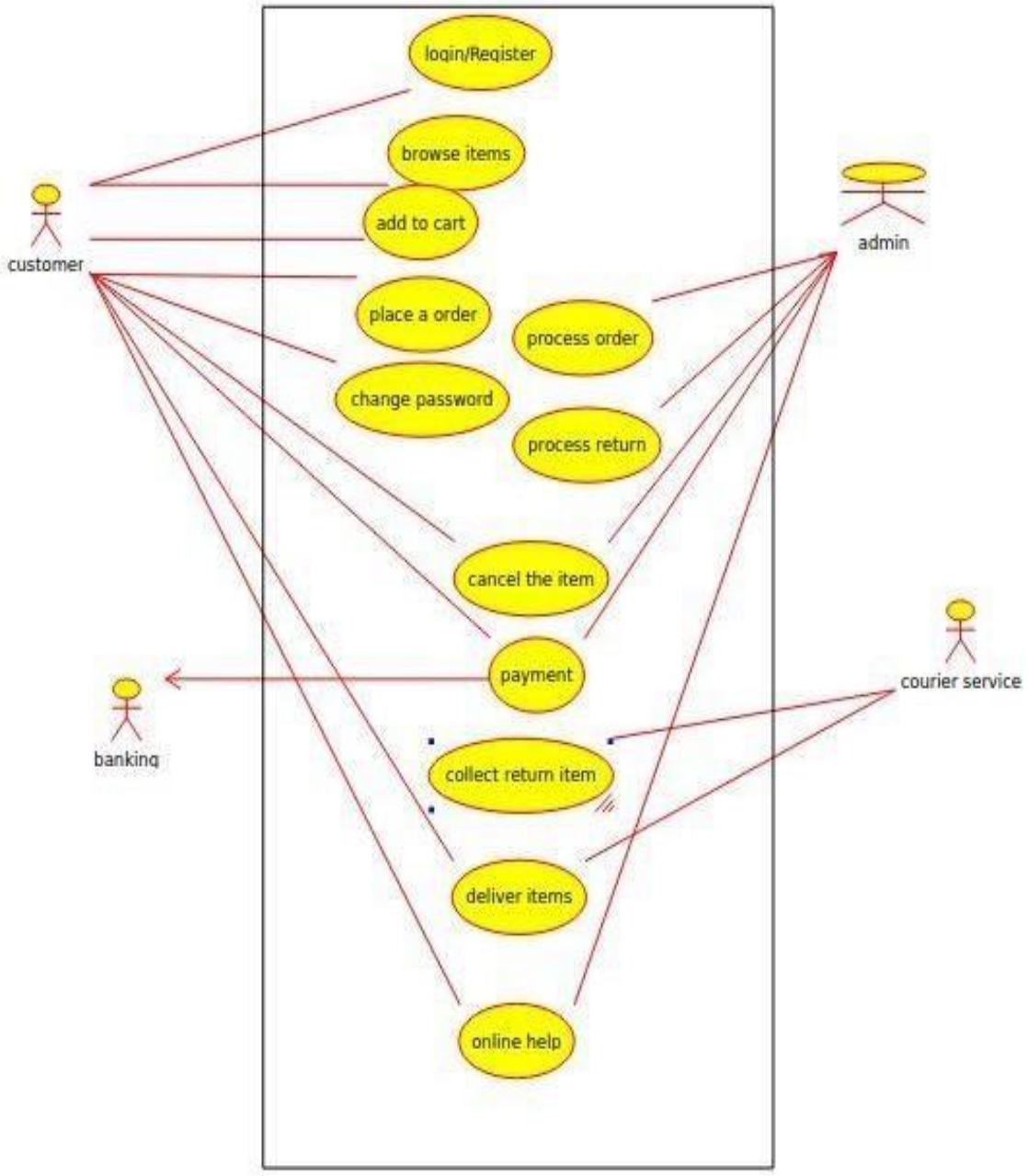
Payment:

The customer choose the payment option when he was placed a order. The payment is handled by the bankand the payment which could be done either by using credit card and debit card or cash on delivery.

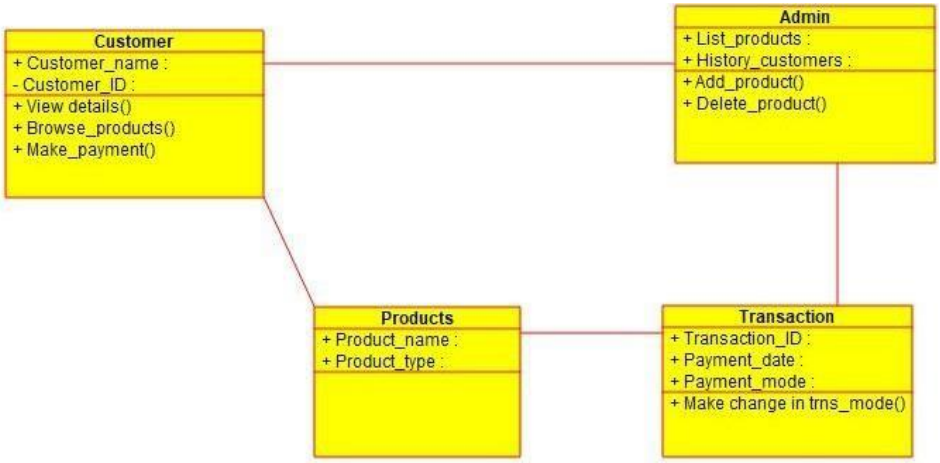
Online help:

When the customers are needs a help the system will provide the online help system.

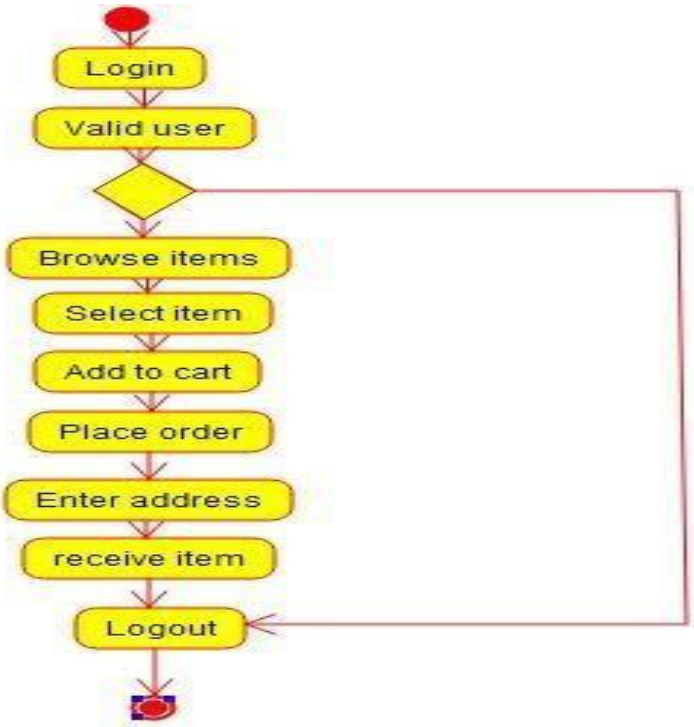
USECASE DIAGRAM:



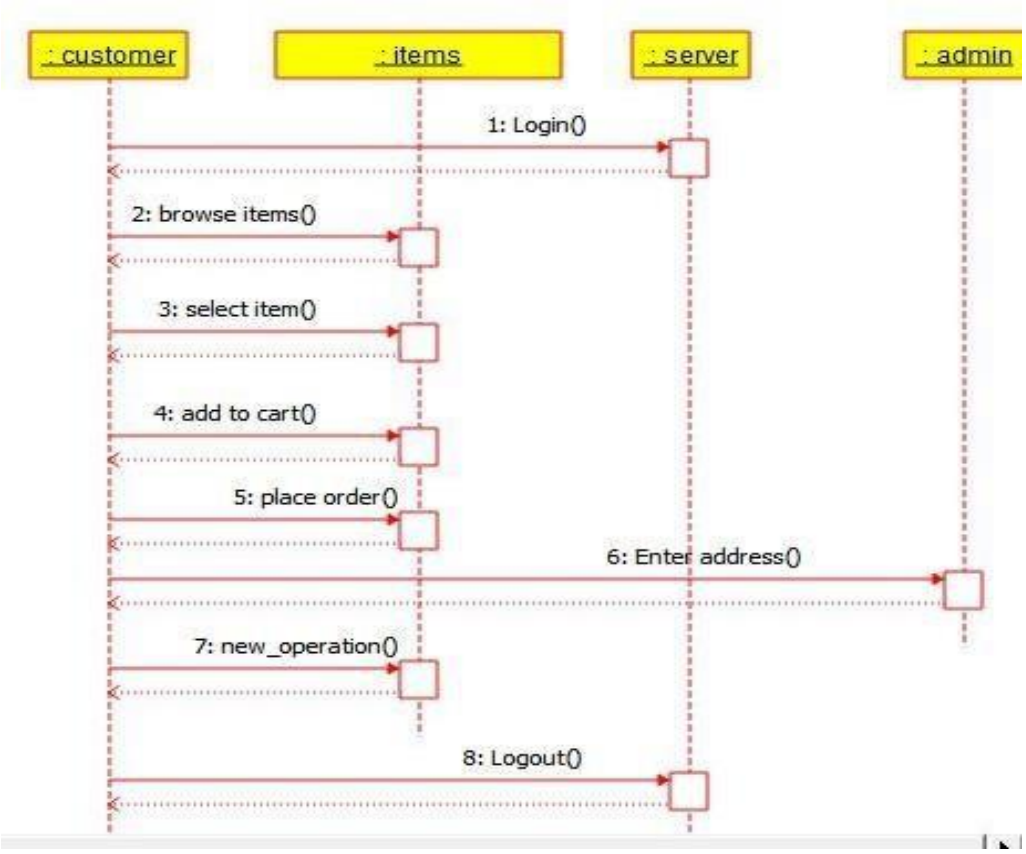
CLASS DIAGRAM:



ACTIVITY DIAGRAM:



SEQUENCE DIAGRAM:



RESULT:

Thus the use-case diagram for online shopping system is drawn and verified successfully.

EXP NO:7A **VALIDATING THE PROCESS FLOW OF CODE USING RAPTOR - TO FIND GIVEN NUMBER IS PRIME OR NOT**

AIM:

To draw a flowchart for finding given number is prime or not and validate the process flow using RAPTOR tool.

ALGORITHM:

STEP 1: Get a number from the user

STEP 2: Assign value of 1 to variable b and 0 to variable A

STEP 3: if the value of b is greater or equal to a, the loop is bypassed. If not the loop is initiated.

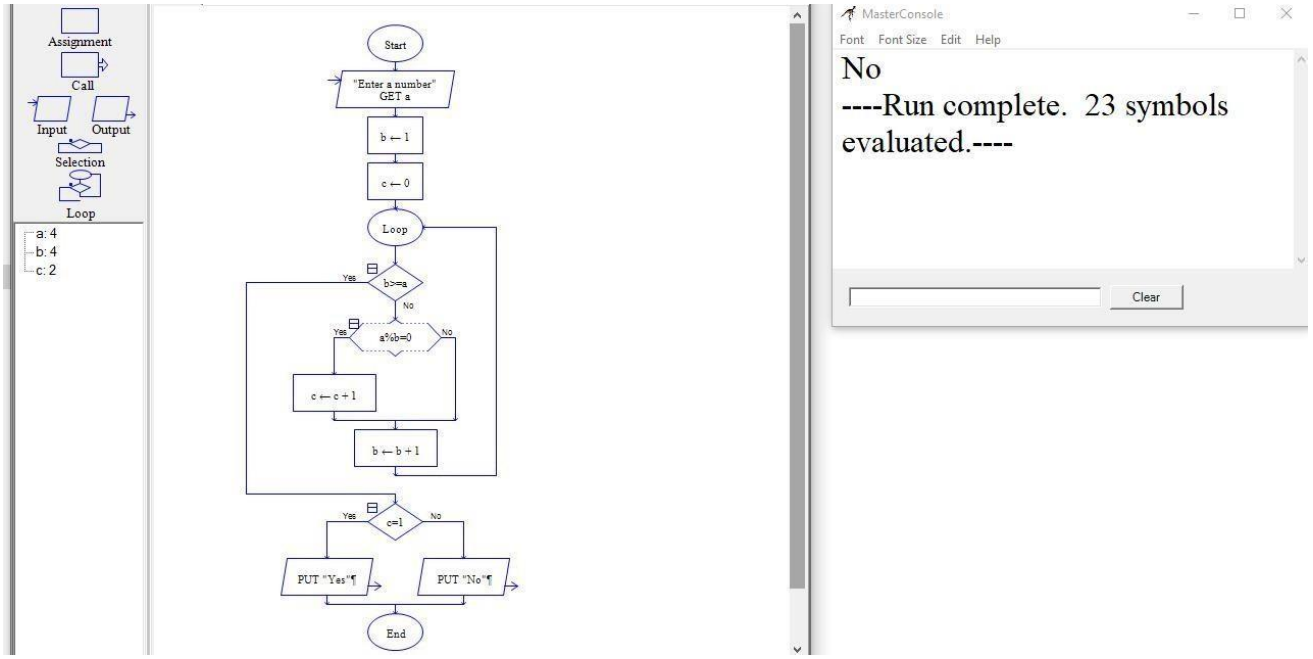
STEP 4: The value of b divided by a is checked with the value 0.

STEP 5: If b%a is equal to 0, then the value of c is incremented by 1 and b is incremented by 1.

STEP 6: If not, the value of b is incremented by 1 and the loop runs again.

STEP 7: If the value of c is equal to 1, entered value is a prime number. Else, it is not a prime number.

FLOWCHART:



RESULT:

Thus,The program is executed and verified successfully.

EXP NO: 7B VALIDATING THE PROCESS FLOW OF CODE USING RAPTOR

- TO COMPUTE QUOTIENT AND REMAINDER

AIM:

To draw a flowchart to compute quotient and remainder and validate the process flow using RAPTOR

STEPS INVOLVED:

STEP 1: Two variables Dd and Dr are initiated. The values of both these variable are obtained from the user.

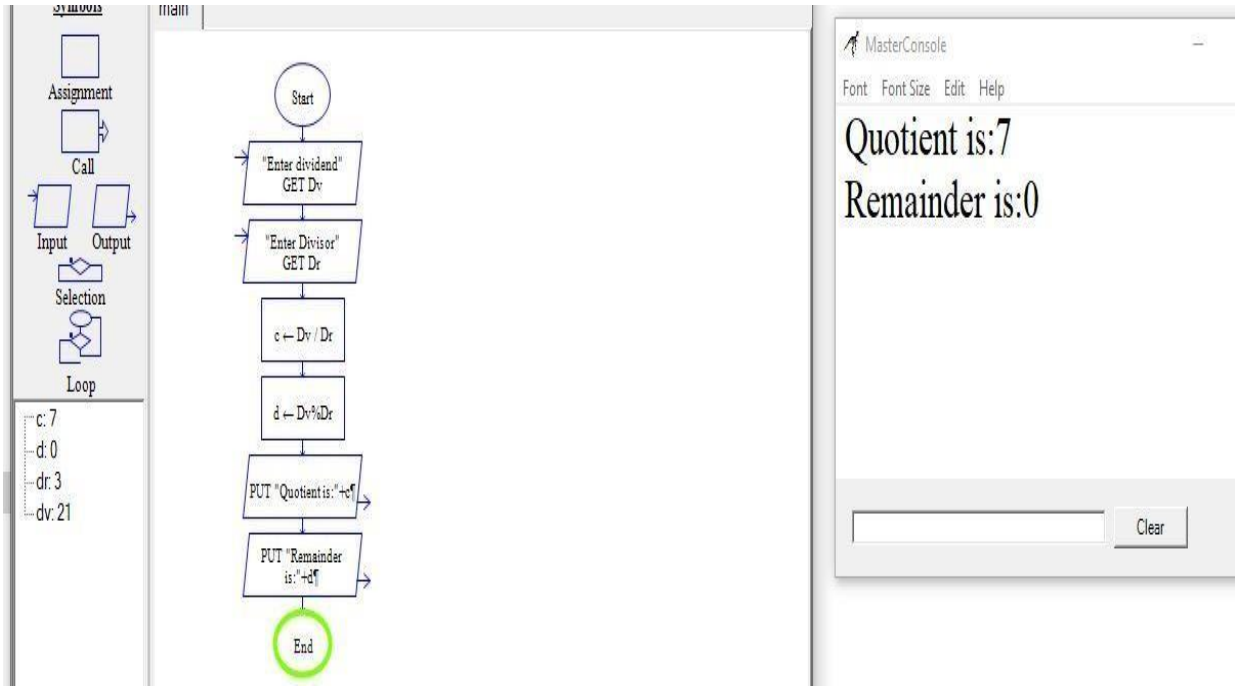
STEP 2: The mod of Dr and Dd are stored in a variable C and returned as the remainder.

STEP 3: The answer of the arithmetic operation Dd/Dr is stored in D

STEP 4: The value of D is returned as the quotient.

STEP 5: The program is terminated.

Flowchart:



RESULT:

Thus, the program is executed and verified successfully.

EXP NO:7C VALIDATING THE PROCESS FLOW OF CODE USING RAPTOR

- TO CHECK GIVEN NUMBER IS PALINDROME OR NOT

AIM:

To draw a flowchart to check given number is palindrome or not and validate the process flow using RAPTOR.

STEPS INVOLVED:

STEP 1: A variable “palindrome” is initiated and assigned the value 0.

STEP 2: A number is obtained from the user and is assigned to “n”.

STEP 3: A variable m is crated and assigned the value of n.

STEP 4: If n is less than or equal to 0, the program is terminated. Else, a loop is initiated.

STEP 5: The value of palindrome is multiplied by 10.

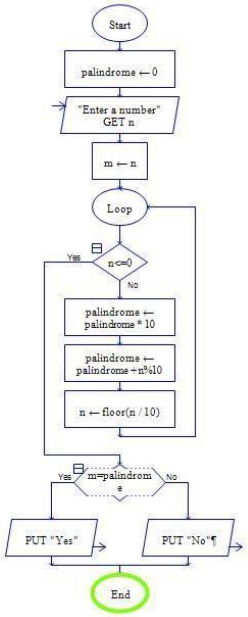
STEP 6: The new value of palindrome is added with the remained of the input number divided by 10.

STEP 7: The value of n is then divided by 10 and the loop is executed until the value of n is less than 0.

STEP 8: A condition is then checked. If the value of palindrome and m are the same, then the number is a palindrome.

STEP 9: Program is terminated.

FLOWCHART:



RESULT:

Thus, The program executed and verified successfully.

EXP NO:7D VALIDATING THE PROCESS FLOW OF CODE USING RAPTOR

- TO PRINT AND COUNT VOWELS IN A GIVEN WORD

AIM:

To draw a flowchart to print and count vowels in a given word and validate the process flow using RAPTOR.

STEPS INVOLVED:

STEP 1: A string is obtained from the user and the string length function is used to calculate the string length.STEP 2: 2 variables, counter and character are created and assigned the values 0 and 1 respectively

STEP 3: A loop is initiated and the relation, number<string length is evaluated.

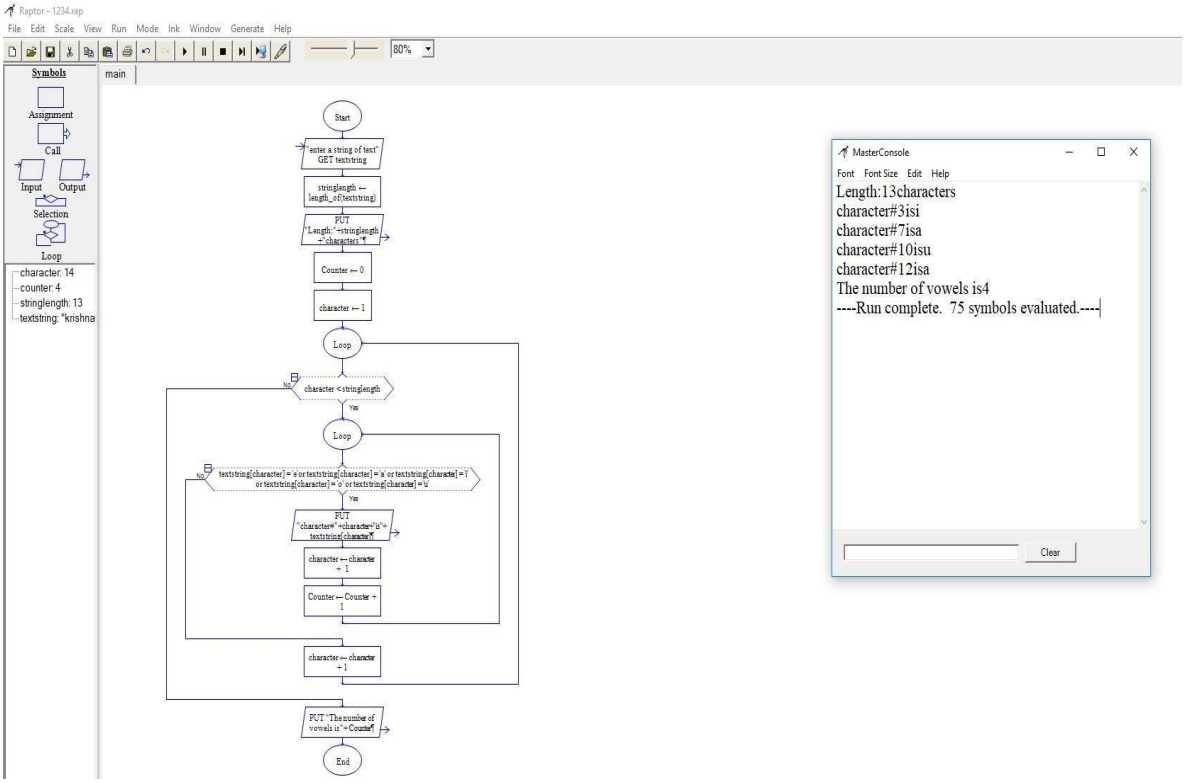
STEP 4: The condition that whether each character is equal to (a,e,i,o and u) is checked.

STEP 5: If the condition is true, the respective character of the string is returned to user and displayed as avowel.

STEP 6: The counter variable is subsequently incremented by 1.

STEP 7: At the end of the program, the counter variable is returned to the user as the total number of vowels in the string.

FLOWCHART:



RESULT:

Thus, The program executed and verified successfully.

EXP NO: 8A VALIDATING THE PROCESS FLOW OF CODE USING RAPTOR

- TO FIND FIBONACCI SERIES

AIM:

To draw the flowchart for finding Fibonacci Series and validate the process flow using RAPTOR tool.

STEPS INVOLVED:

STEP 1: Start the Program

STEP 2: Get the input variables.

STEP 3: Assign ‘a’ – 0, ‘b’ – 1, ‘c’ – 1 and ‘i’ – 1.

STEP 4: Loop is initiated.

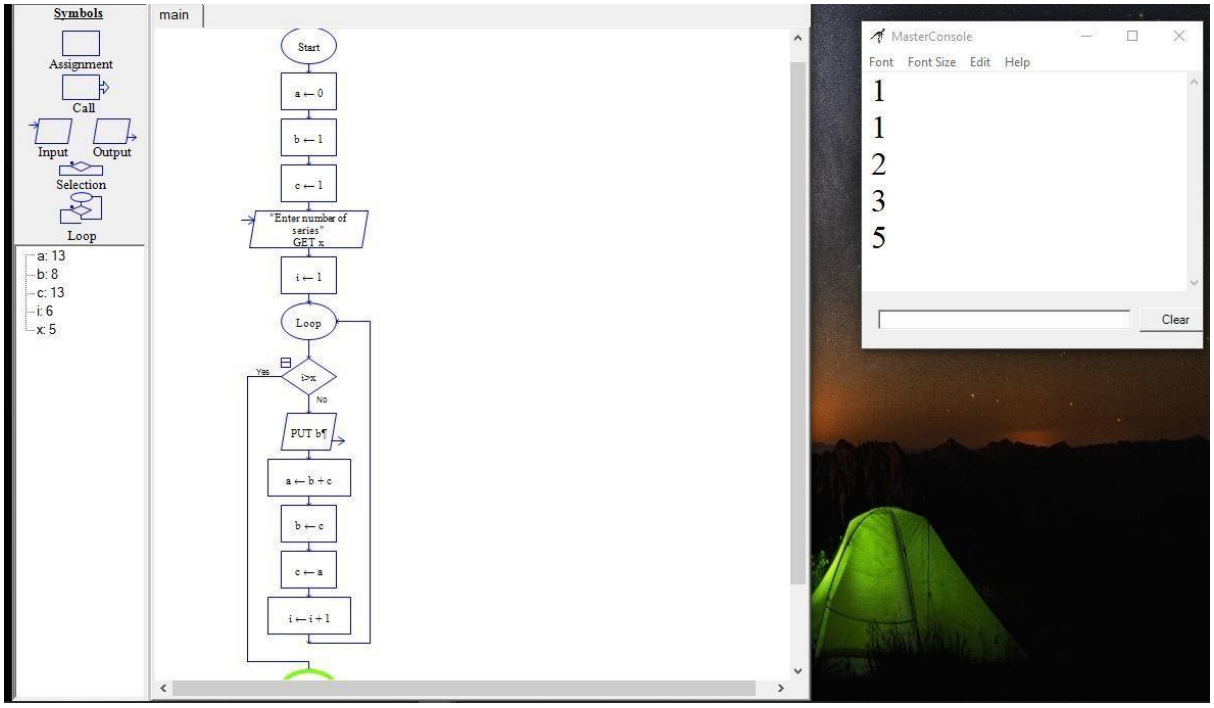
STEP 5: If ‘i’ value greater than ‘x’ end the program.

STEP 6: Else the value ‘b’ is written as output, ‘a’ is assigned the value sum of ‘b’ and ‘c’, ‘b’ is assigned the value of ‘c’ and ‘c’ is assigned the value of ‘a’.

STEP 7: ‘i’ is incremented by 1 after each turn.

STEP 8: Stop the program.

FLOWCHART:



RESULT:

Thus, The program is executed and verified successfully.

EXP NO: 8B VALIDATING THE PROCESS FLOW OF CODE USING RAPTOR

- TO DISPLAY THE POSITION OF THE NUMBER

AIM:

To draw a flowchart to display the position of the number and validate the process flow using RAPTOR.

STEPS INVOLVED:

STEP 1: A string of numbers is obtained from the user.

STEP 2:stringlength function is used to calculate the length of the string and is displayed.

STEP 3: A variable number is initiated and assigned the value 1.

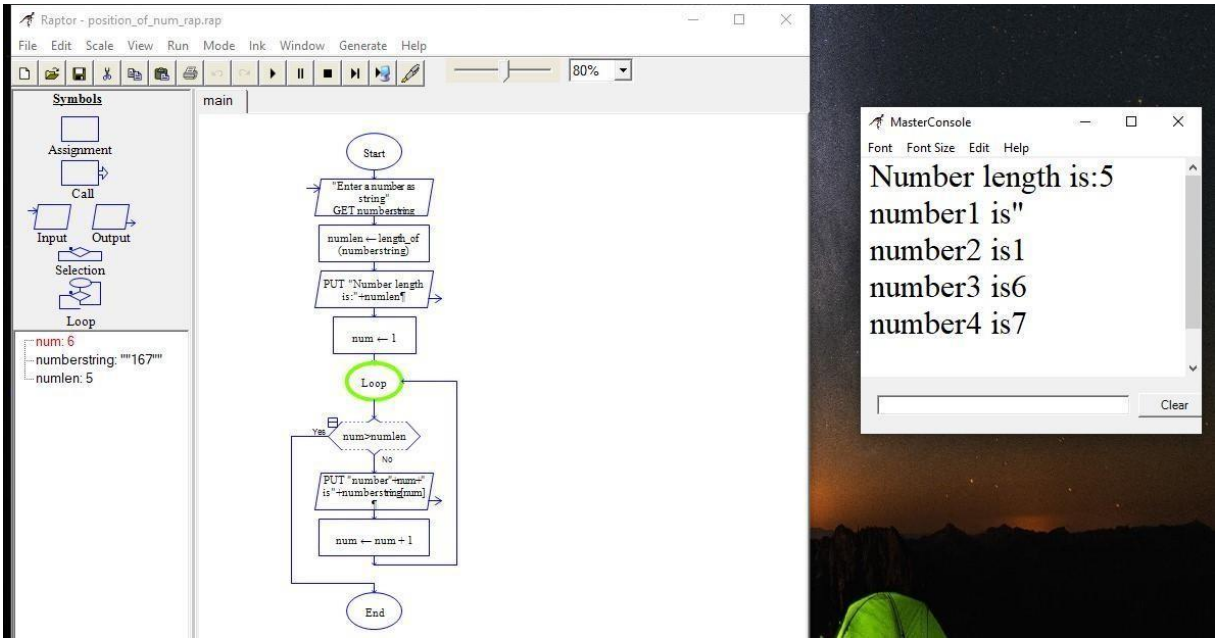
STEP 4: Loop is inititated, where the condition number>stringlength is checked.

STEP 5: If above condition is true, each number of the array is displayed along with position with the help of"number" variable.

STEP 6: At the end of each turn, number is incremented by

1.STEP 7: Once "number"> stringlength, the program ends.

FLOWCHART:



RESULT:

The program is executed and verified successfully

EXP NO: 8C VALIDATING THE PROCESS FLOW OF CODE USING RAPTOR

- TO SWAP TWO CHARACTERS

AIM:

To draw a flowchart to swap two characters and validate the process flow using RAPTOR tool.

STEPS INVOLVED:

STEP 1: Three variables n1,n2, and temp are initiated and assigned the value 0.

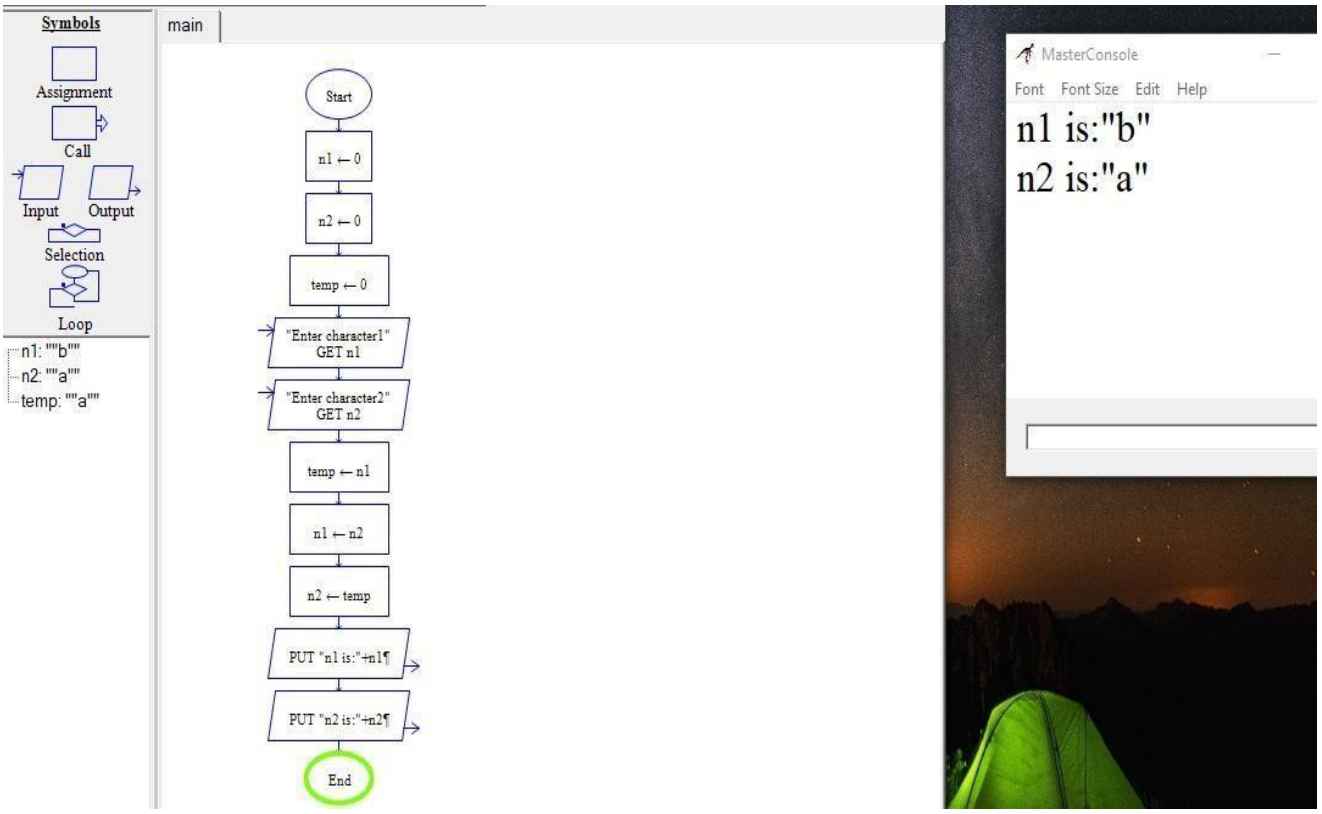
STEP 2: The user enters the value for n1 and n2.

STEP 3: temp variable is assigned the value of n1 and then n1 is assigned the value of n2.

STEP 4: n2 is now assigned the value of temp.

STEP 5: The exchanged values of n1 and n2 are returned.

FLOWCHART:



RESULT:

The program is executed and verified successfully.

EXP NO:8D VALIDATING THE PROCESS FLOW OF CODE USING RAPTOR

- TO FIND GIVEN NUMBER IS ARMSTRONG OR NOT

AIM:

To draw a flowchart to check given number is armstrong or not and validate the process flow using RAPTOR tool.

STEPS INVOLVED:

STEP 1: A number is acquired as the input from the user.

STEP 2:2 variables, sum and n are initiated and assigned the values of 0 and the input number respectively.

STEP 3: A loop is initiated and the condition n<1 is analyzed.

STEP 4: If the condition is not true, a variable remainder is created and assigned the value of the remainder of n divided by 10.

STEP 5: The variable sum is assigned the value of sum of the variable “sum” and the cube of “remainder”.

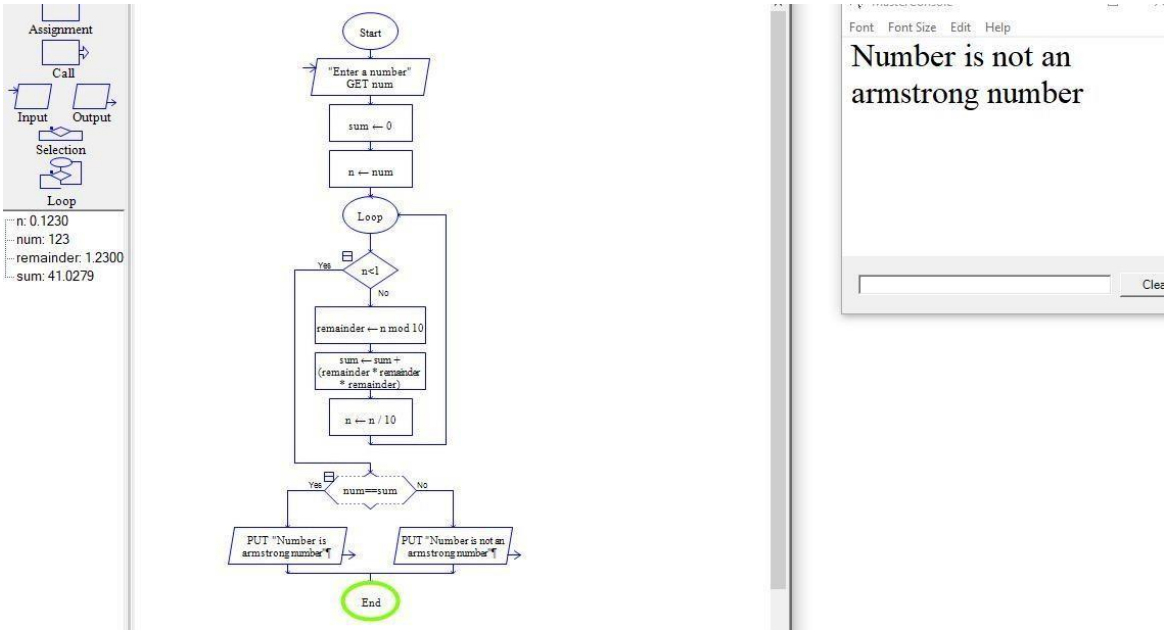
STEP 6: The remainder of n is subtracted from n and is subsequently divided by 10.

STEP 7: The loop is repeated until the initial condition is met.

STEP 8: The condition, (sum=num) is analyzed and if true, the number entered by the user is returned as nArmstrong number.

STEP 9: The program is then terminated.

FLOWCHART:



RESULT: The program is executed and verified successfully.

EXP NO: 9B VALIDATING THE PROCESS FLOW OF CODE USING RAPTOR

- TO DISPLAY PRIME NUMBERS FROM 1-100 NUMBERS

AIM:

To draw the flowchart for displaying the prime numbers from 1 to 100 numbers.

ALGORITHM:

STEP 1: 3 variables i,j and k are created and assigned the value 1,1 and 0 respectively.

STEP 2:A number is then obtained from the user.

STEP 3: The condition i>1 and j>1 are analyzed. If not true, a loop is initiated.

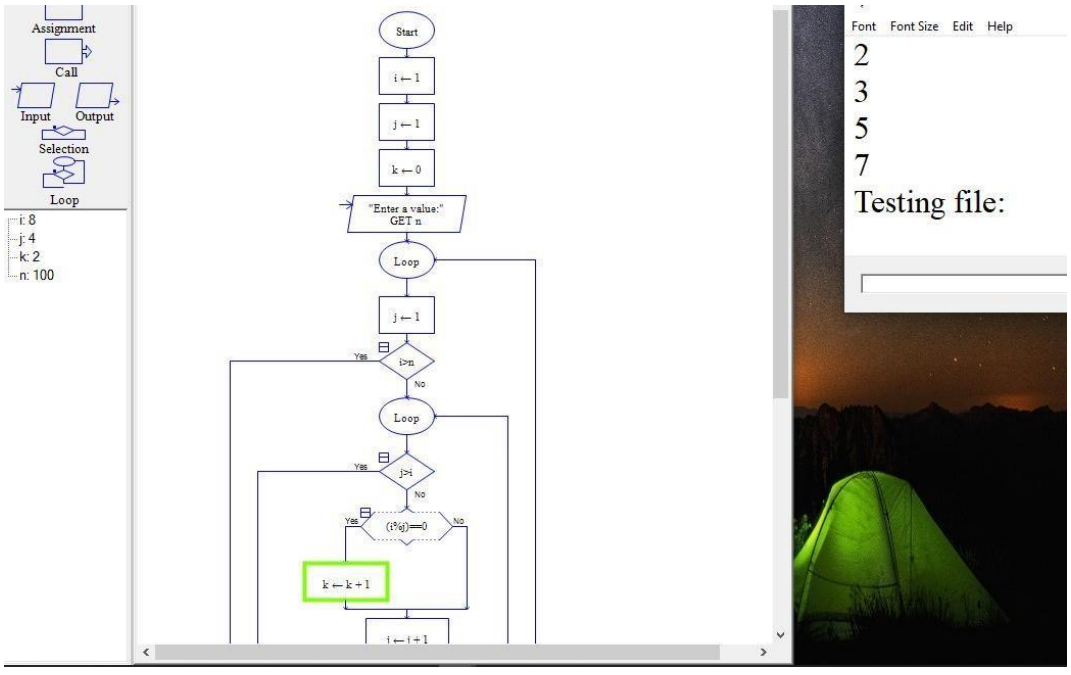
STEP 4:Each turn, the remainder of I divided by j is calculated, and if it is equal to zero, the value of k and j are initiated by 1. When j>i, the loop passes on to condition check.

STEP 5: if the value of k is equal to 2, the value of i is returned as a prime number, and the value of k is assigned the value of 0.

STEP 6: The value of i is incremented by 1 abd again the condition i>n is checked. Until it is true, the above mentioned loop is executed.

STEP 7: When i>n, the program is terminated.

FLOWCHART:



RESULT:

Thus,The program is executed and verified successfully.

EXP NO:10A VALIDATING THE PROCESS FLOW OF CODE USING RAPTOR

- TO SWAP TWO NUMBERS

AIM:

To draw a flowchart to swap two number and validate the process flow using RAPTOR tool.

ALGORITHM:

STEP 1: Three variables n1,n2, and temp are initiated and assigned the value 0.

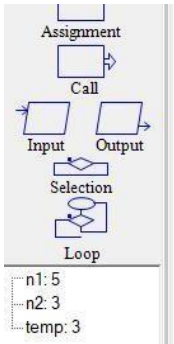
STEP 2: The user enters the value for n1 and n2.

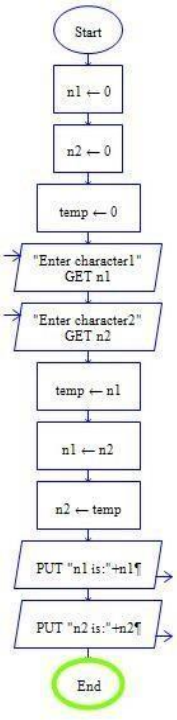
STEP 3: TEMP variable is assigned the value of n1 and then n1 is assigned the value of n2.

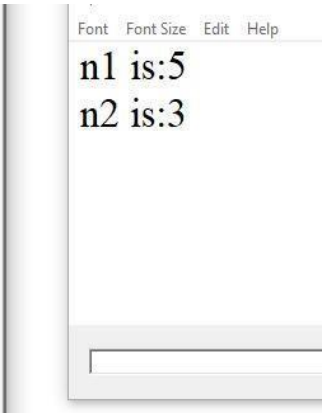
STEP 4: n2 is now assigned the value of temp.

STEP 5: The exchanged values of n1 and n2 are returned.

FLOWCHART:







RESULT:

Thus, The program is executed and verified successfully

EXP NO:10B **VALIDATING THE PROCESS FLOW OF CODE USING RAPTOR**

- TO DISPLAY LENGTH OF THE STRING

AIM:

To draw the flowchart for displaying the length of the string using Raptor tool.

ALGORITHM:

STEP 1: Start.

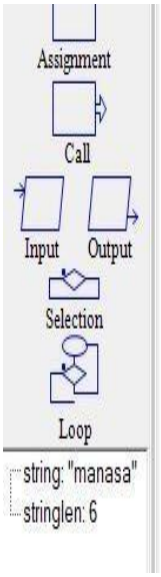
STEP 2: Get the string from user and name that string.

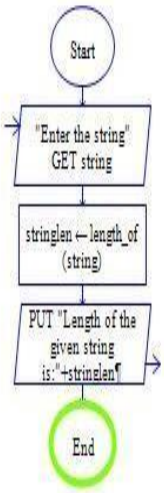
STEP 3: Assign string length as length_of(string) function.

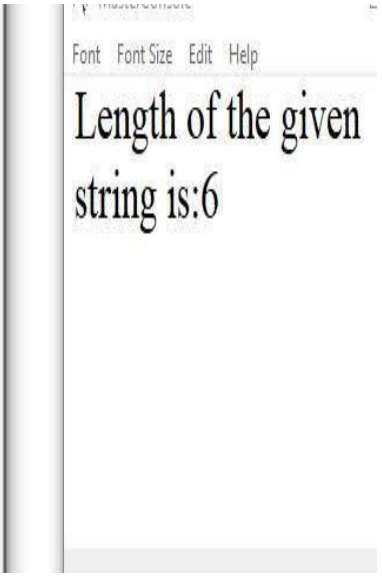
STEP 4: Print the length of string.

STEP 5: Stop.

FLOWCHART:







RESULT:

Thus, The program executed and verified successfully.

EXP.No 11: Automation Testing using Selenium tool for student login of their Institution(Ex-Arms Portal)

Aim: Testing the arms portal by using selenium tools

Algorithm:

Step1 : Start the programme

Step2: Import selenium tool in java editor

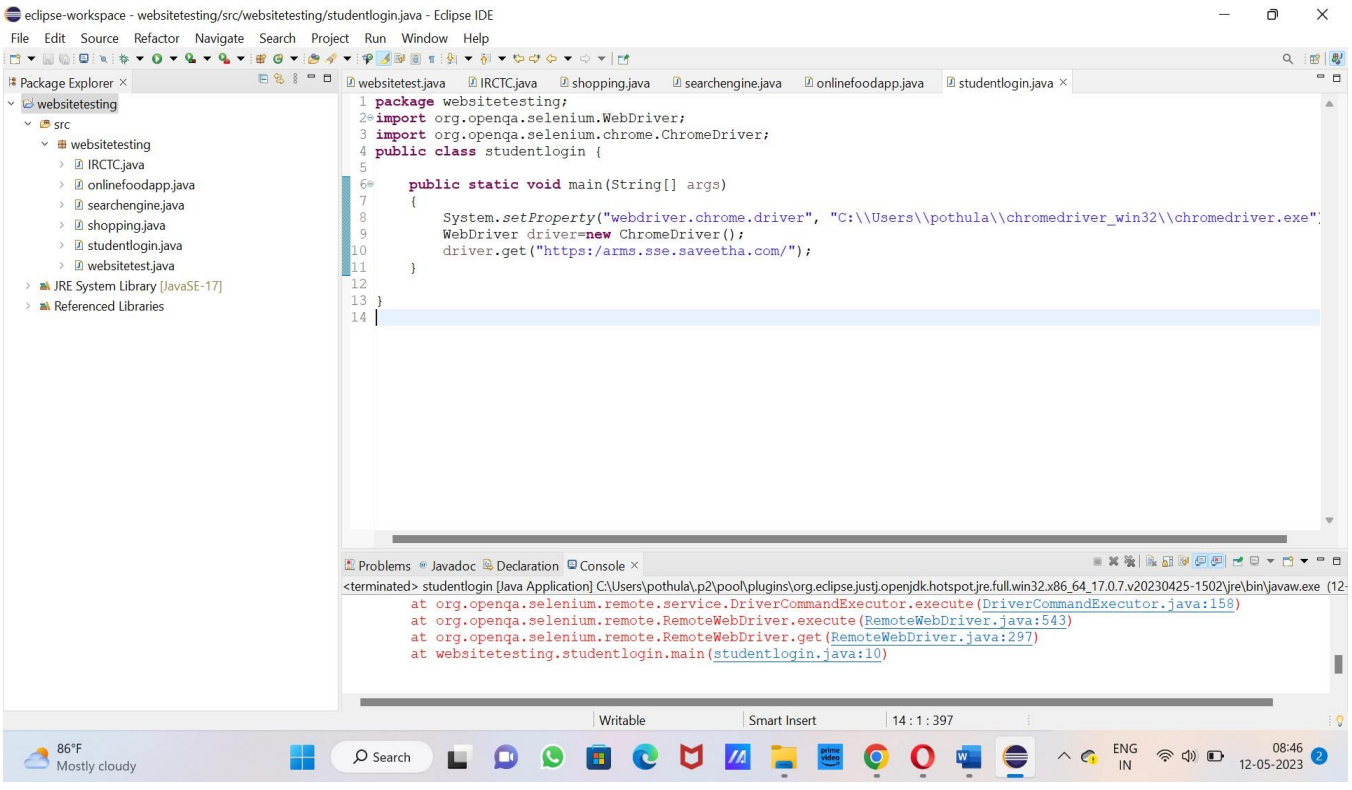
Step3 : Select the element to test

Step4: check element is present in wesite or not.

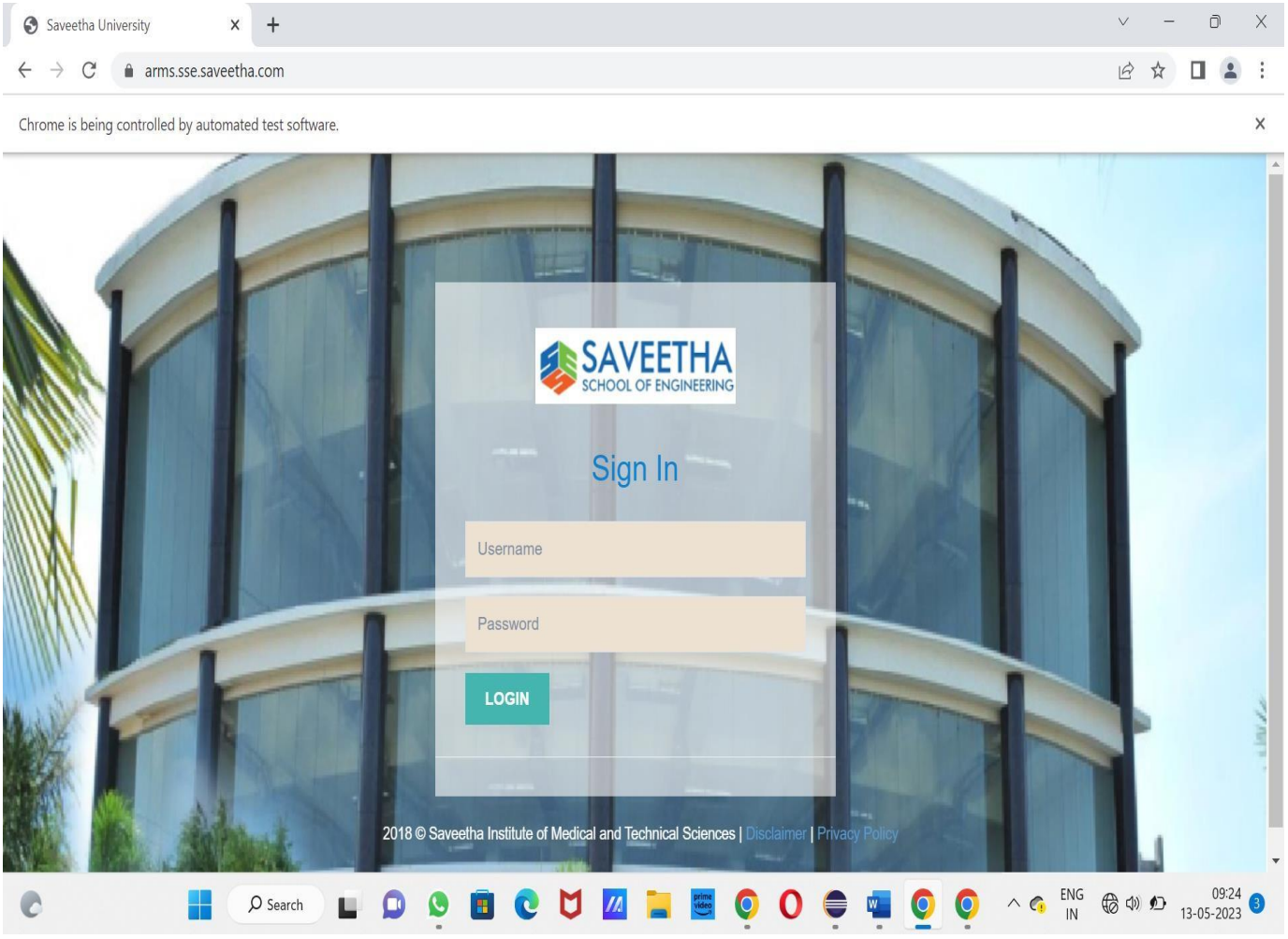
Step5: End the programme.

Code:

```
Package websitetesting;
Importnorg.openqa.selenium.WebDriver;
Importnorg.openqa.selenium.Chrome.ChromeDriver;
Public class studentlogin  {

        Public static void main(string[] args)
 {

System.setproperty("webdriver.chrome.driver","c:\\Users\\pothula\\chromedriver_win32\\chromedriver.exe
")
WebDriver driver=new ChromeDriver();
Driver.get("'https:/arms.sse.saveetha.com/");
```



Output:



Result:

Finally analysed and verified the output

EXP.No 12: Automation Testing using Selenium tool for online food application

Aim: Testing online food application by using selenium tools

Algorithm:

Step1 : Start the programme

Step2: Import selenium tool in python editor

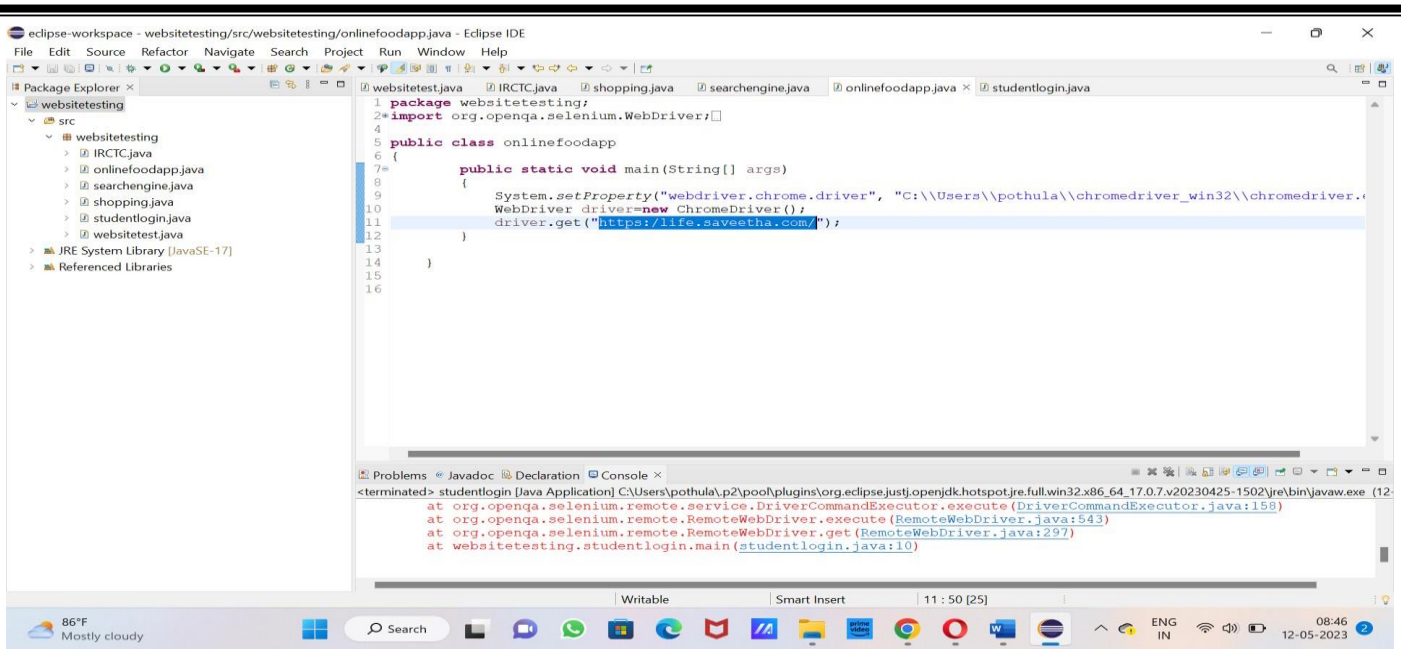
Step3 : Enter user name and password

Step4: Check the authentication in the website

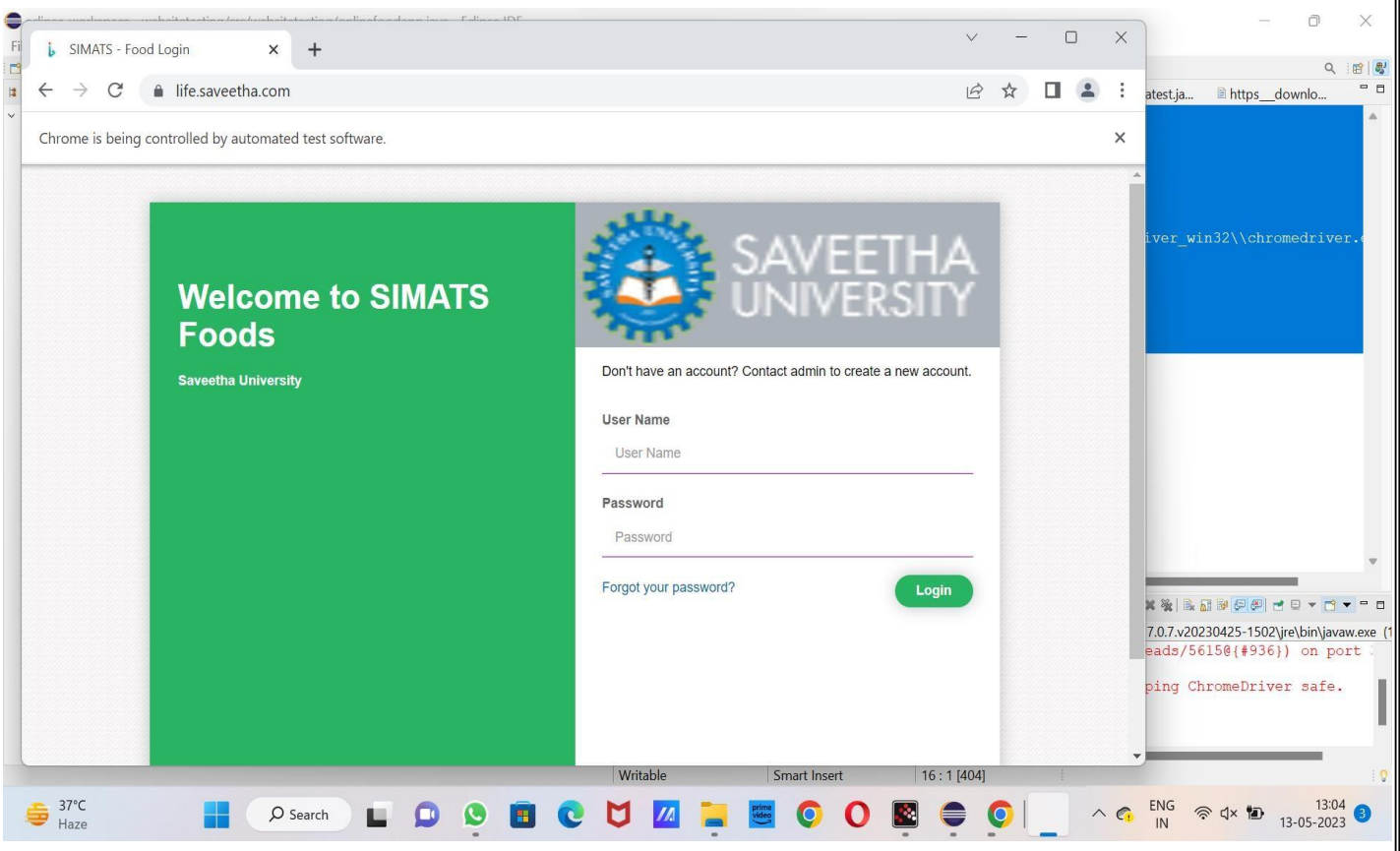
Step5: End the programme.

Code:

```
package websitetesting;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;


public class onlinefoodapp

{

        public static void main(String[] args)

        {

                System.setProperty("webdriver.chrome.driver",
"C:\\Users\\pothula\\chromedriver_win32\\chromedriver.exe");

                WebDriver driver=new ChromeDriver();

                driver.get("https:/life.saveetha.com/");

        }

}
```



OUT PUT:



Result:

Finally analysed and verified the output

EXP.No 13: Automation Testing using Selenium tool for Ticket Reservation System(IRCTC)

Aim:

Testing Ticket Reservation System (IRCTC) By Using Selenium Tools

Algorithm:

Step1: Start the programme

Step2: Import selenium tool in python editor

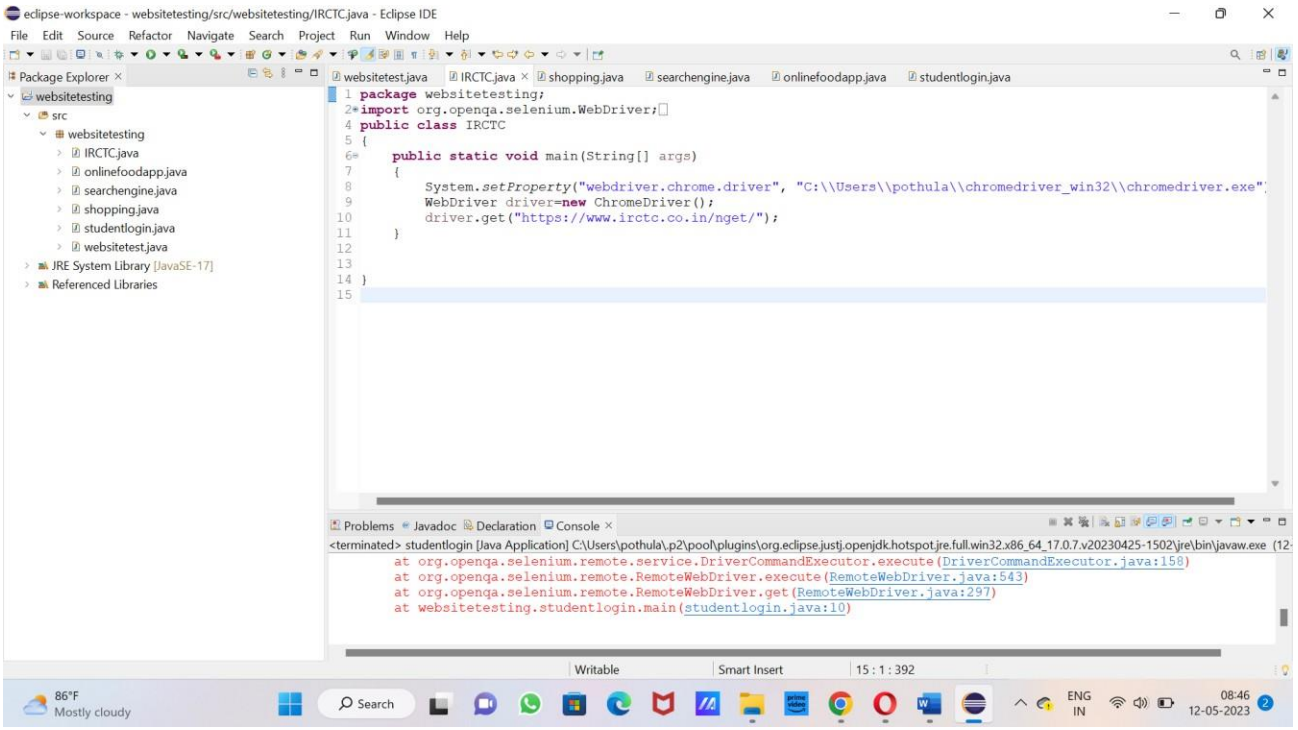
Step3: Enter user name and password

Step4: Check the authentication in the website

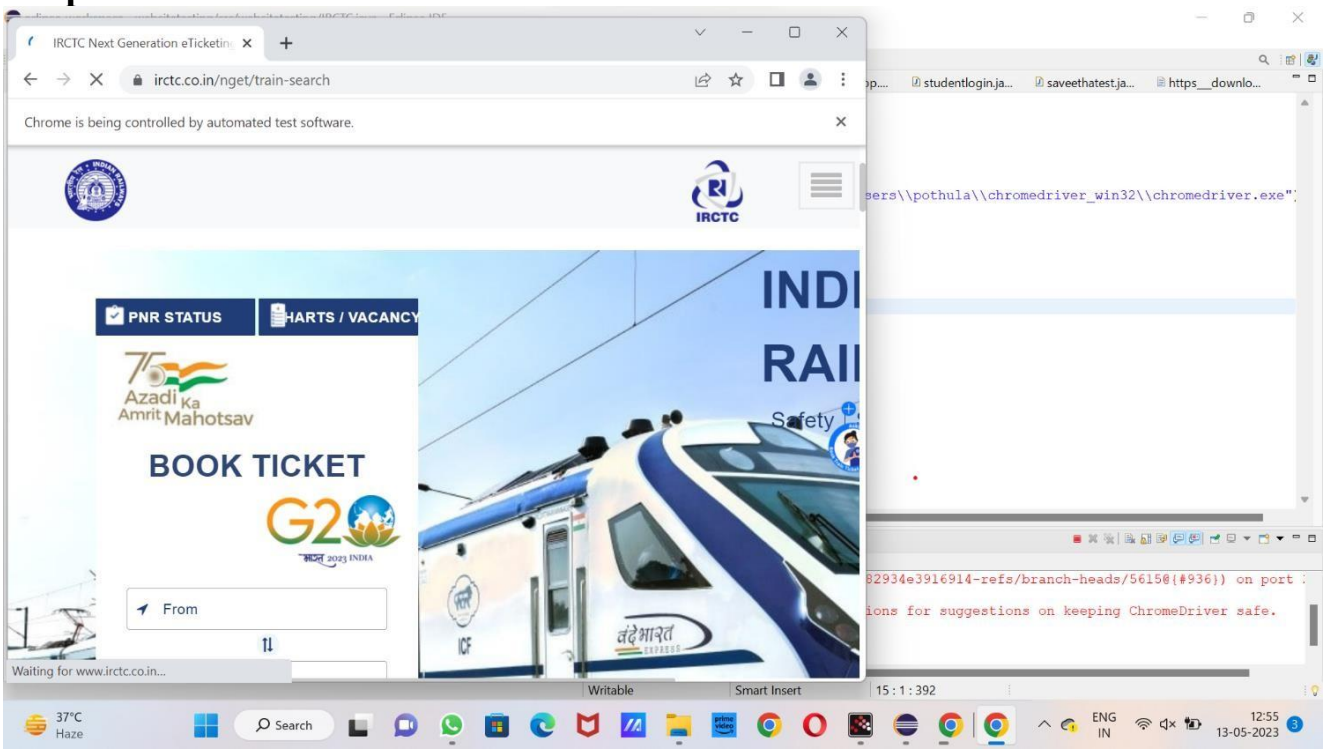
Step5: End the programme.

Code:

```
package websitetesting;

import org.openqa.selenium.WebDriver;

importorg.openqa.selenium.chrome.ChromeD

river;

public class IRCTC

{

      public static void main(String[] args)

      {

System.setProperty("webdriver.chrome.driver

",”C:\\Users\\pothula\\chromedriver_win32\\c

homedriver.exe");

WebDriver   driver=newChromeDriver();

driver.get("https://www.irctc.co.in/nget/);

      }

}
```



Output:



Result:

Finally analyzed and verified the output

EXP.No 14: Automation Testing using Selenium tool for Shopping Domain

Aim:

Testing Shopping Domain Using Selenium tool.

Algorithm:

Step1: Start the programme

Step2: Import selenium tool in python editor

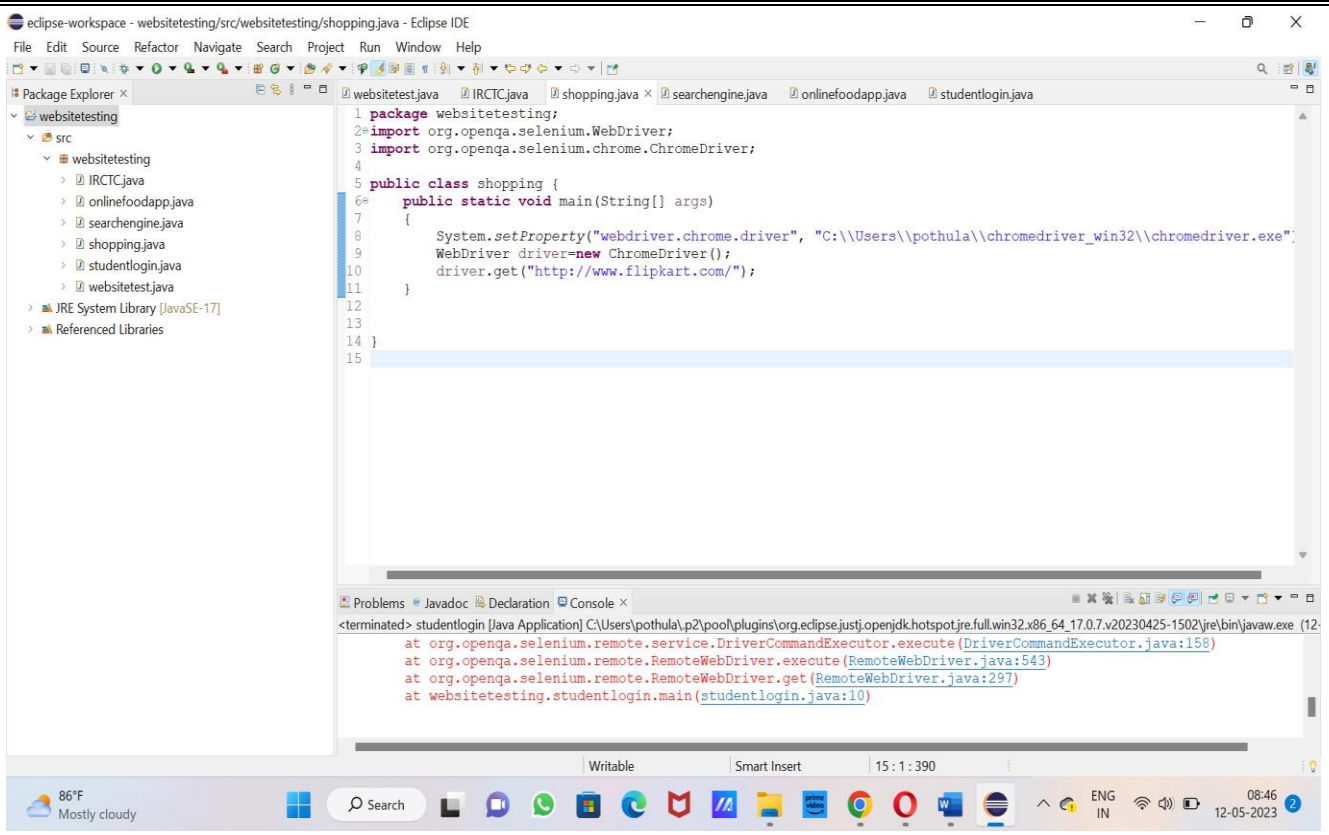
Step3: Enter user name and password

Step4: Check the authentication in the website

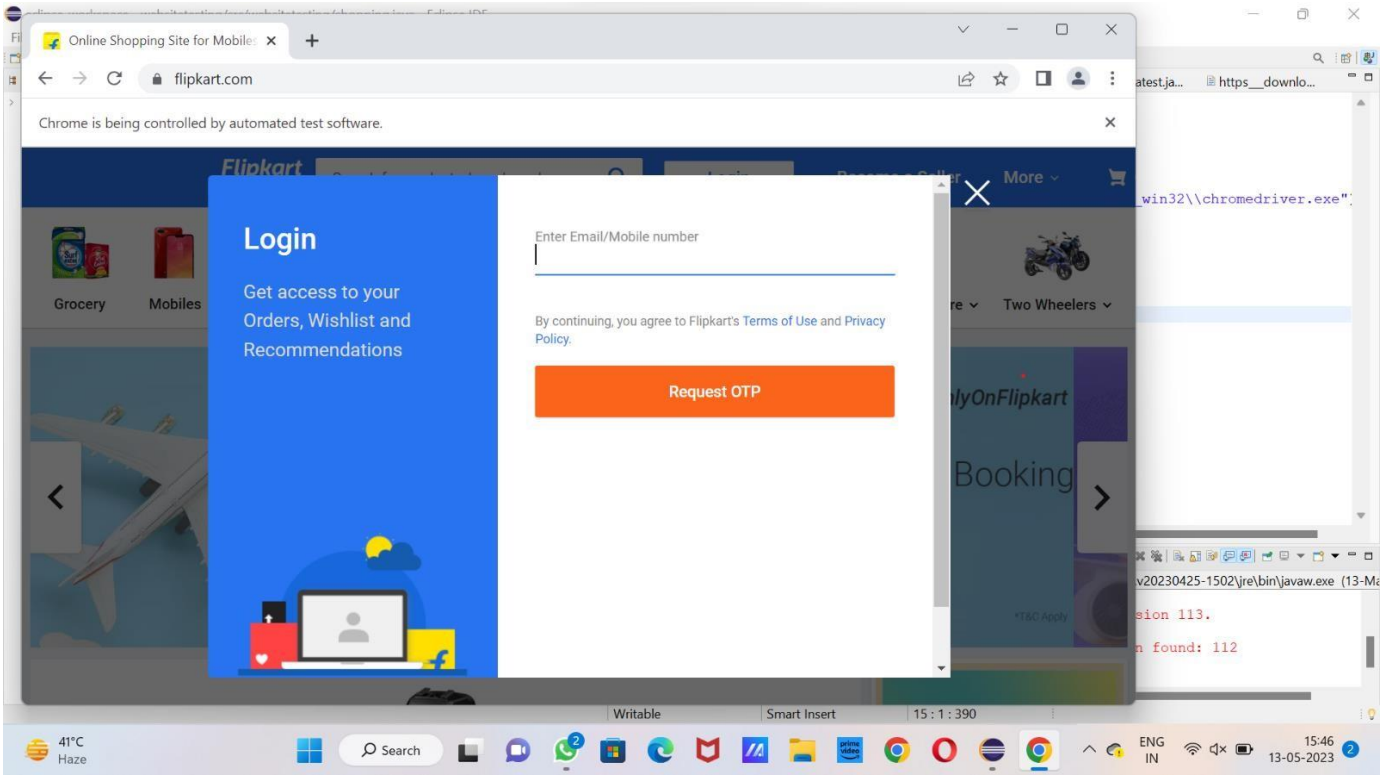
Step5: End the programme.

Code:

```
package websitetesting;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class shopping {
        public static void main(String[] args)
        {
System.setProperty("webdriver.chrome.driver","C:\\Users\\pothula\\chromedriver_win32\\chromedriver.exe");
WebDriver driver=new ChromeDriver();
driver.get("http://www.flipkart.com/");
        }
}
```



Output:



Result:

Finally analyzed and verified the output.

EXP.No 15: Automation Testing using Selenium tool for Opening any Search Engine (EX-google.com, yahoo.com)

Aim:

Testing Opening any Search Engine (ex-google.com, yahoo.com)

Algorithm:

Step1: Start the programme

Step2: Import selenium tool in python editor

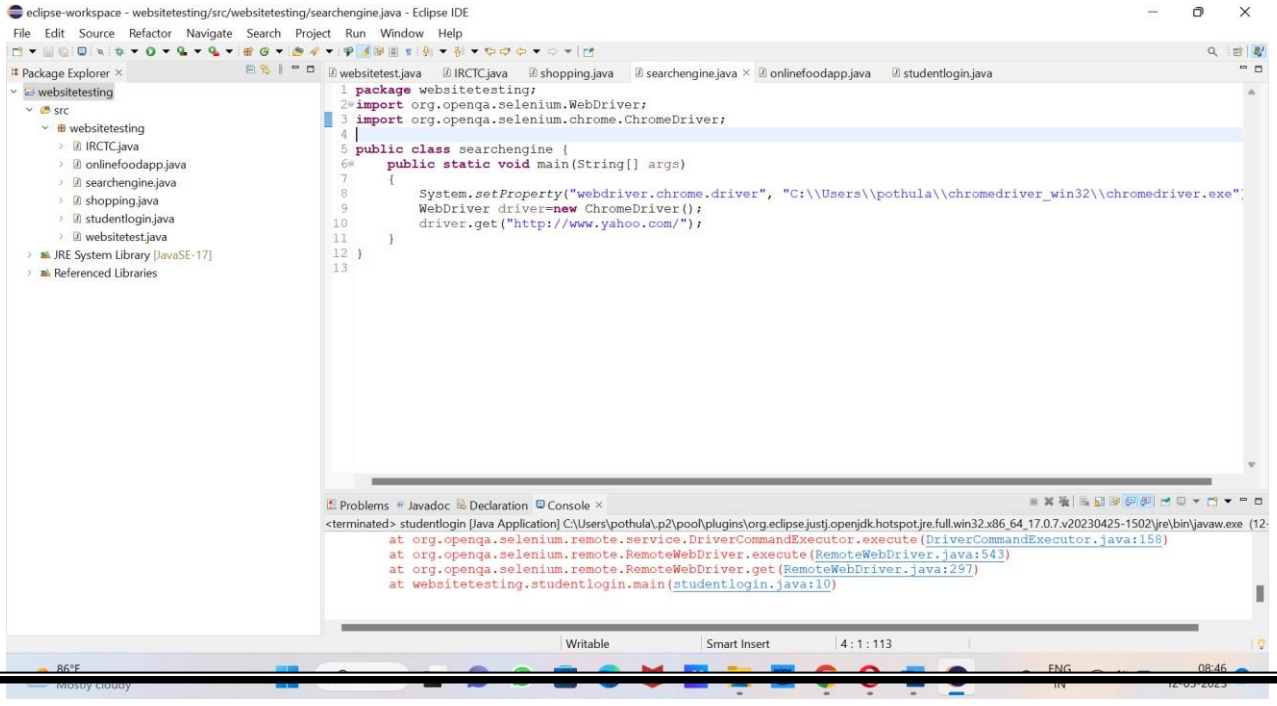
Step3: Enter user name and password

Step4: Check the authentication in the website

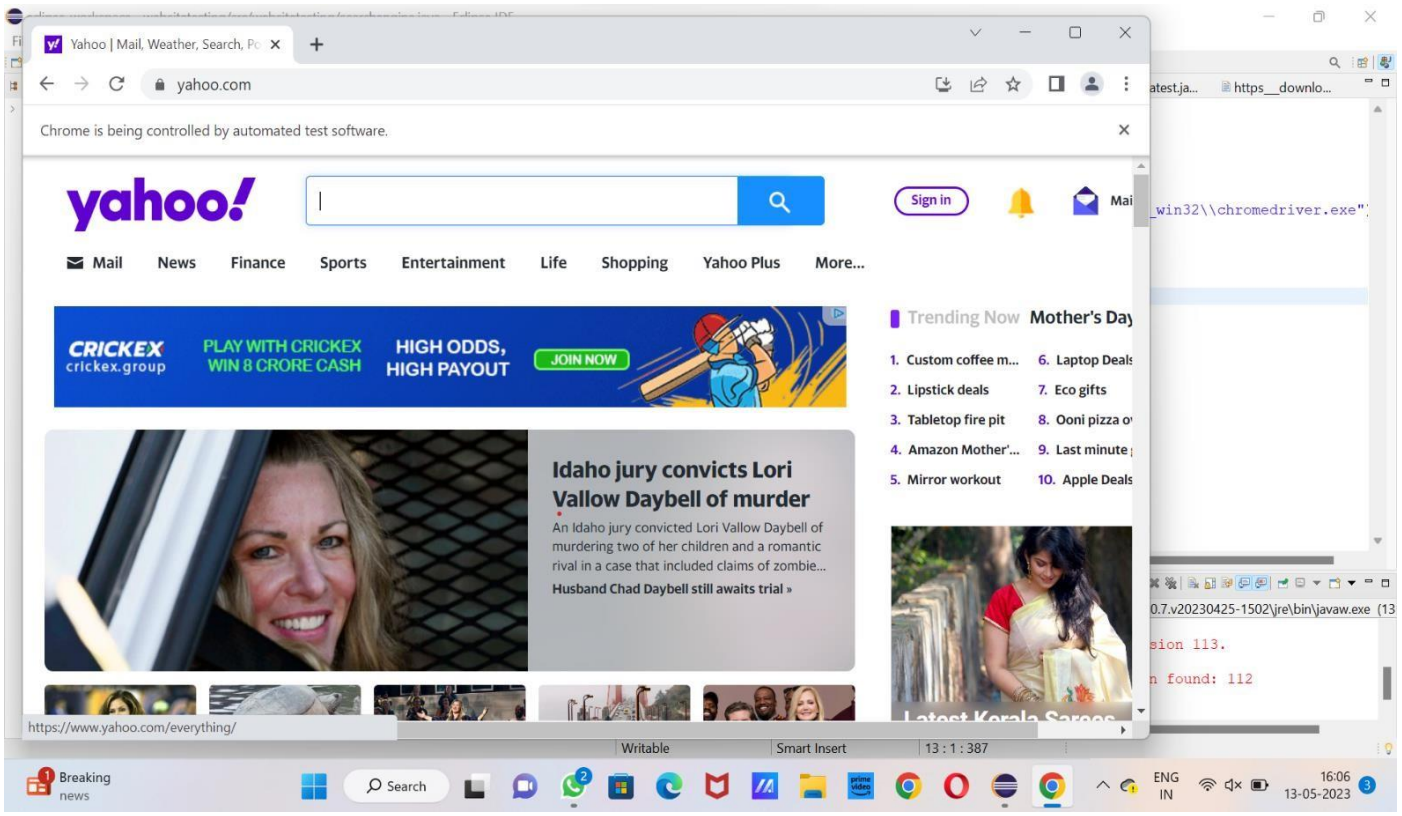
Step5: End the programme.

Code:

```
package websitetesting;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class searchengine {
        public static void main(String[] args)
        {
System.setProperty("webdriver.chrome.driver","C:\\Users\\pothula\\chromedriver_win32\\chromedriver.exe");
WebDriver driver=new ChromeDriver();
driver.get("http://www.yahoo.com/");
        }
}
```



Output:



Result:

Finally analyzed and verified the output.

EXP.No 16: Automation Testing using Selenium tool for any Social Media Platform

Aim:

Testing Social Media Platform Using Selenium tool

Algorithm:

Step1: Start the programme

Step2: Import selenium tool in python editor

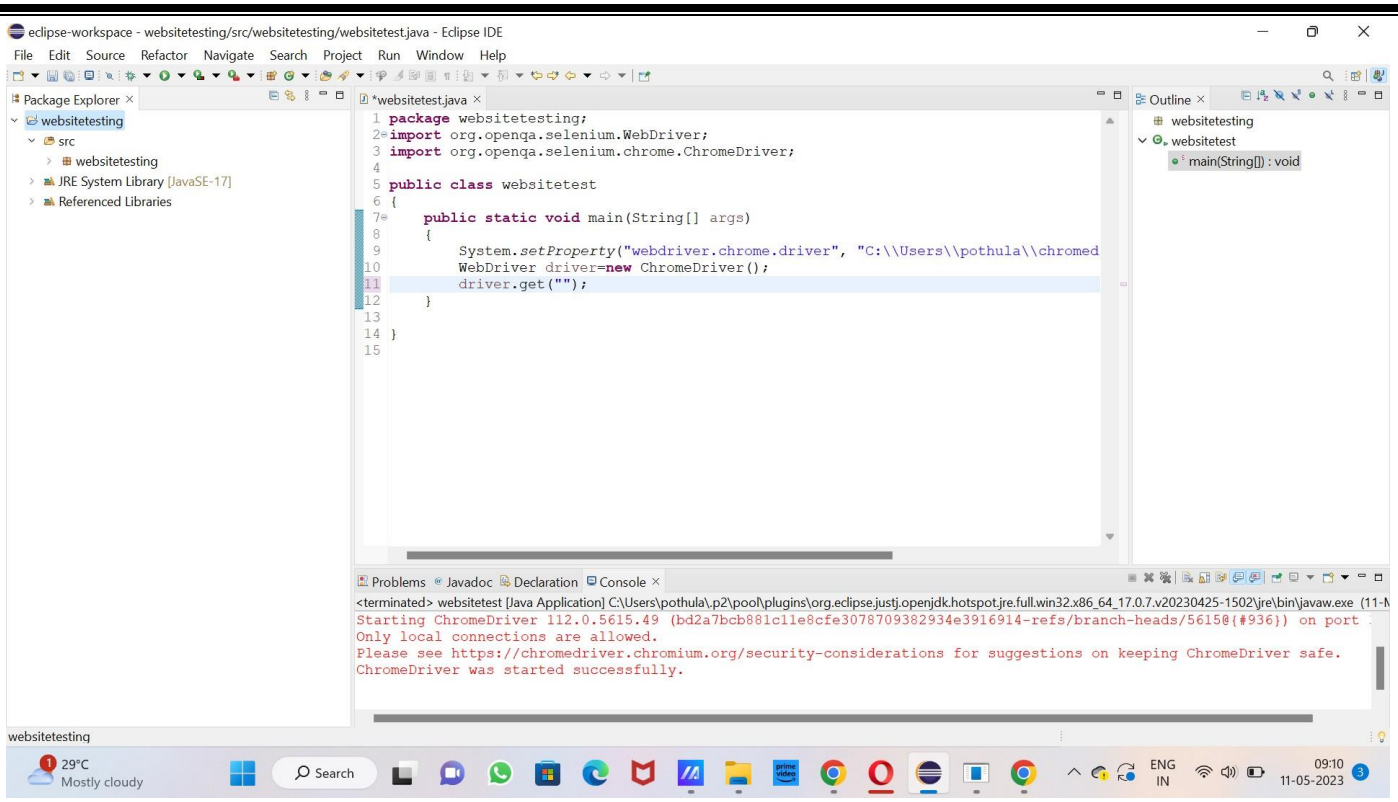
Step3: Enter user name and password

Step4: Check the authentication in the website

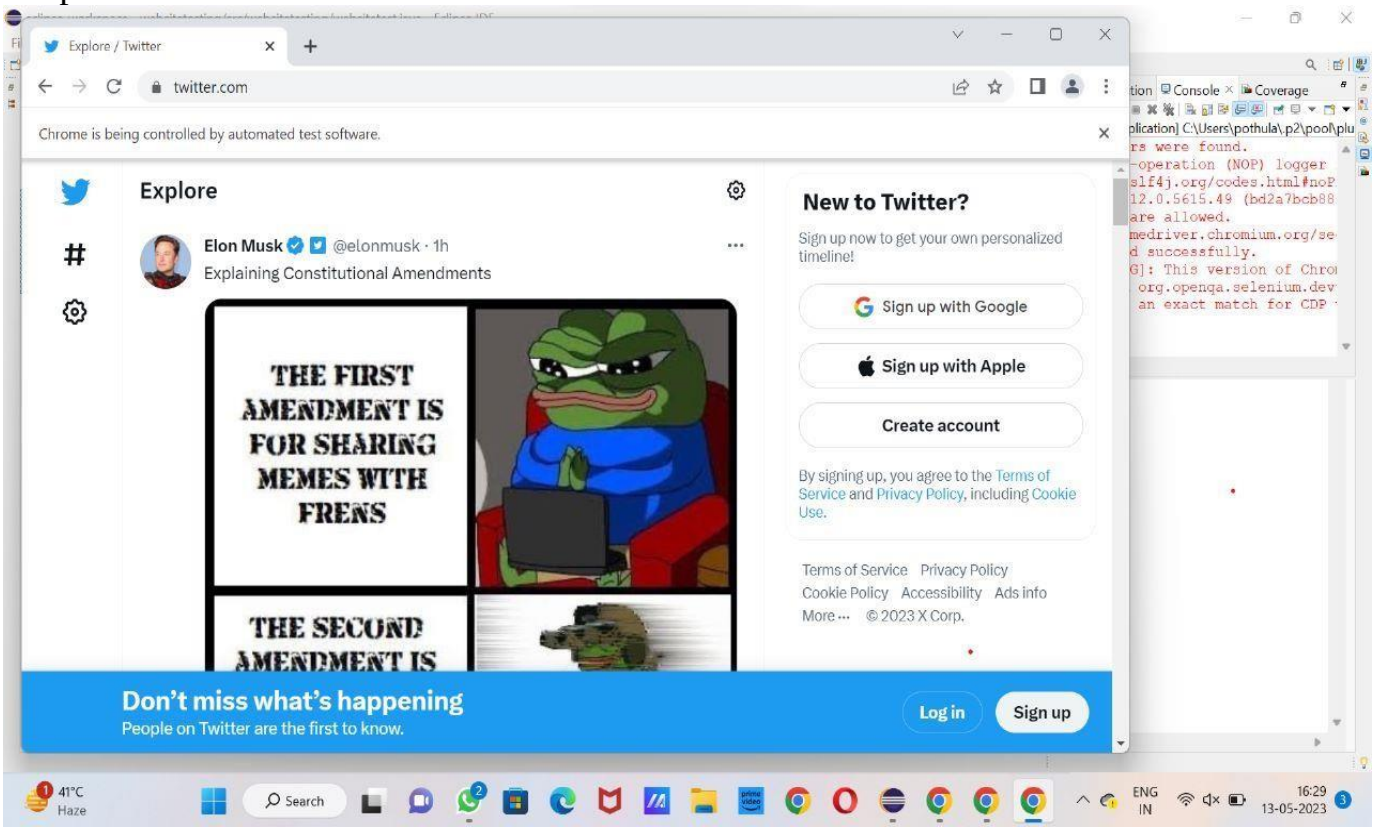
Step5: End the programme.

Code:

```
package websitetesting;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class websitetest
{
	public static void main (String[] args)
	{
System.setProperty("webdriver.chrome.driver","C:\\Users\\pothula\\chromedriver_win32\\chromedriver.exe");
	WebDriver driver=new ChromeDriver();
	driver.get("https://www.twitter.com/");
	}

}
```



Output:



Result:

Finally analyzed and verified the output.

EXP.No 17: Using Junit Testing Check whether the given string is printed or not

Aim:

Check whether the given string is printed or not

Algorithm:

Step1: Start the programme

Step2: Import selenium tool in python editor

Step3: Enter user name and password

Step4: Check the authentication in the website

Step5: End the programme.

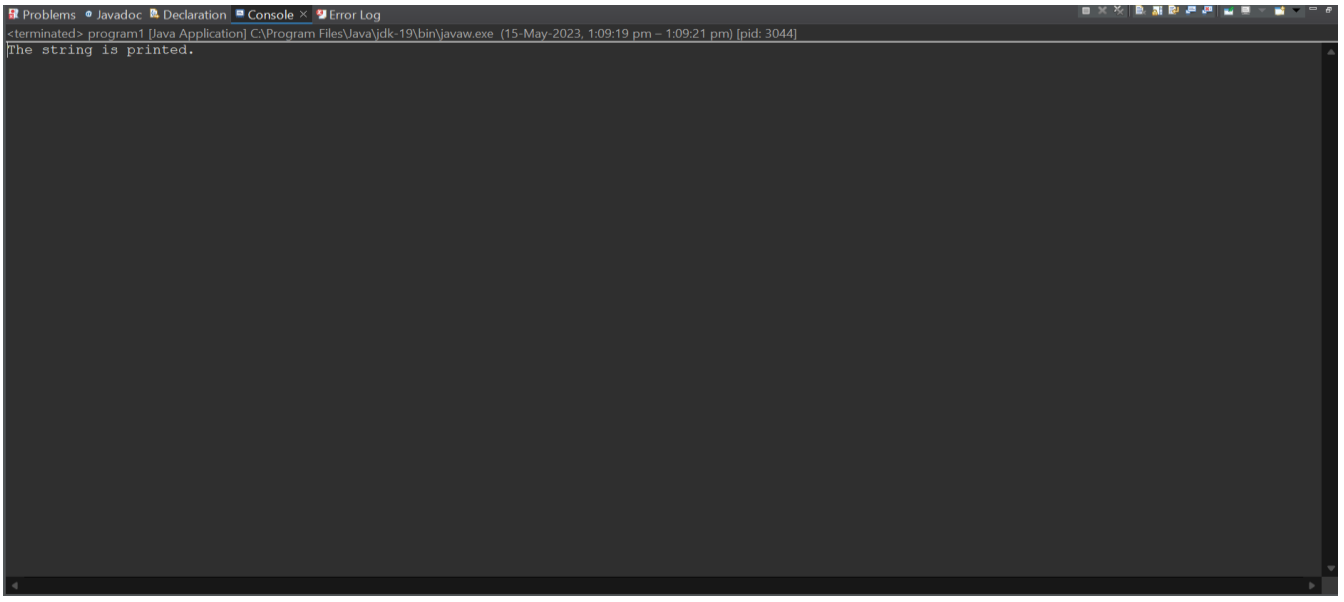
Code:

```
package junit;
import java.io.ByteArrayOutputStream;
import java.io.PrintStream;
public class program1
{
   public static void main(String[] args)
   {
      String myString = "saveetha college";
      String searchString = "saveetha";
      ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
      PrintStream printStream = new PrintStream(outputStream);
      PrintStream originalPrintStream = System.out;
      System.setOut(printStream);
      System.out.println(myString);
      System.setOut(originalPrintStream);
      String consoleOutput = outputStream.toString();
      if (consoleOutput.contains(searchString))
      {
         System.out.println("The string is printed.");
```

```
    } else
    {
        System.out.println("The string is not printed.");
    }
  }
}
```

```java
package junit;
import java.io.ByteArrayOutputStream;
public class program1
{
    public static void main(String[] args)
    {
        String myString = "saveetha college";
        String searchString = "saveetha";
        ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
        PrintStream printStream = new PrintStream(outputStream);
        PrintStream originalPrintStream = System.out;
        System.setOut(printStream);
        System.out.println(myString);
        System.setOut(originalPrintStream);
        String consoleOutput = outputStream.toString();
        if (consoleOutput.contains(searchString))
        {
            System.out.println("The string is printed.");
        } else
        {
            System.out.println("The string is not printed.");
        }
    }
}
```

Output:



EXP.No 18: Using Junit Testing Check Whether the given string is getting Reversed or not

Aim:

Check Whether the given string is getting Reversed or not

Algorithm:

Step1: Start the programme

Step2: Import selenium tool in python editor

Step3: Enter user name and password

Step4: Check the authentication in the website

Step5: End the programme.

Code:

```
package junit;
import java.io.ByteArrayOutputStream;
import java.io.PrintStream;
public class string1
{
	public static void main(String[] args)
	{
		String originalString =
"malayalam";
		String reversedString =
string1(originalString);
		if
(originalString.equals(reversedString))
		{
		   System.out.println("The string
is getting reversed.");
		}
```

```
            else
            {
                System.out.println("The string
is not getting reversed.");
            }
        }
        private static String string1(String
str)
        {
            StringBuilder reversed = new
StringBuilder(str);
            reversed.reverse();
            return reversed.toString();
        }
}
```

```java
package junit;
import java.io.ByteArrayOutputStream;
public class string1
{
    public static void main(String[] args)
    {
        String originalString = "malayalam";
        String reversedString = string1(originalString);
        if (originalString.equals(reversedString))
        {
            System.out.println("The string is getting reversed.");
        }
        else
        {
            System.out.println("The string is not getting reversed.");
        }
    }
    private static String string1(String str)
    {
        StringBuilder reversed = new StringBuilder(str);
        reversed.reverse();
        return reversed.toString();
    }
}
```

OUTPUT:

```
<terminated> string1 [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe  (15-May-2023, 1:22:26 pm – 1:22:28 pm) [pid: 9188]
The string is getting reversed.
```

EXP.No 19: Using Junit Testing Check Whether the given Username is Valid or not

Aim:

Check Whether the given Username is Valid or not

Algorithm:

Step1: Start the programme

Step2: Import selenium tool in python editor

Step3: Enter user name and password

Step4: Check the authentication in the website

Step5: End the programme.

Code:

```
package junit;
import java.io.*;
import java.util.*;
public class string1
{
	public static void main(String[] args)
	{
		String a,b;
		Scanner s=new
Scanner(System.in);
		System.out.println("Enter the
User name :");
		a=s.nextLine();
		System.out.println("Reenter
the User name :");
		b=s.nextLine();
		if(a.equals(b))
		{
```

```
        System.out.println("User is Valid");
                }
                else
                {

        System.out.println("User is
Inv8oalid");
                }
        }
}
```

```java
package junit;
import java.io.*;
import java.util.*;
public class string1
{
    public static void main(String[] args)
    {
        String a,b;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the User name :");
        a=s.nextLine();
        System.out.println("Reenter the User name :");
        b=s.nextLine();
        if(a.equals(b))
        {
            System.out.println("User is Valid");
        }
        else
        {
            System.out.println("User is Inv8oalid");
        }
    }
}
```

OUTPUT:

```
Enter the User name :
saveetha123
Reenter the User name :
saveetha321
User is Invalid
```

EXP.No 20: Using Junit Testing Check Whether the Calculated simple interest is valid or not

Aim:

Check Whether the Calculated simple interest is valid or not

Algorithm:

Step1: Start the programme

Step2: Import selenium tool in python editor

Step3: Enter user name and password

Step4: Check the authentication in the website

Step5: End the programme.

Code:

```
package junit;
import java.util.Scanner;
public class string1
{
    public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter principal amount: ");
    double principal = scanner.nextDouble();

    System.out.print("Enter rate of interest: ");
    double rate = scanner.nextDouble();

    System.out.print("Enter time (in years): ");
    double time = scanner.nextDouble();

    double simpleInterest = (principal * rate * time) / 100;

    if (simpleInterest > 0) {
```

```
            System.out.println("The calculated simple interest " +simpleInterest+ " is valid");
        } else {
            System.out.println("The calculated simple interest is not valid.");
        }

        scanner.close();
    }
}
```

```
package junit;
import java.util.Scanner;
public class string1
{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter principal amount: ");
        double principal = scanner.nextDouble();

        System.out.print("Enter rate of interest: ");
        double rate = scanner.nextDouble();

        System.out.print("Enter time (in years): ");
        double time = scanner.nextDouble();

        double simpleInterest = (principal * rate * time) / 100;

        if (simpleInterest > 0) {
            System.out.println("The calculated simple interest " +simpleInterest+ " is valid");
        } else {
            System.out.println("The calculated simple interest is not valid.");
        }

        scanner.close();
    }
```

OUTPUT:

```
Enter principal amount: 20000
Enter rate of interest: 10
Enter time (in years): 2
The calculated simple interest 4000.0 is valid
```

EXP.No 21: Using Junit Testing check whether the given number is palindrome or not

Aim:

check whether the given number is palindrome or not

Algorithm:

Step1: Start the programme

Step2: Import selenium tool in python editor

Step3: Enter user name and password

Step4: Check the authentication in the website

Step5: End the programme.

Code:

```
package junit;
import java.util.Scanner;
public class string1
{
	public static void main(String arg[])
	{
		String s,b="";
		char d;
		Scanner c=new
Scanner(System.in);
		System.out.print("Enter a
Number :");
		s=c.nextLine();
		for(int i=0;i<s.length();i++)
		{
			d=s.charAt(i);
			b=d+b;
		}
```

```
            if(s.equals(b))
            {
                System.out.print("The
given number is palindrome");
            }
            else
            {
                System.out.print("The
given number is not palindrome");
            }
    }
}
```

```java
package junit;
import java.util.Scanner;
public class string1
{
    public static void main(String arg[])
    {
        String s,b="";
        char d;
        Scanner c=new Scanner(System.in);
        System.out.print("Enter a Number :");
        s=c.nextLine();
        for(int i=0;i<s.length();i++)
        {
            d=s.charAt(i);
            b=d+b;
        }
        if(s.equals(b))
        {
            System.out.print("The given number is palindrome");
        }
        else
        {
            System.out.print("The given number is not palindrome");
        }
    }
}
```

OUTPUT:

```
Enter a Number :121
The given number is palindrome
```