B.Sc. (Hons) Computer Applications
Department of Computer Science
Aligarh Muslim University

CCB-6S4: Senior Project II

Title: Algorithm Visualizer

Senior Supervisor: Dr.Faisal Anwer

Submitted by: Haris Moin **Faculty Roll no.: 20CAB106** **Enrolment no.: GK2110**

Algorithm Visualizer

- Responsive web-application
- Provides an Interactive way for visualizing the logic behind algorithms or snippet of code.
- Gives its users a chance to contribute in the project

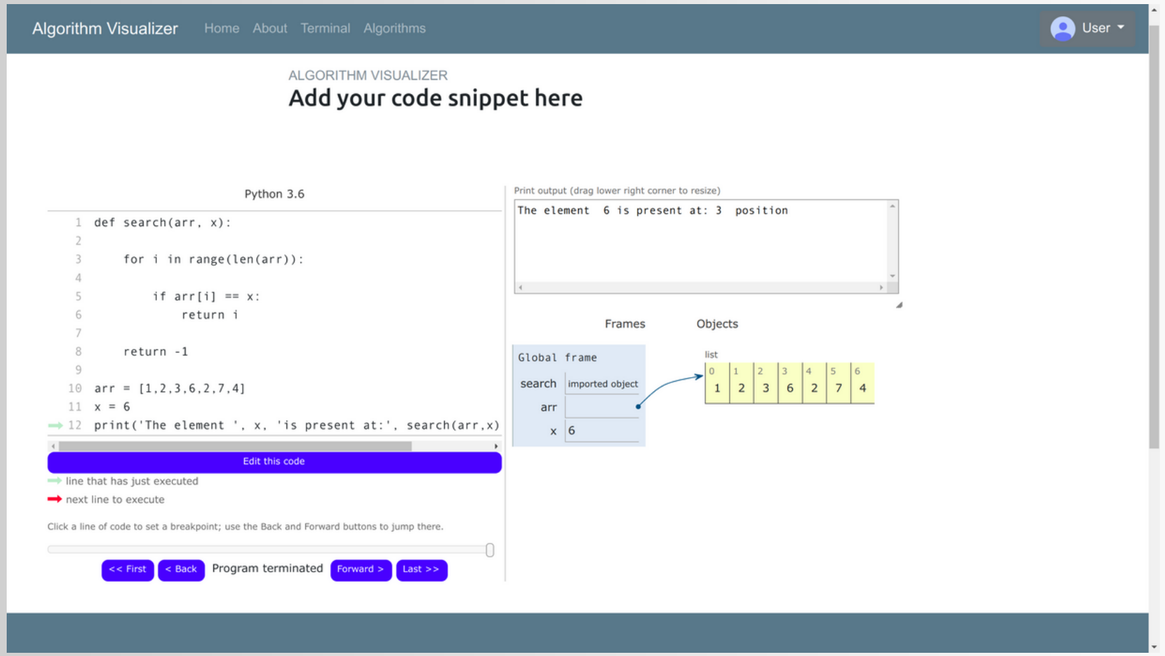


Table of Content

Problems Faced

- Variety of paradigms leads to Code versatility
- Hands-on editing of code is rarely seen with its pictorial representation
- Let down feeling amongst beginners

How this application deals with these problems

Functionalities

- **Contributor Registration:** Being a part of the development community of this project and helping others learn. Users could sign up as contributor and be part of the cause.

- **Make Contribution**: Once a user is registered, they could add their snippet of codes and provide their descriptions which would later be displayed in the documentation alongside all algorithms in the visualizer.

- **Access the terminal:** at any point you find yourself in a logical turmoil regarding a piece of code, try it out on the terminal and see what it shows

- **Search**: the visualizer would provide a search feature to filter algorithms by their category which is very helpful in different scenarios.

- **Description**: when the request of an algorithm is recieved by th visualizer, it would typically respond back with a brief overview of that algorithm provided by its contributor and code which could be directly executed.

- **Update or Delete post:** even if some user doesn't possesses administrative features, they will at least be able to delete or update their contributed posts from the visualizer.

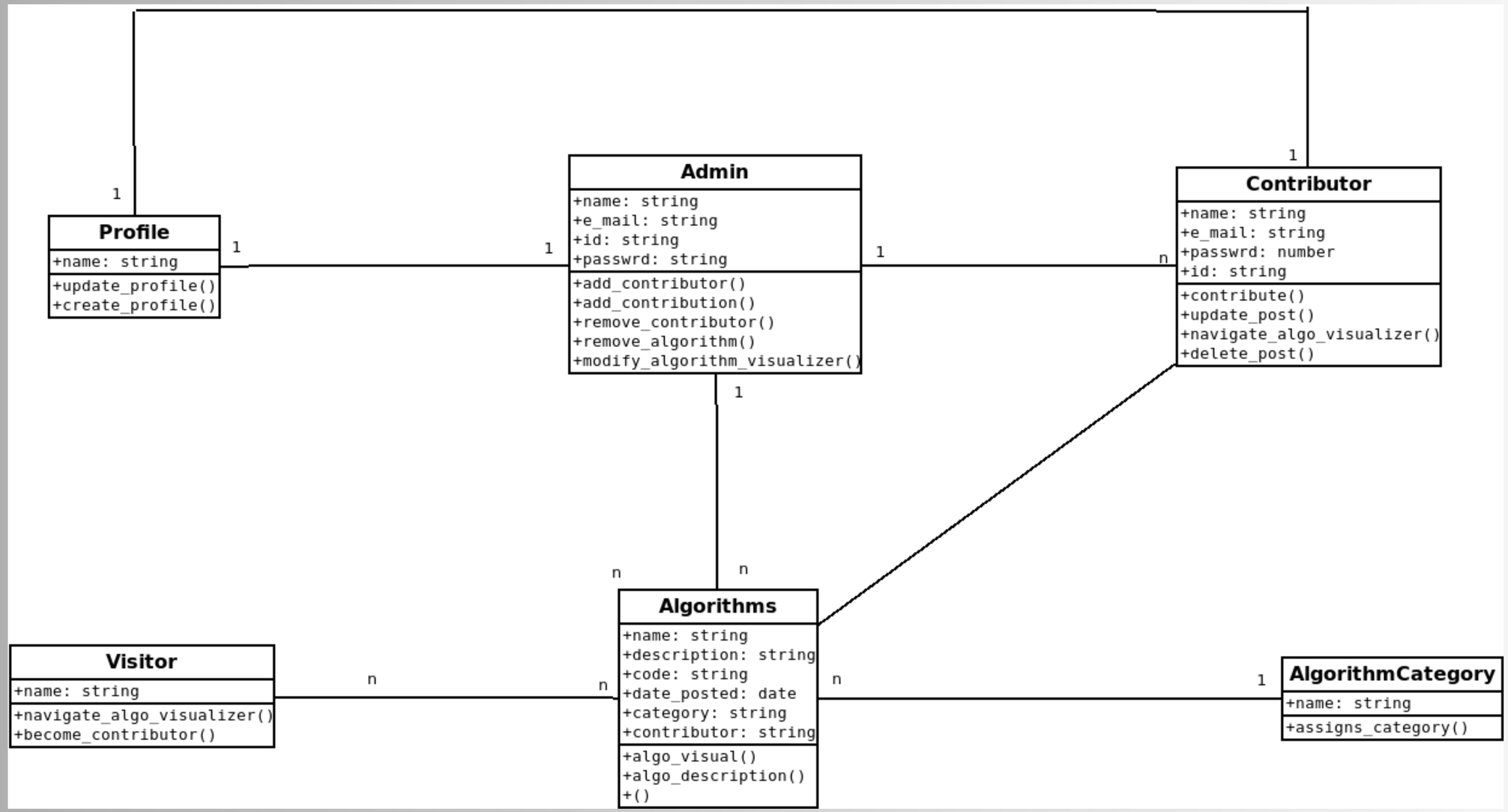
Build of the Project

- For the development of this system, **Django (framework of python), HTML, CSS, JavaScript and bootstrap** is used. Also for the intermediary required by the terminal for its code execution the use of common gateway interface(CGI) was done

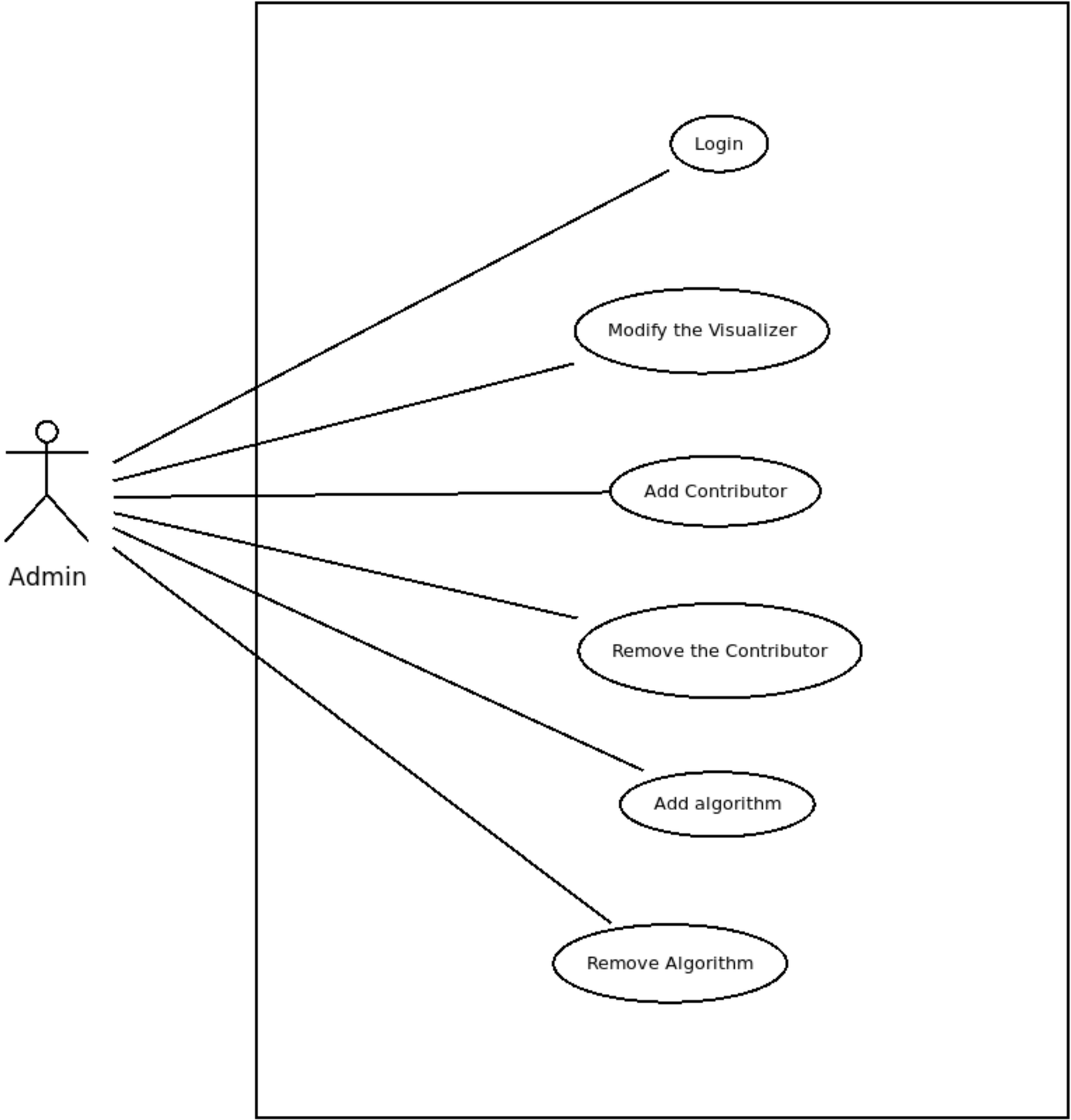
- This web based system will be deployed and maintained on a online hosting platform for better **Accessibility and Scalability**

- The System will be Accessible with any common device which supports a browser and have internet connectivity

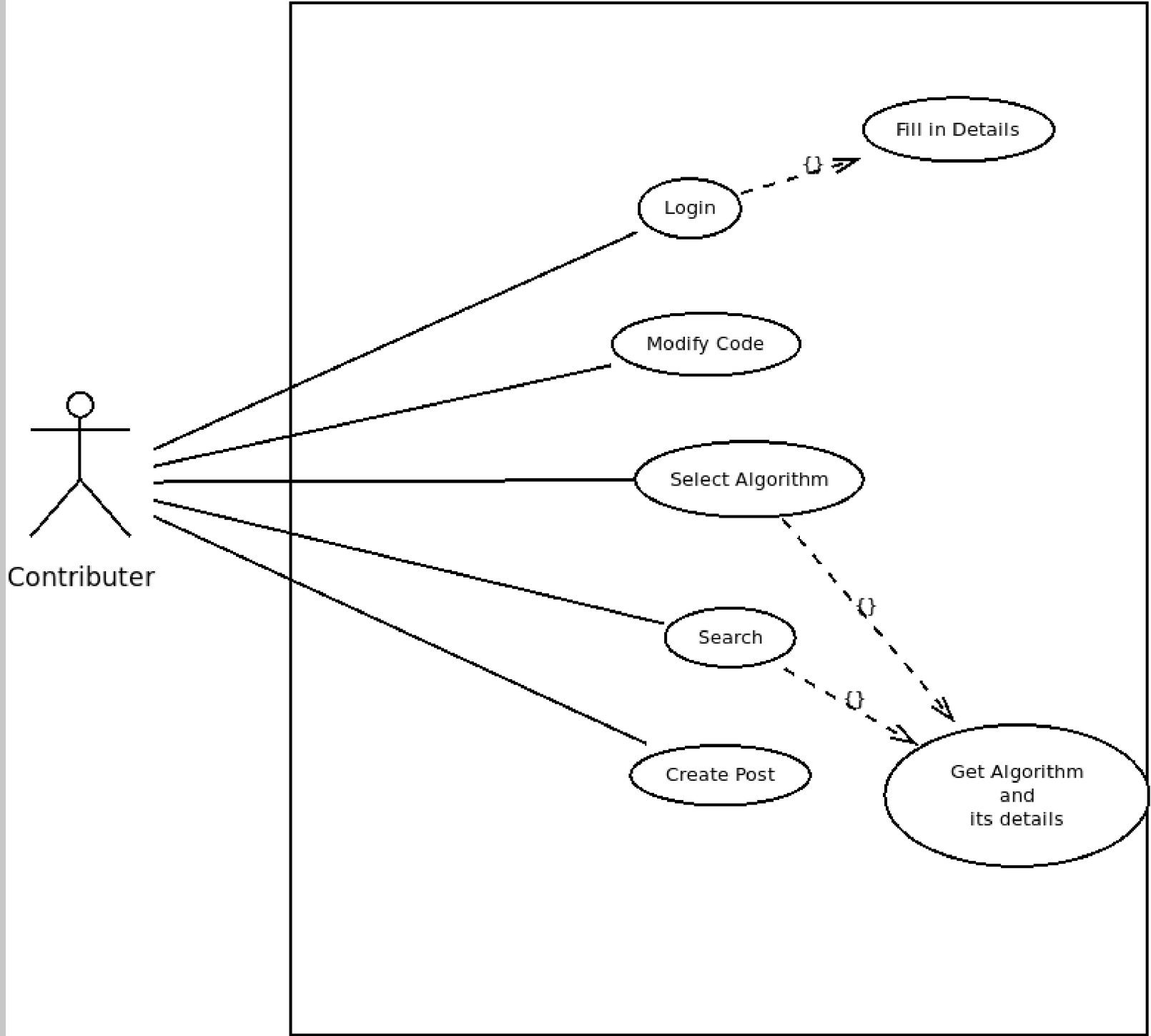
Schema Diagram



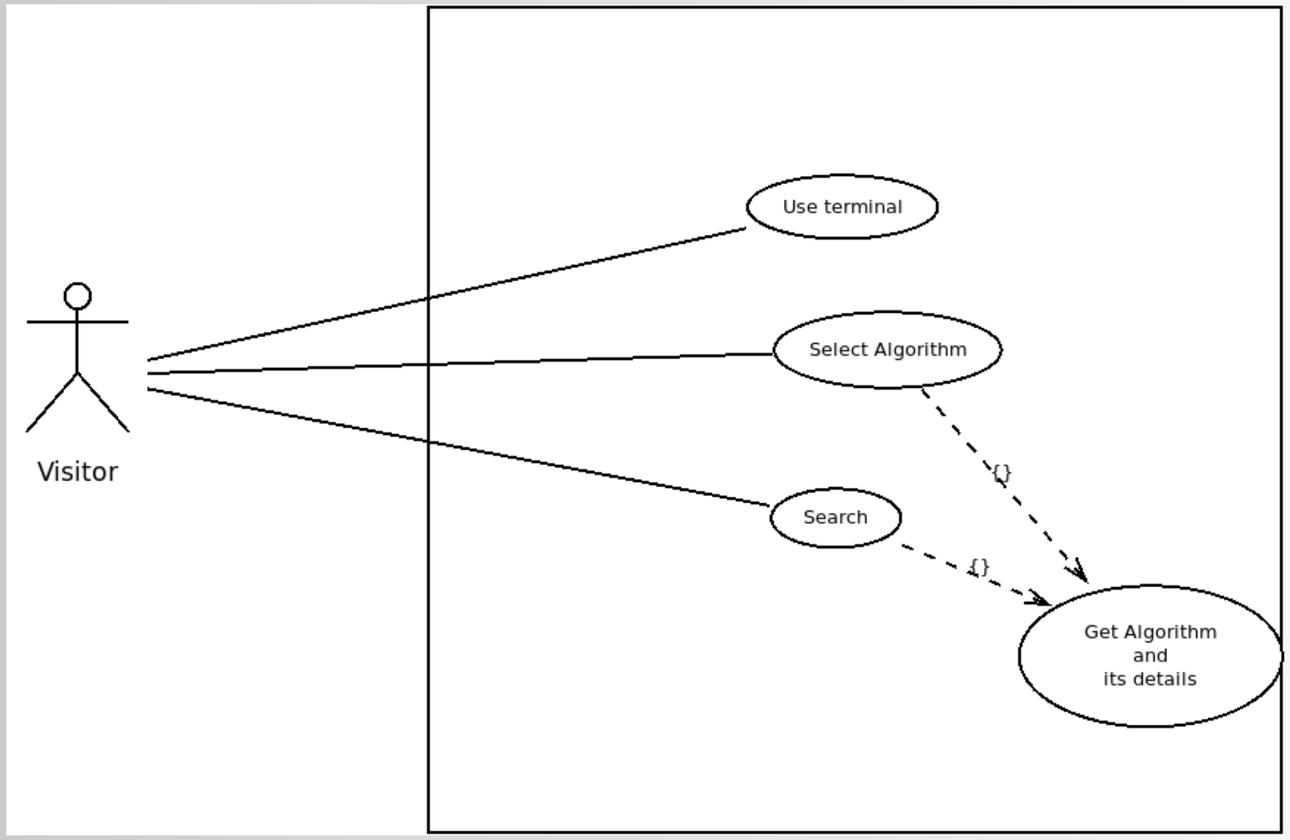
Admin
Use Case



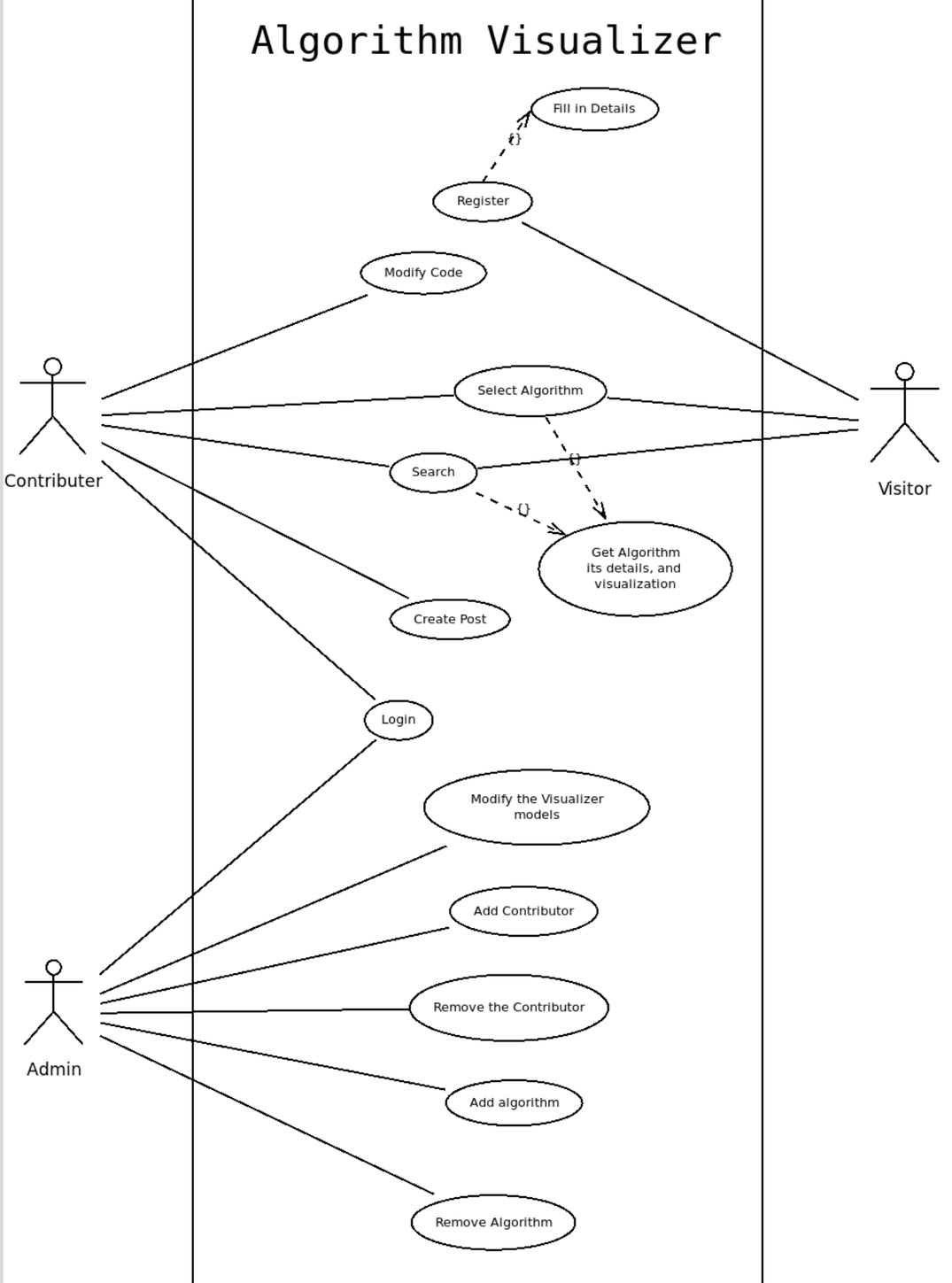
Contributor Use Case



Visitor
Use Case

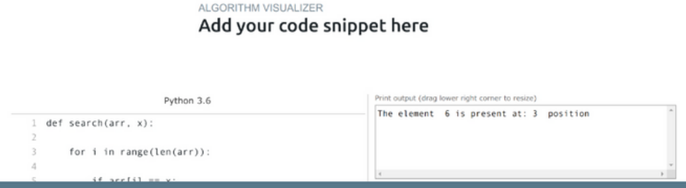


Main
Use Case



Go Straight to the Terminal

An interactive code visualizing application equiped with loads of algorithms



About Us

Algorithm Visualizer is an interactive web-based platform enriched with lots of algorithms and code snippets contributed by its users which range in a categories such as Greedy algorithms, Divide and Conquer algorithms and many more. This platform serves its users with the hands-on visual representation of these algorithms through the means of a javascript library named visualize.bundle.js which is most famously used for building flow charts and roadmaps by multiple websites also making there contribution to this cause of learning through the means of graphical representation as *"a picture speaks more than a thousand words".*

Become a registered user & Contribute

Sign Up

ALGORITHM VISUALIZER

Add your code snippet here

Write code in Python 3.6

```
1  def factorial(n):
2
3      return 1 if (n==1 or n==0) else n * factorial(n - 1)
4
5
6  num = 5
7  print("Factorial of",num,"is",factorial(num))
```

Visualize Execution

ALGORITHM VISUALIZER

Add your code snippet here

Python 3.6

```
1  def search(arr, x):
2
3      for i in range(len(arr)):
4
5          if arr[i] == x:
6              return i
7
8      return -1
9
10 arr = [1,2,3,6,2,7,4]
11 x = 6
12 print('The element ', x, 'is present at:', search(arr,x)
```

Edit this code

line that has just executed

next line to execute

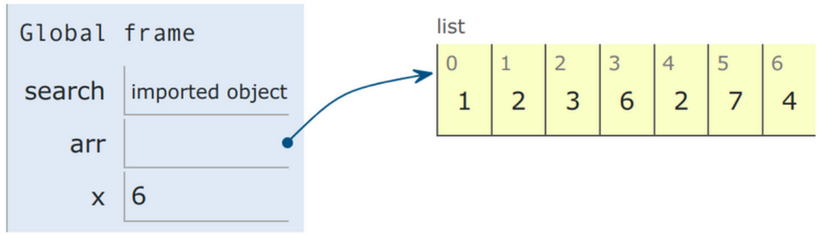
Click a line of code to set a breakpoint; use the Back and Forward buttons to jump there.

<< First | < Back | Program terminated | Forward > | Last >>

Print output (drag lower right corner to resize)

```
The element  6 is present at: 3  position
```

Frames Objects



Search by category: All algorithms Search

April 14, 2023

Tower of Hanoi

Tower of Hanoi is a mathematical puzzle where we have three rods (A, B, and C) and N disks. Initially, all the disks are stacked in decreasing value of diameter i.e., the smallest disk is placed on the top and they are on rod A. The objective of the puzzle is to move the entire stack to another rod (here considered C), obeying the following simple rules:

- Only one disk can be moved at a time.
- Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
- No disk may be placed on top of a smaller disk.

Worst Case Time Complexity: O(2^N)
Best Case Time Complexity: O(2^N)

April 14, 2023

Selection Sort

Selection sort is a simple and efficient sorting algorithm that works by repeatedly selecting the smallest (or largest) element from the unsorted portion of the list and moving it to the sorted portion of the list. The algorithm repeatedly selects the smallest (or largest) element from the unsorted portion of the list and swaps it with the first element of the unsorted portion. This process is repeated for the remaining unsorted portion of the list until the entire list is sorted.

April 14, 2023 Update Delete

Binary Search

Binary Search is a searching algorithm used in a sorted array by repeatedly dividing the search interval in half.

Worst-case time complexity: O(log n)
Best-case time complexity: O(1)

```
def binarySearch(arr, l, r, x):


if r >= l:

mid = l + (r - l) // 2


if arr[mid] == x:
return mid


elif arr[mid] > x:
return binarySearch(arr, l, mid-1, x)
```

Visualize

Become a Contributor

Username*

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email*

Password*

- Your password can’t be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can’t be a commonly used password.
- Your password can’t be entirely numeric.

Password confirmation*

Enter the same password as before, for verification.

Sign Up

Already Have An Account? Sign In

Login

Username*

Password*

Login Now

Already Have An Account? Sign Up

harismoin

harismoin88@gmail.com

Profile Info

Username*

harismoin

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email*

harismoin88@gmail.com

Image*

Currently: profile_pics/13232937_1073987026026779_3685797144781879523_n_zWr5TQA.jpg

Change:

Choose File No file chosen

Update

Site administration

ALGOVIZ		
Algorithm categorys	+ Add	Change
Algorithms	+ Add	Change

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

CONTRIBUTORS		
Profiles	+ Add	Change

Recent actions

My actions

- ✖ Merge Sort2
  Algorithm
- ✖ Merge Sort2
  Algorithm
- + Others
  Algorithm category
- ✖ testuser7
  User
- ✖ testuser6
  User
- ✖ testuser5
  User
- ✖ testuser4
  User
- + iharismoin Profile
  Profile
- ✖ testuser9
  User
- ✖ testuser7
  User

Algorithm Post

Name*

Description*

Code*

Category*

Contribute

Guidelines

Follow the given rules in order to contribute:

- Choose a descriptive name
- Write a brief overview of your code. It is recommended to provide the complexities with them as well
- Select the category under which your code might reside
- Provide the code with proper indentations and correct syntax. If any of these are not fulfilled, you might need to update the post.
- Make sure to keep your code compact.

Thank You