**INDEX**

1. Introduction
   1. Project Introduction……………………………………. 3
   2. Purpose of the project………………………………..... 3
2. Information Gathering
3. Benchmark Products………………………………….. 4
4. Feature Comparison………………………………….. 11
5. Feature List
6. Feature List…………………………………………… 12
7. Site Map……………………………………………... 13
8. Feasibility Study
9. SWOT Analysis……………………………………… 85
10. Cash Flow……………………………………………. 86
11. UI Design or MOQUP…………………………………… 15
12. System Design
13. Structural Design
14. Data Flow Design (DFD)……………………. 18
15. Design by UML
16. Context Diagram…………………………….. 20
17. Activity Diagram ……………………………. 21
18. Use Case Diagram………………………….. 24
19. Class Diagram……………………………… 11
20. Sequence Diagram…………………………. 89
21. State Diagram……………………………….. 12
22. Deployment diagram………………………... 87
23. Project Management Tools……………………………….. 12
24. Gantt chart………………………………………………….. 12
25. Future plan…………………………………………………. 12

**INTRODUCTION**

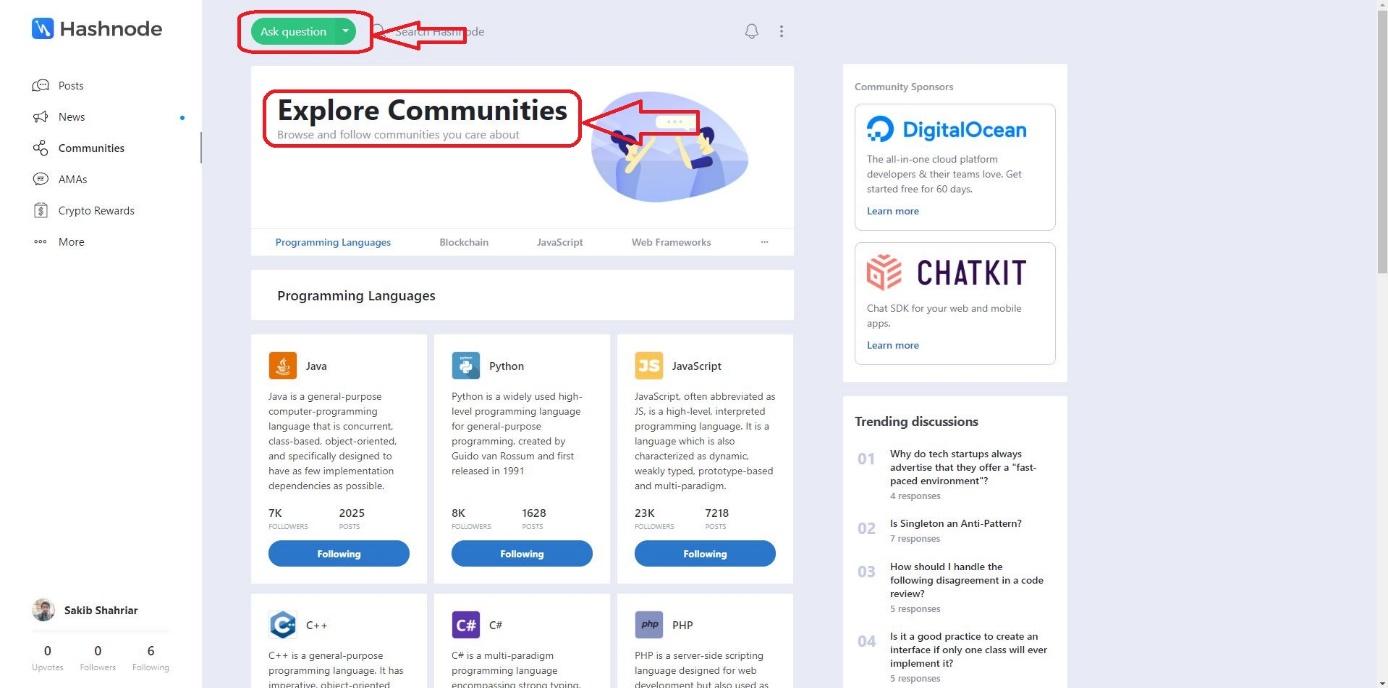
**1. Project Introduction:** Devquery is an online community for developers. This site is made for providing a network for developers so that they can share their problems and get some efficient solution from other developers’ people.

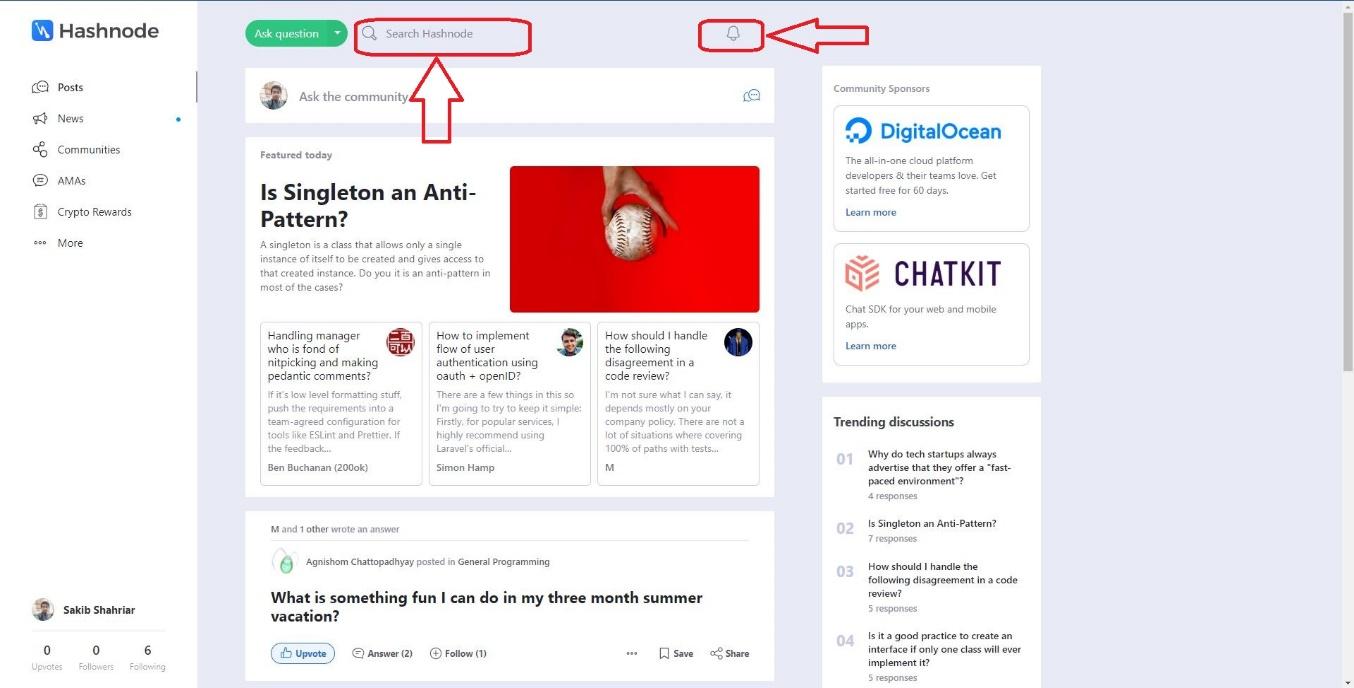
**2.** **Purpose of the project:** Developers are often face many problems which they cannot solve by themselves. They need some help from someone who can help them. Our main objective is to helping developers by connecting them in a network or community where they can know each other, share their problems on specific topic, help others by answering their questions.

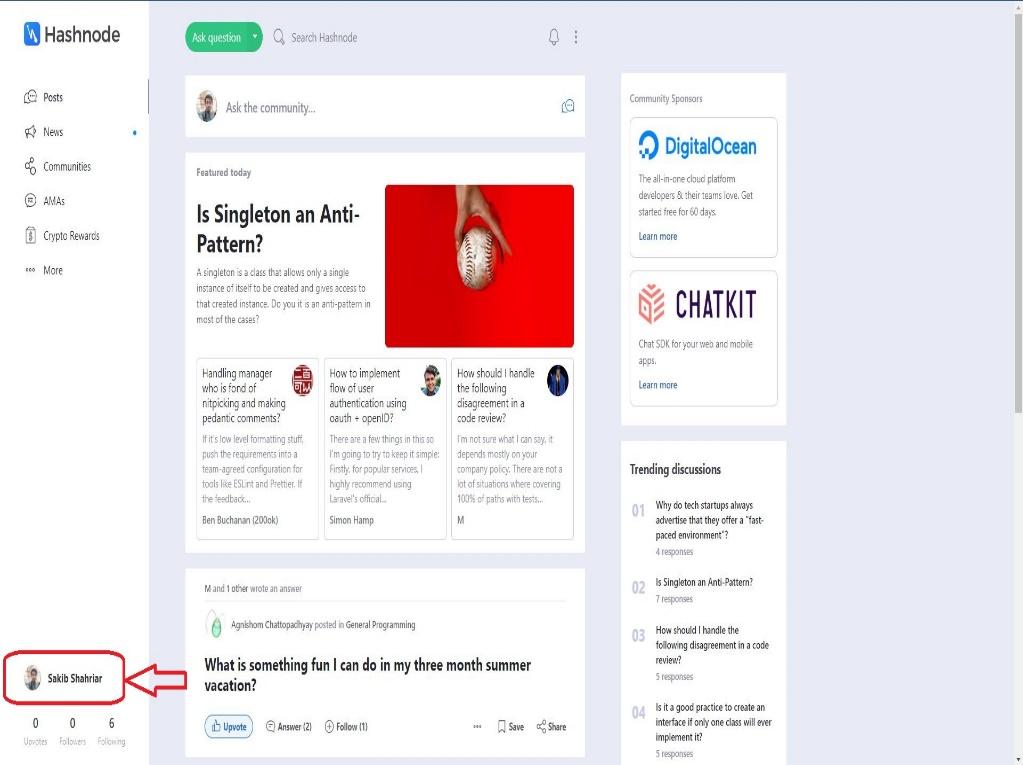
Information Gathering

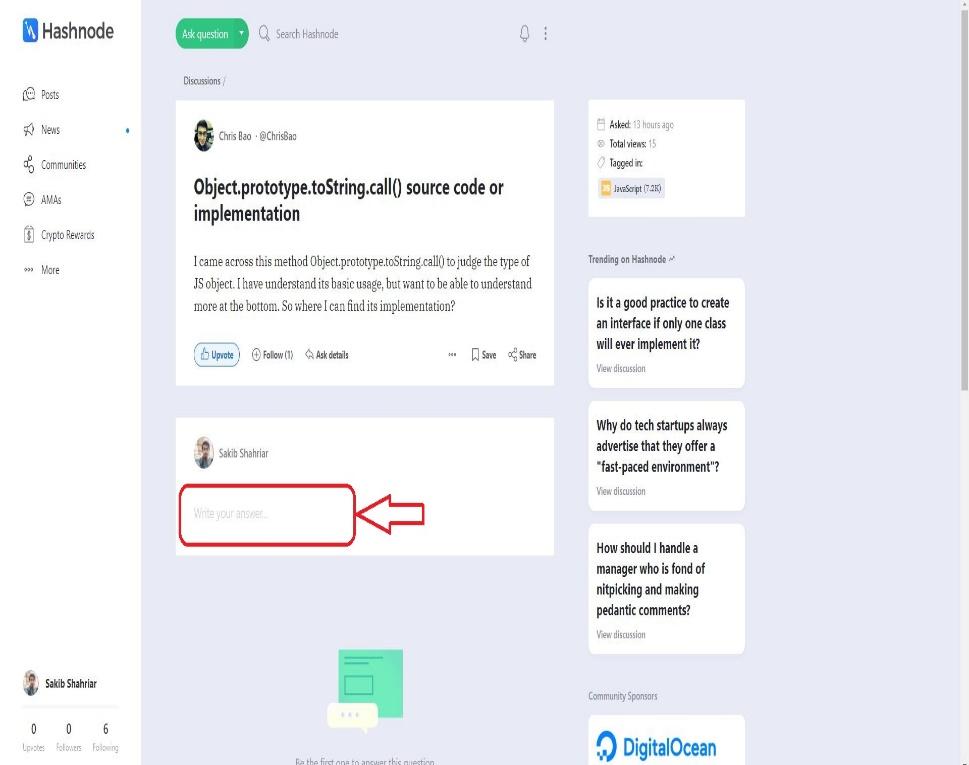
**1. Benchmark Products:**

HASHNODE

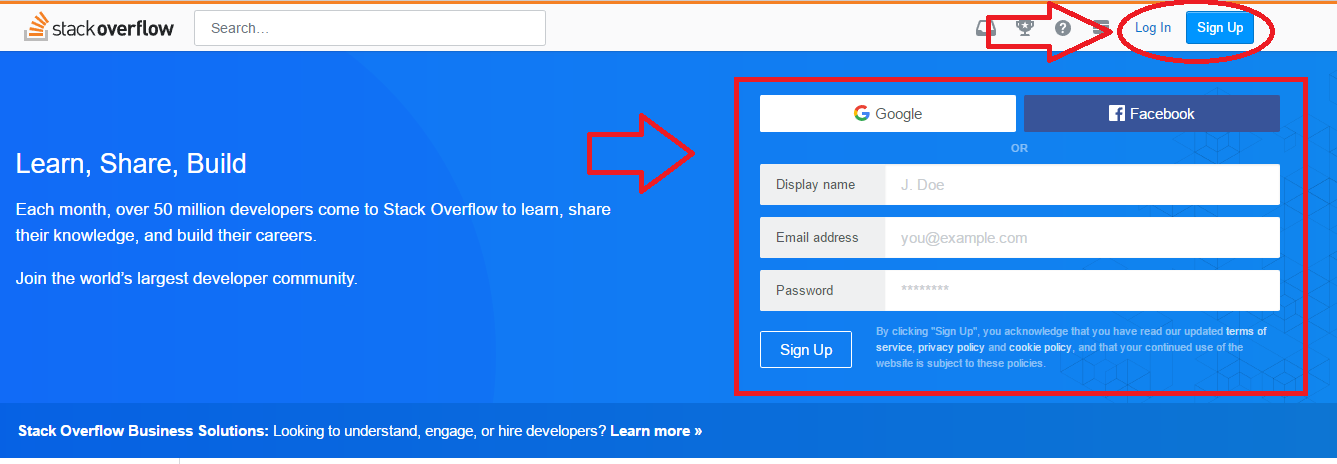


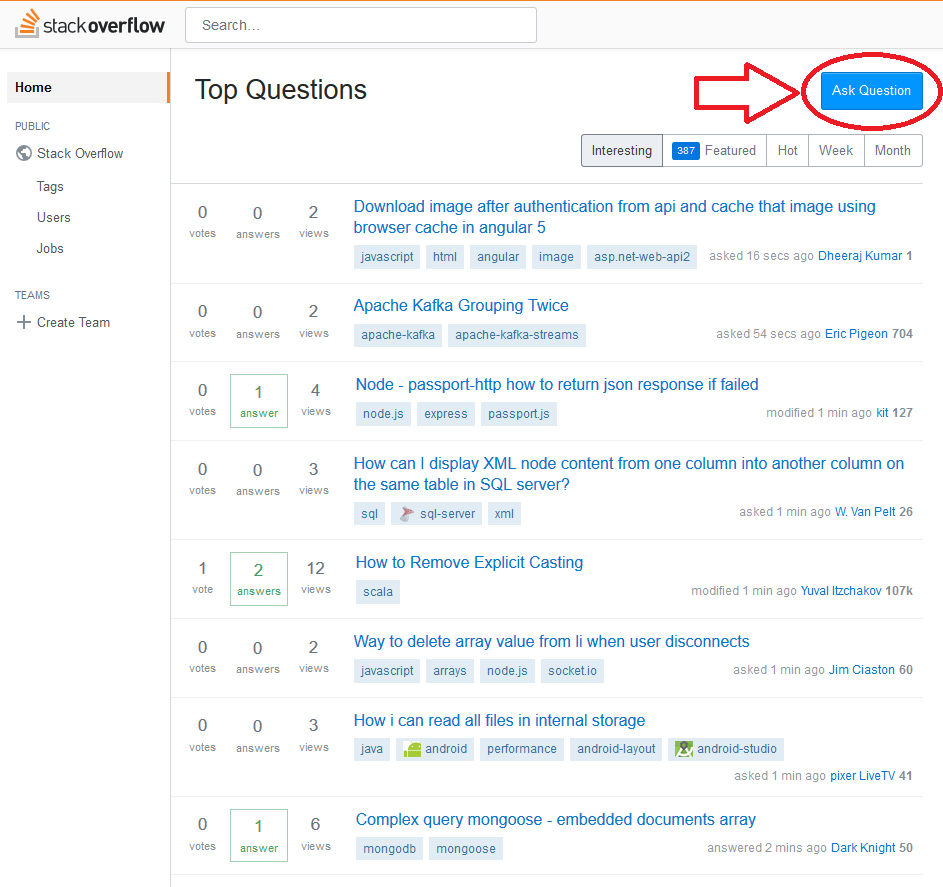


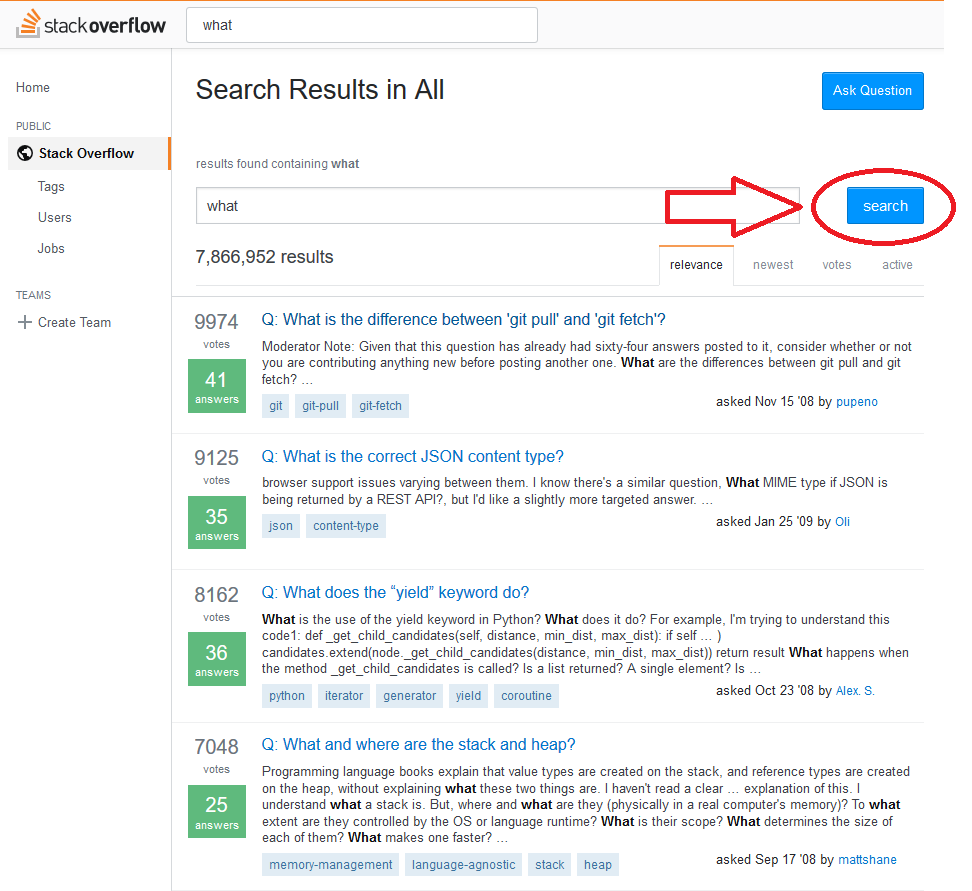


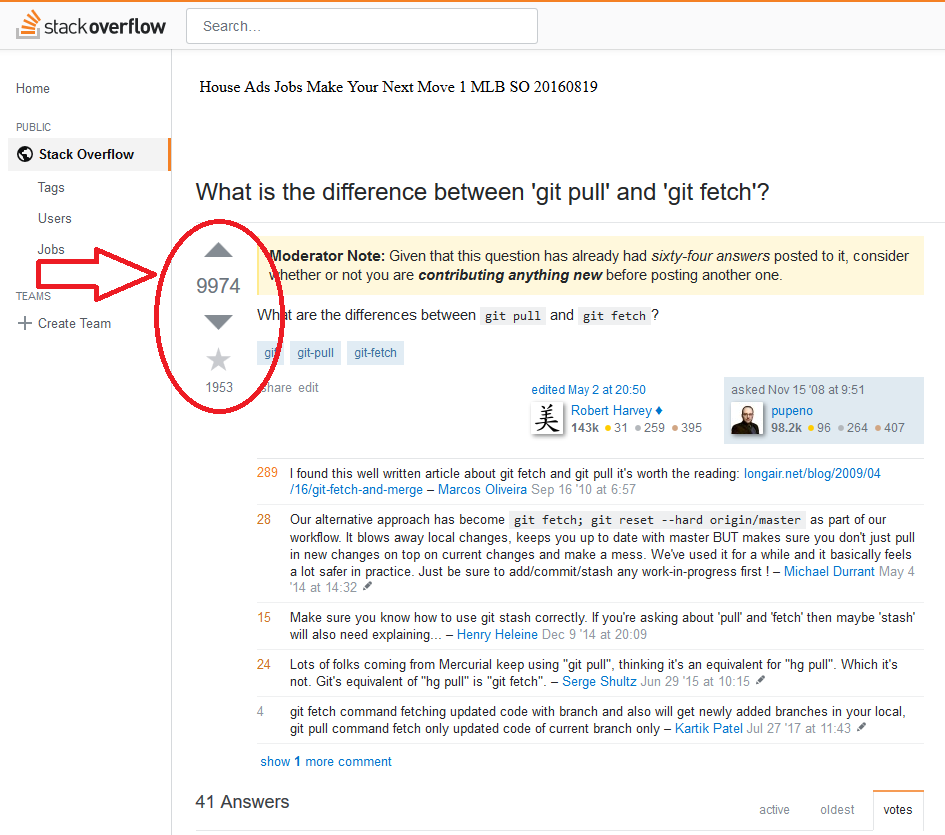


STACK OVERFLOW

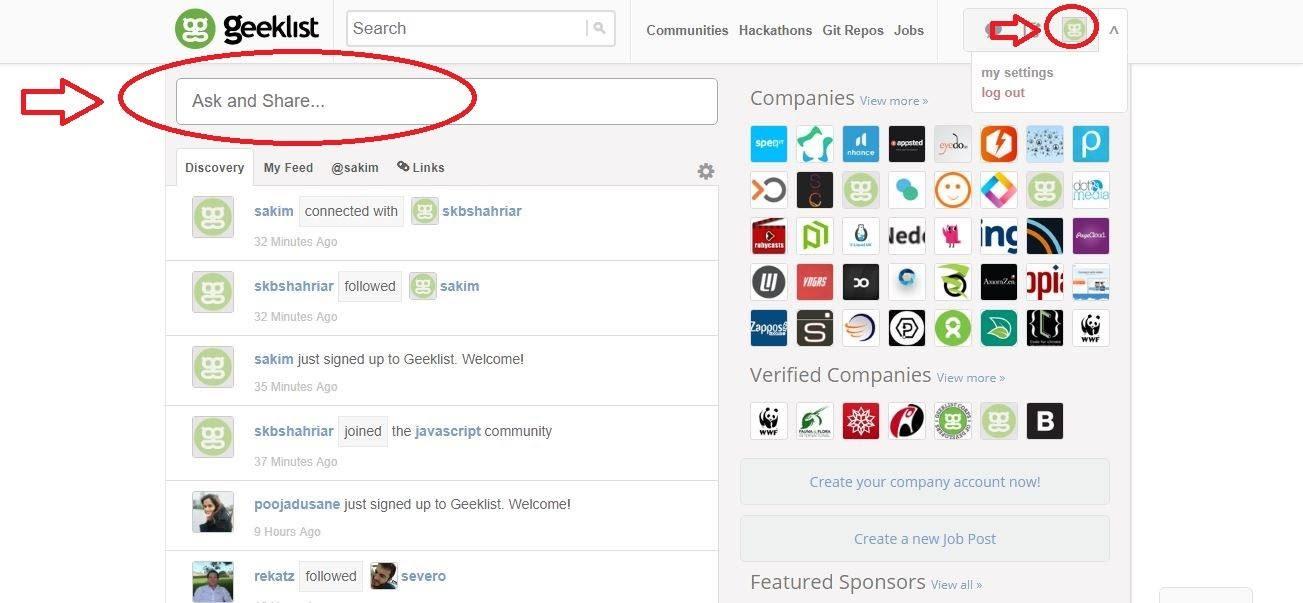


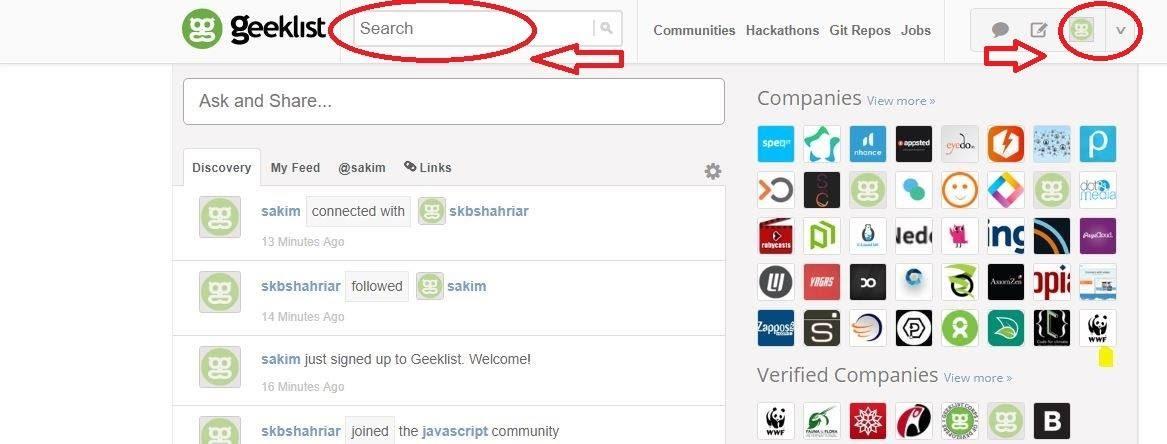


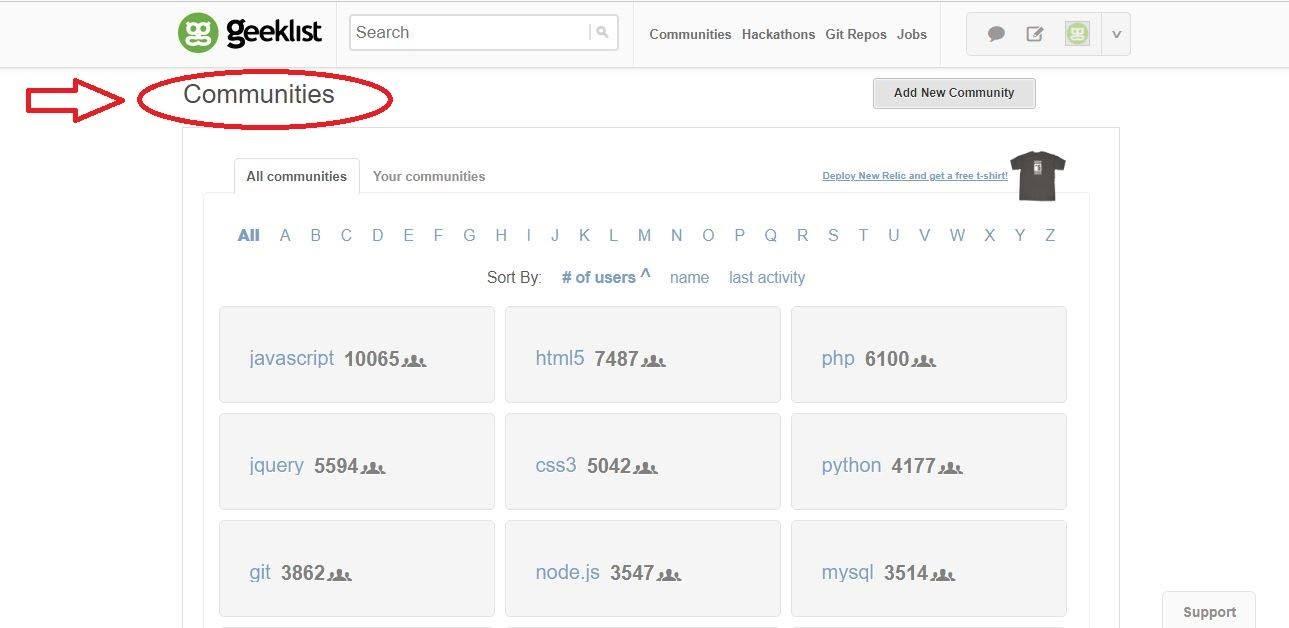


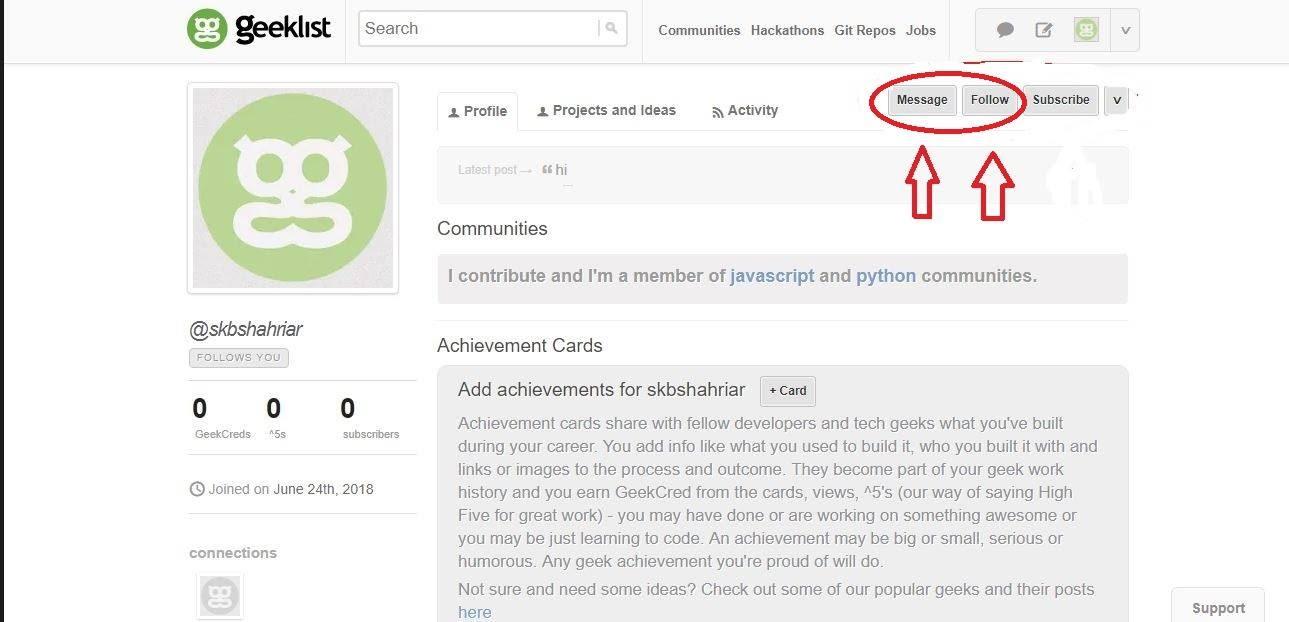


GEEKLIST

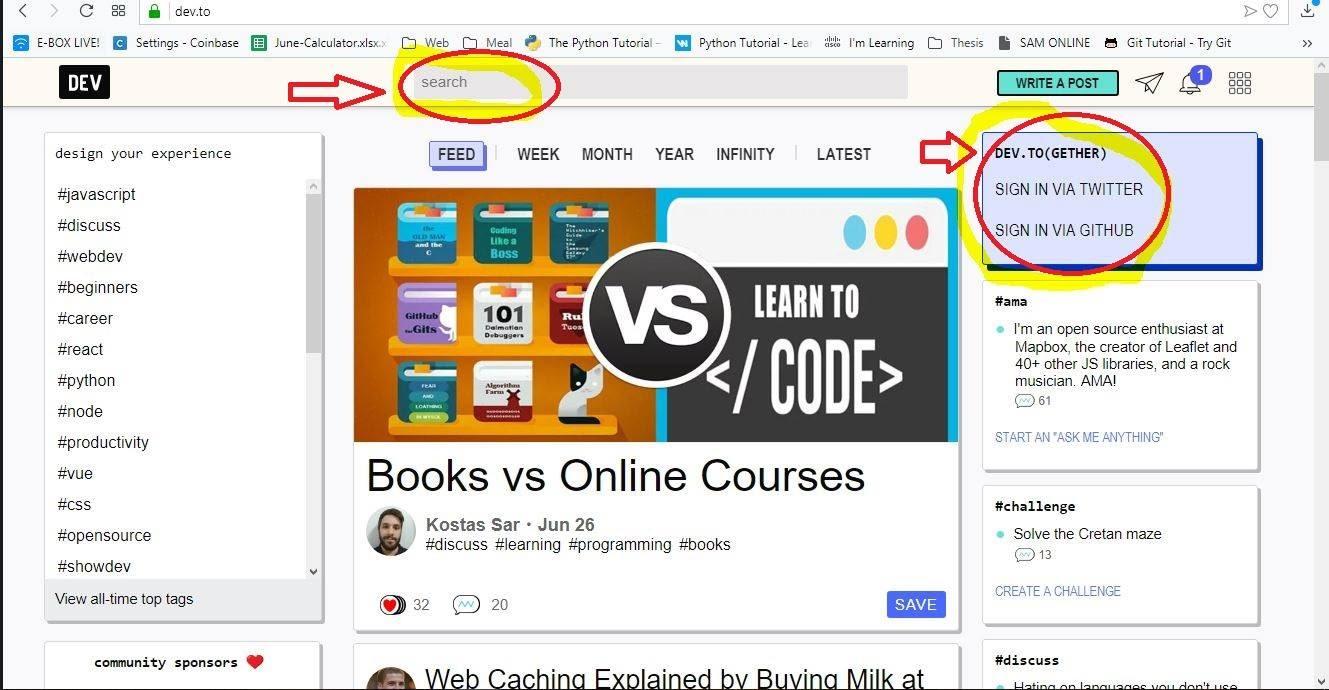


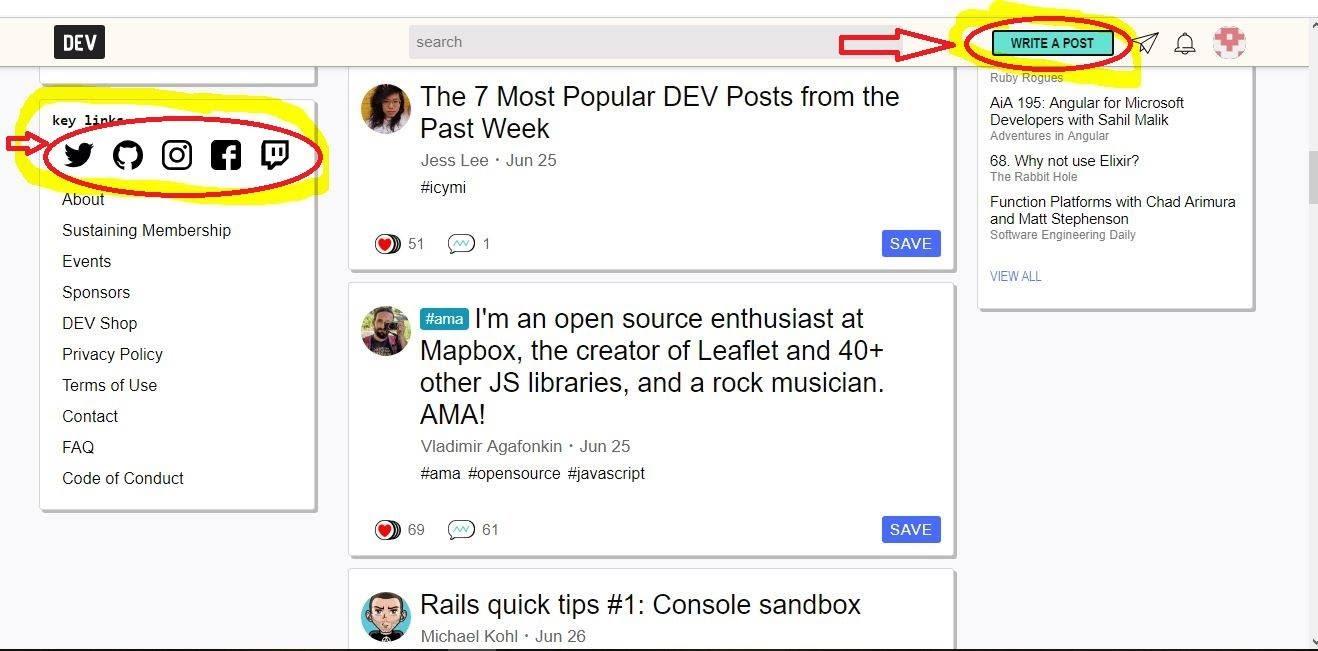


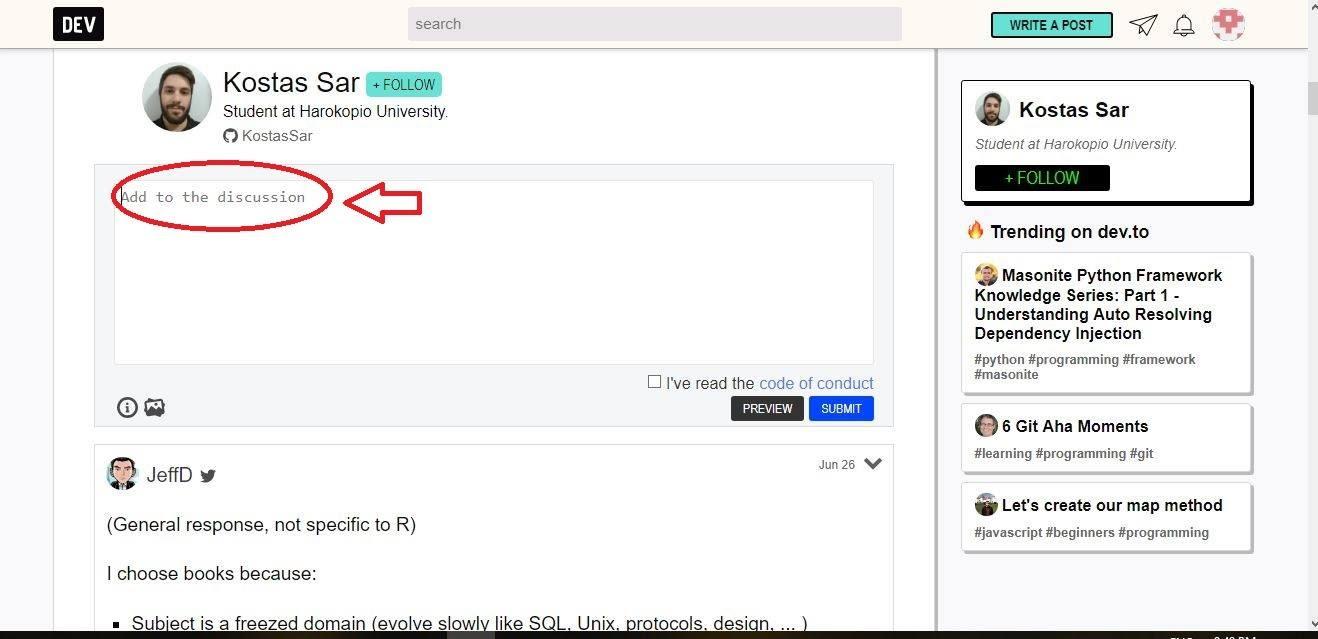


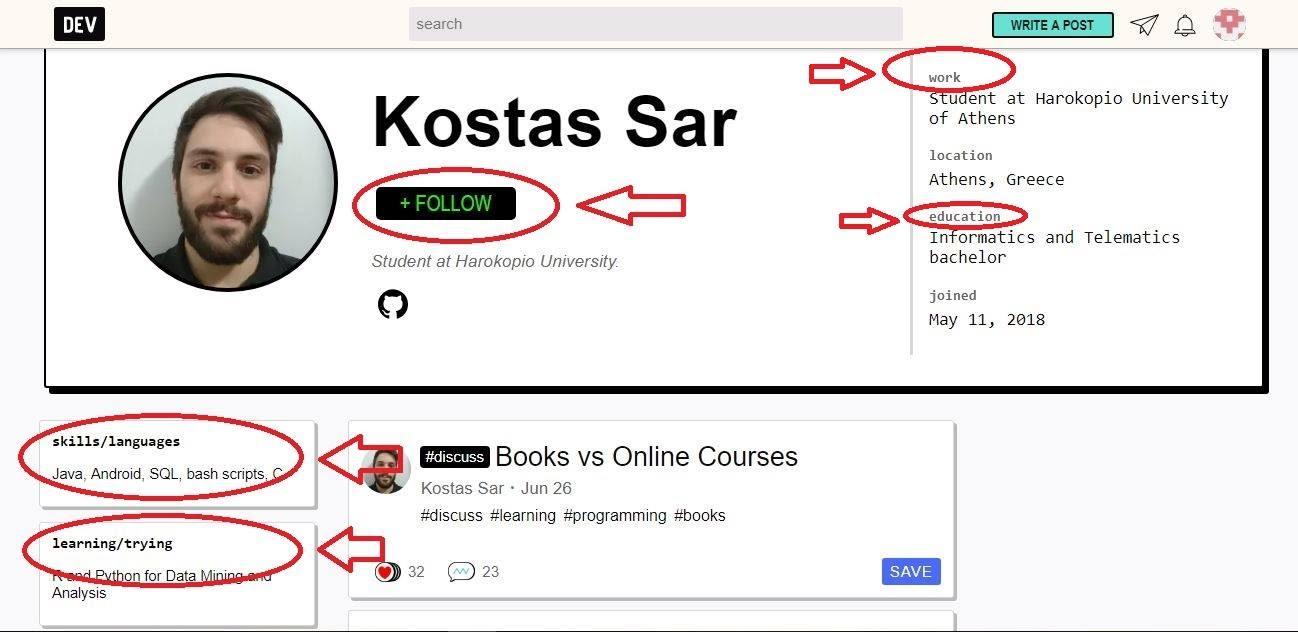


DEV.TO



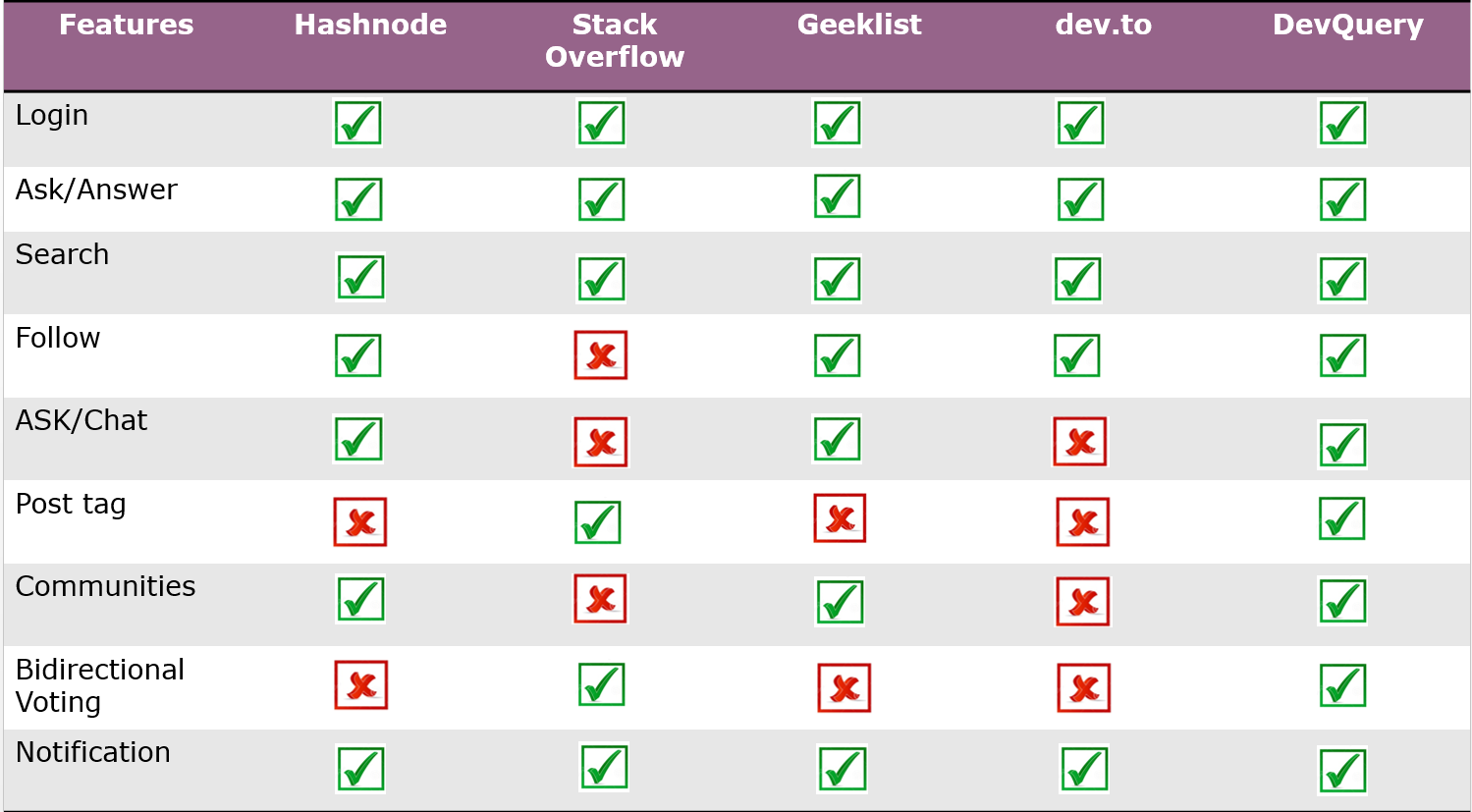






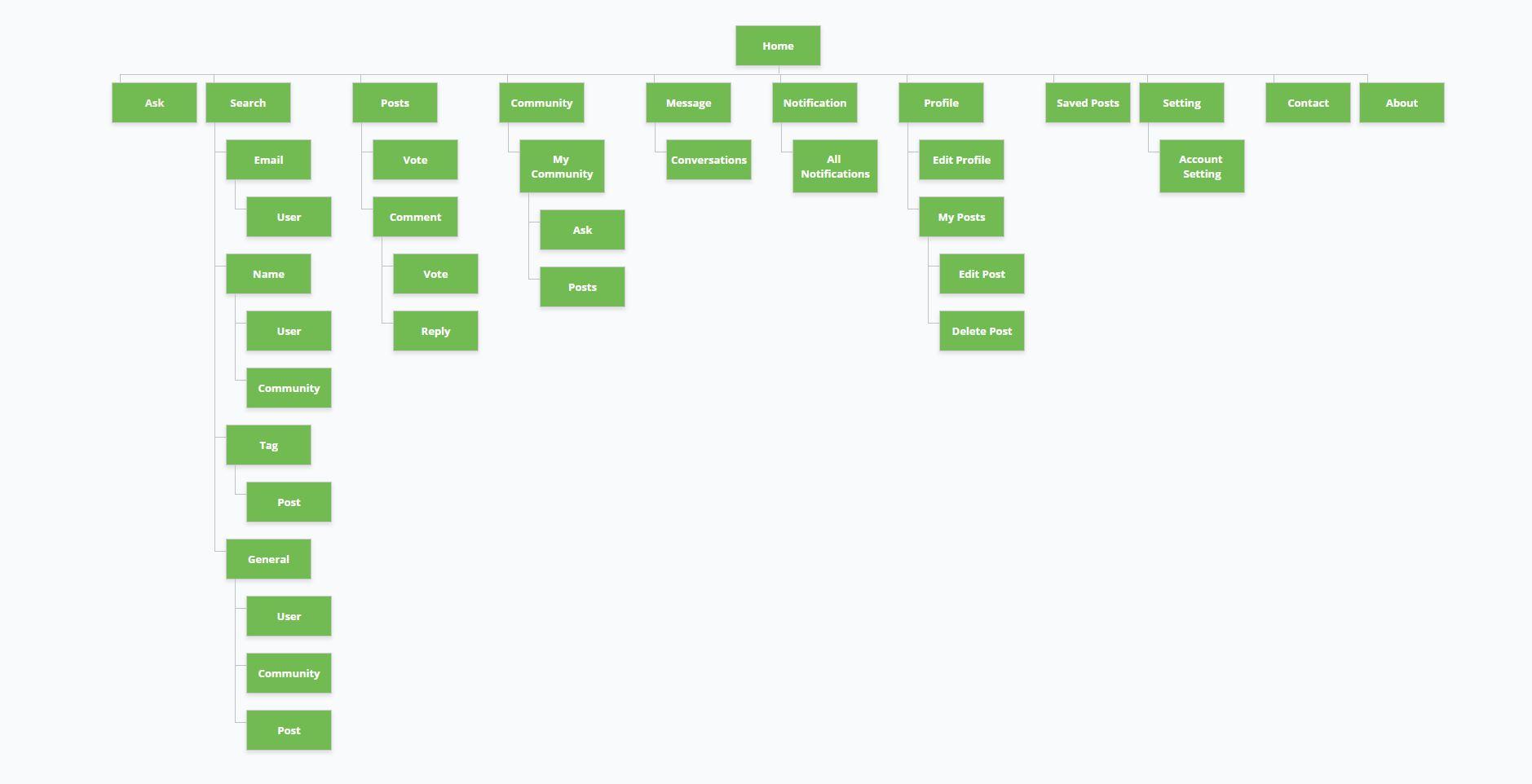
* LINKS OF BENCHMARK PRODUCTS:
* <https://hashnode.com/>
* <https://stackoverflow.com/>
* <https://geekli.st/>
* <https://dev.to/>

**2. Feature Comparison:**



**Feature List**

* **Feature List:**
* Log in
* Post
* Search
* Follow
* Chat
* Post tag
* Community
* Bidirectional voting
* Notification
* **Site Map:**

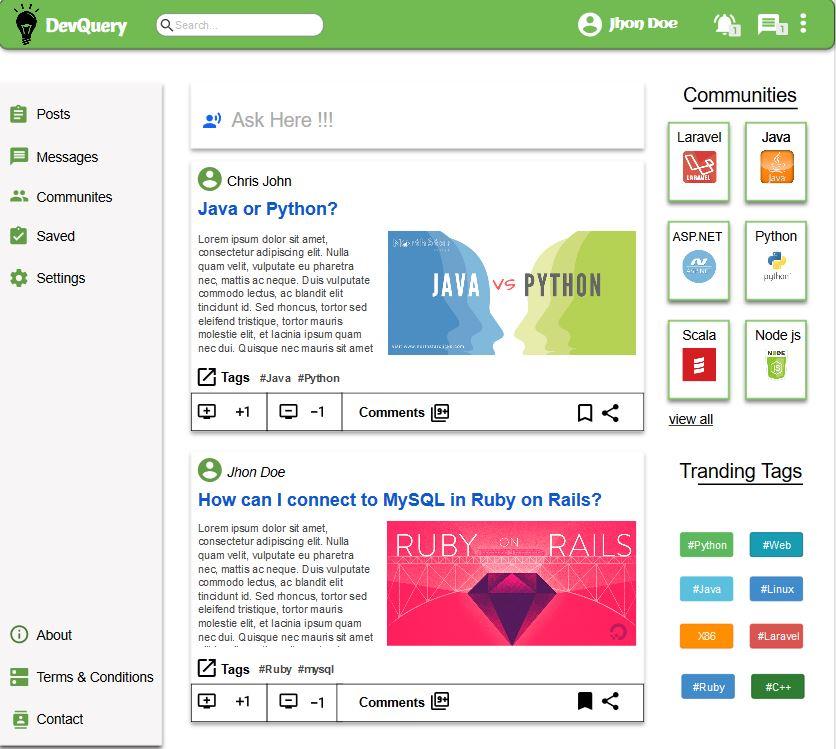


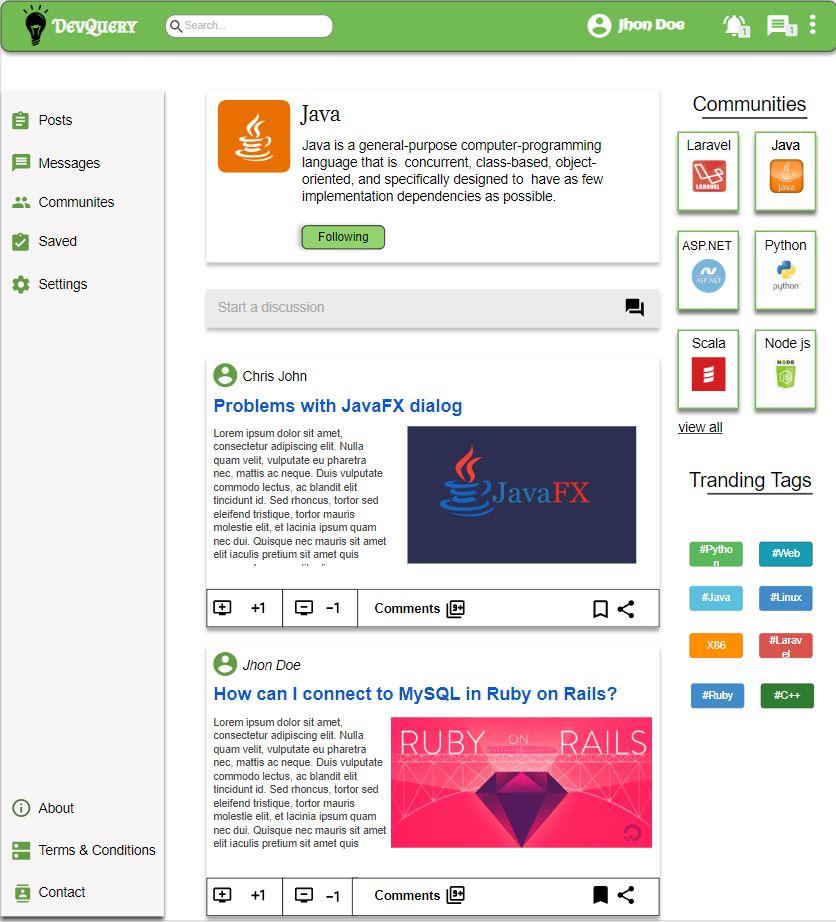
**Feasibility Study**

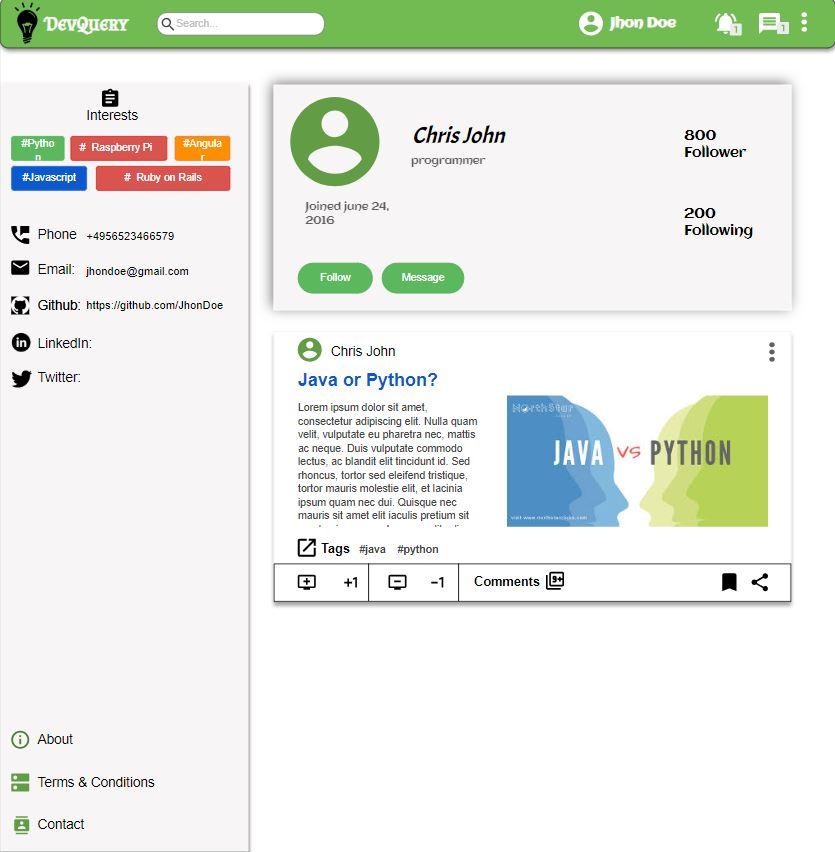
* **SWOT Analysis:**
* Strengths
* Weakness
* Opportunities
* Threats
* Strengths:

**UI Design**

* **Moqup:**

****

****

****

**System Design**

1. **Structural Design:**
2. Data Flow Diagram (DFD):

* Rules of Data Flow diagram:

1. Each process has name and number

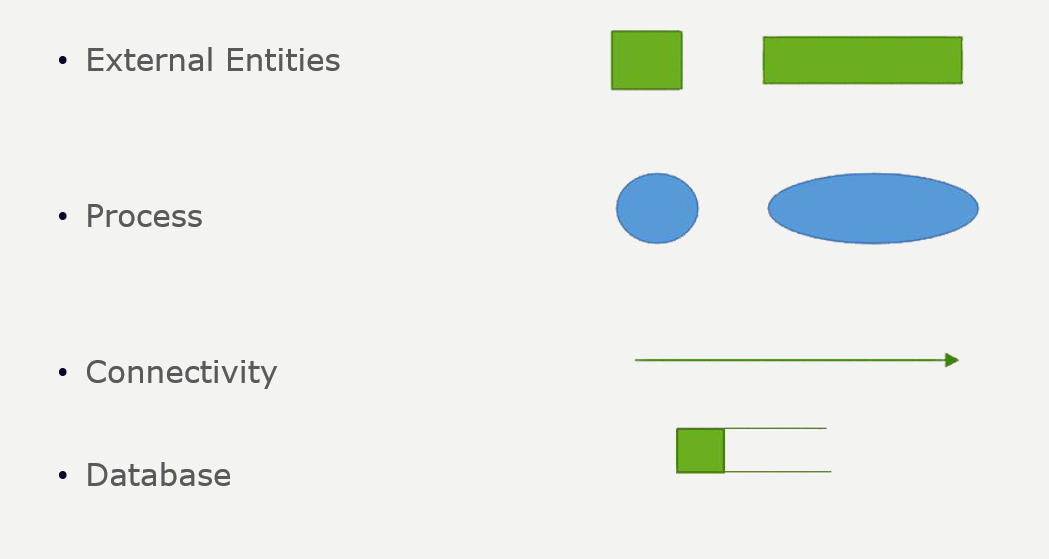
2. Database will be named capital letter

3. Database will be connected with only the process

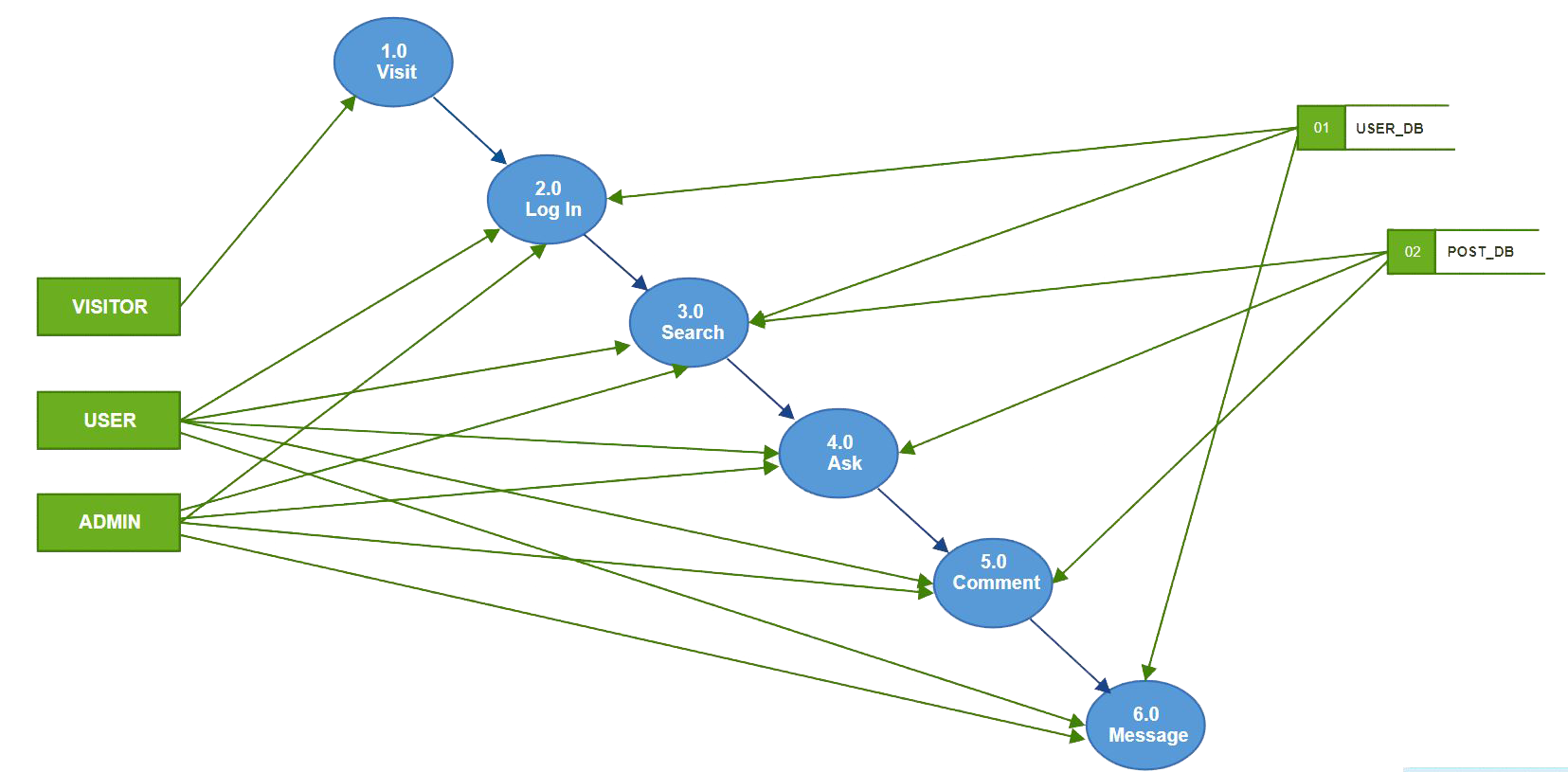
4. Start from top left and ended at right bottom.

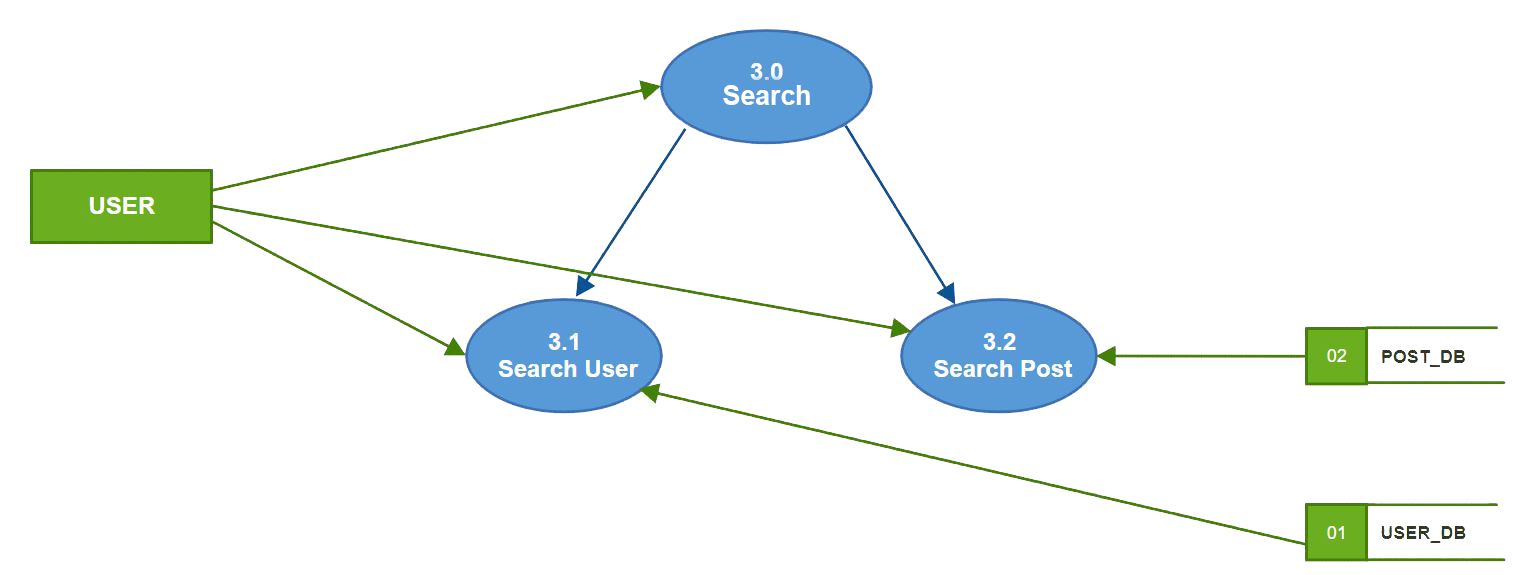
5. Database - database, external entity - database, external entity to external entity connection is not allowed

* Symbol for Data Flow Diagram:



* Diagram:

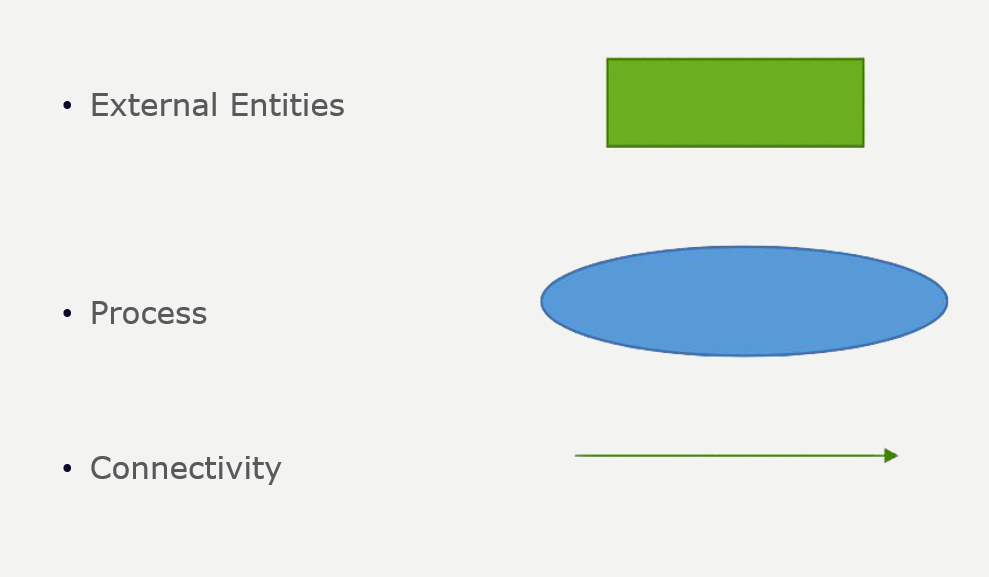




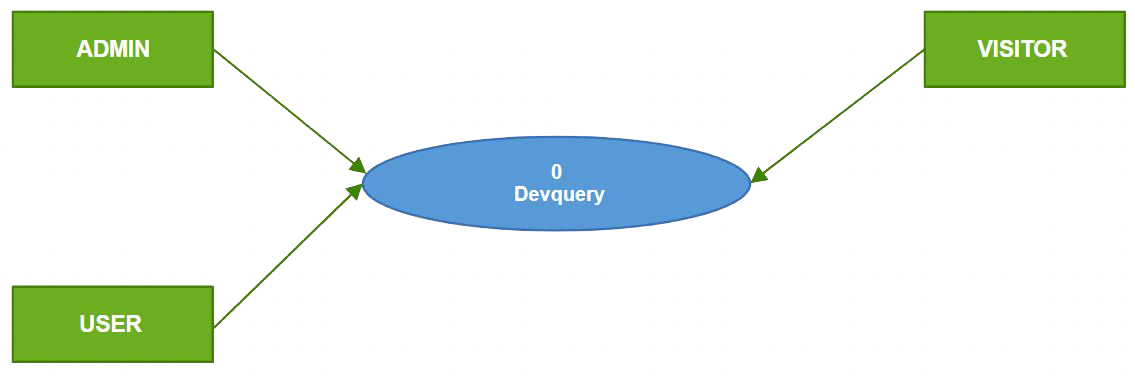
1. **Design by UML:**

**Context Diagram**

* Rules of Context Diagram:
  + 0 level Diagram
  + start form 0
  + External Entities must written in capital letters
* Symbols of Context Diagram:

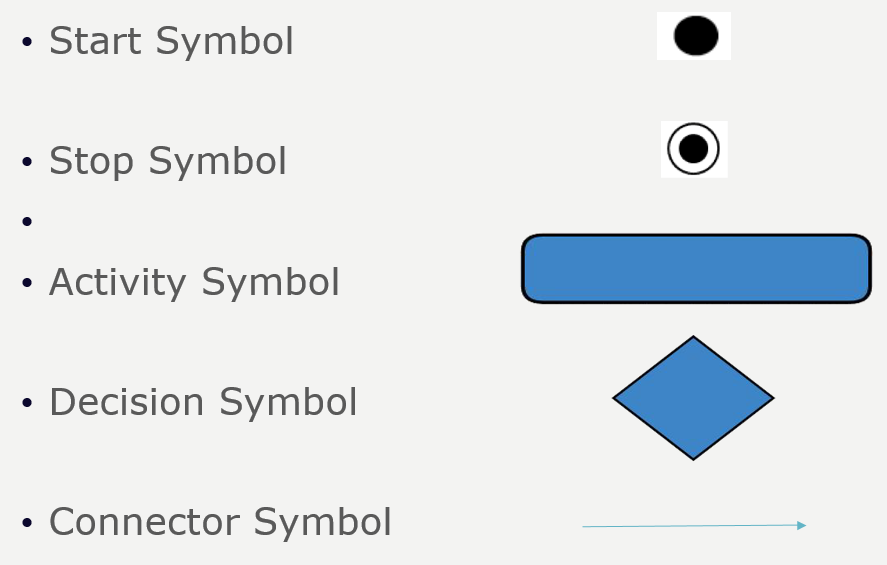


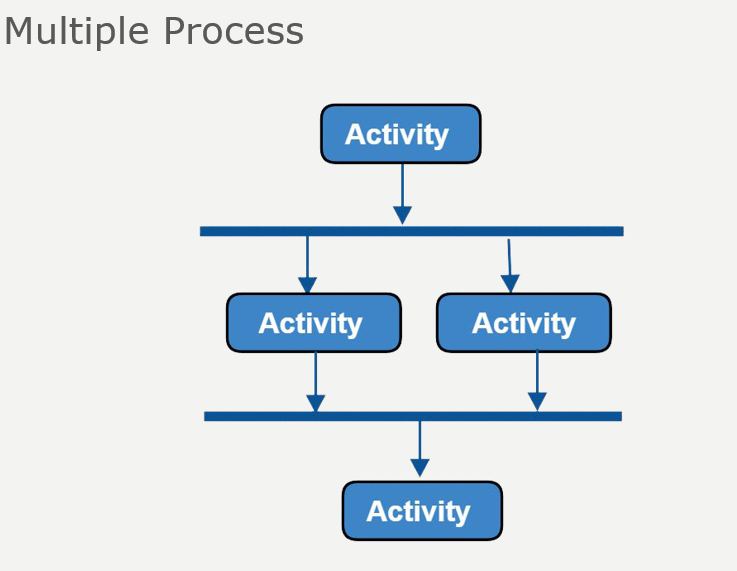
* Diagram:



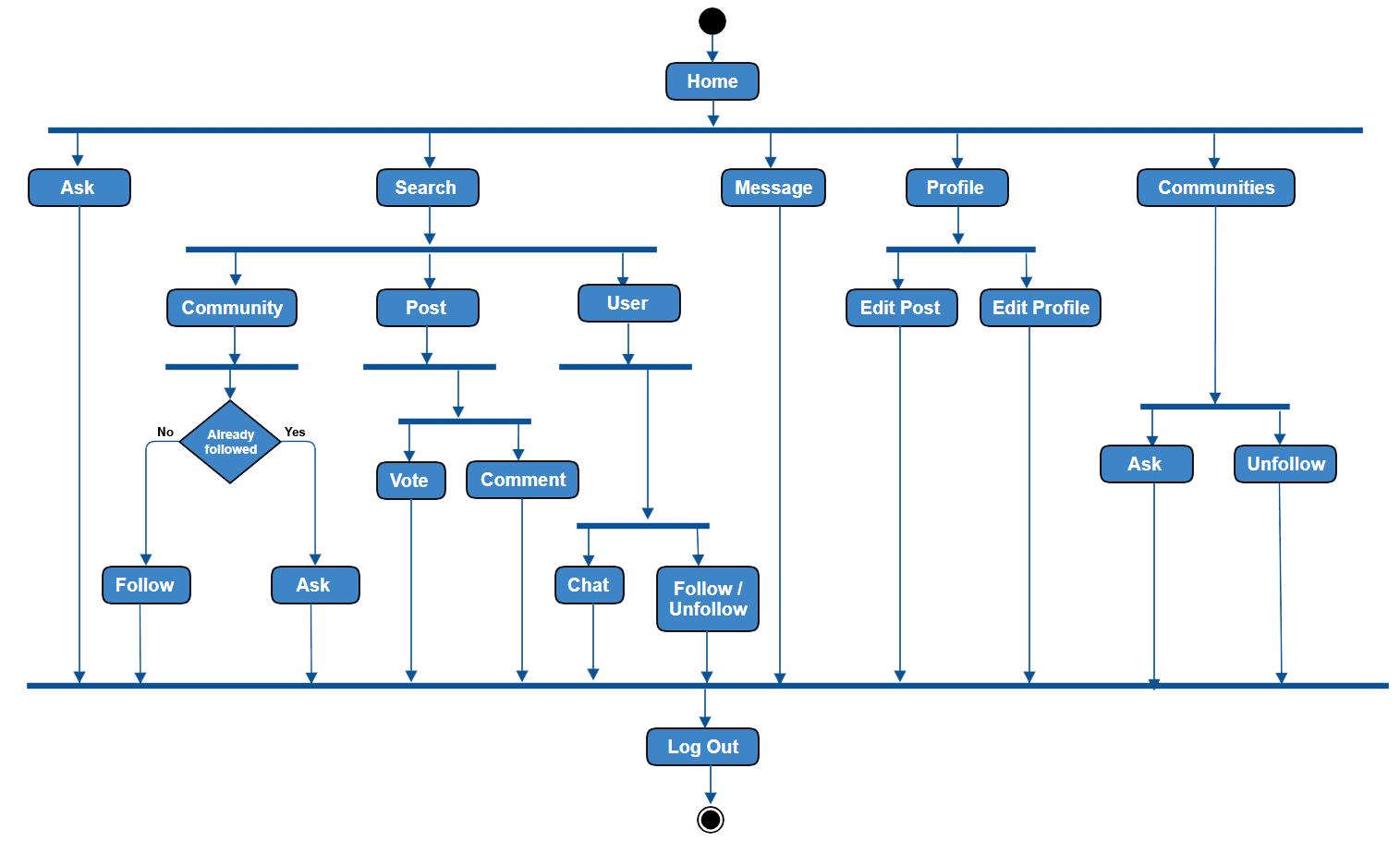
**Activity Diagram:**

* Rules of Activity Diagram:
  + Starts on top
  + Ends on bottom
  + Decision Box is used to check condition
  + Rectangular box indicates Process or Feature
  + Fork represent parallel processes
  + Each fork has one input and one output only
* Symbols of Activity Diagram:



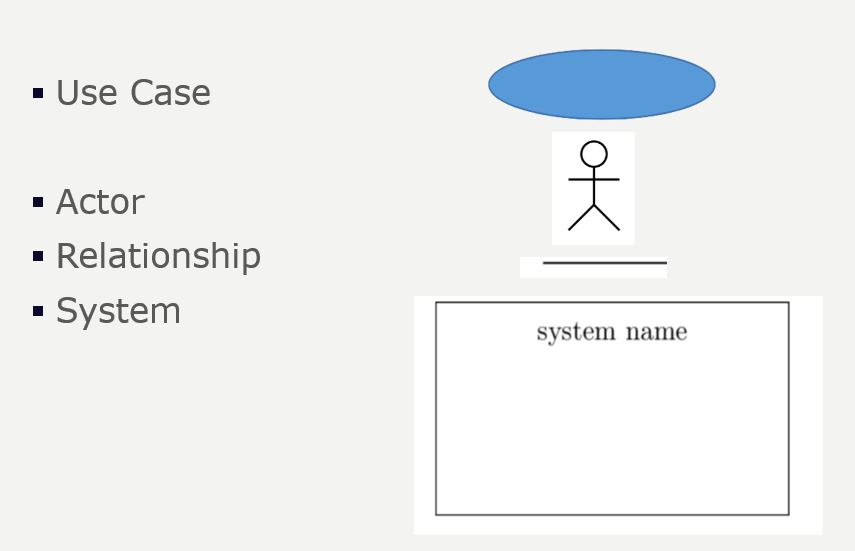


* Diagram:

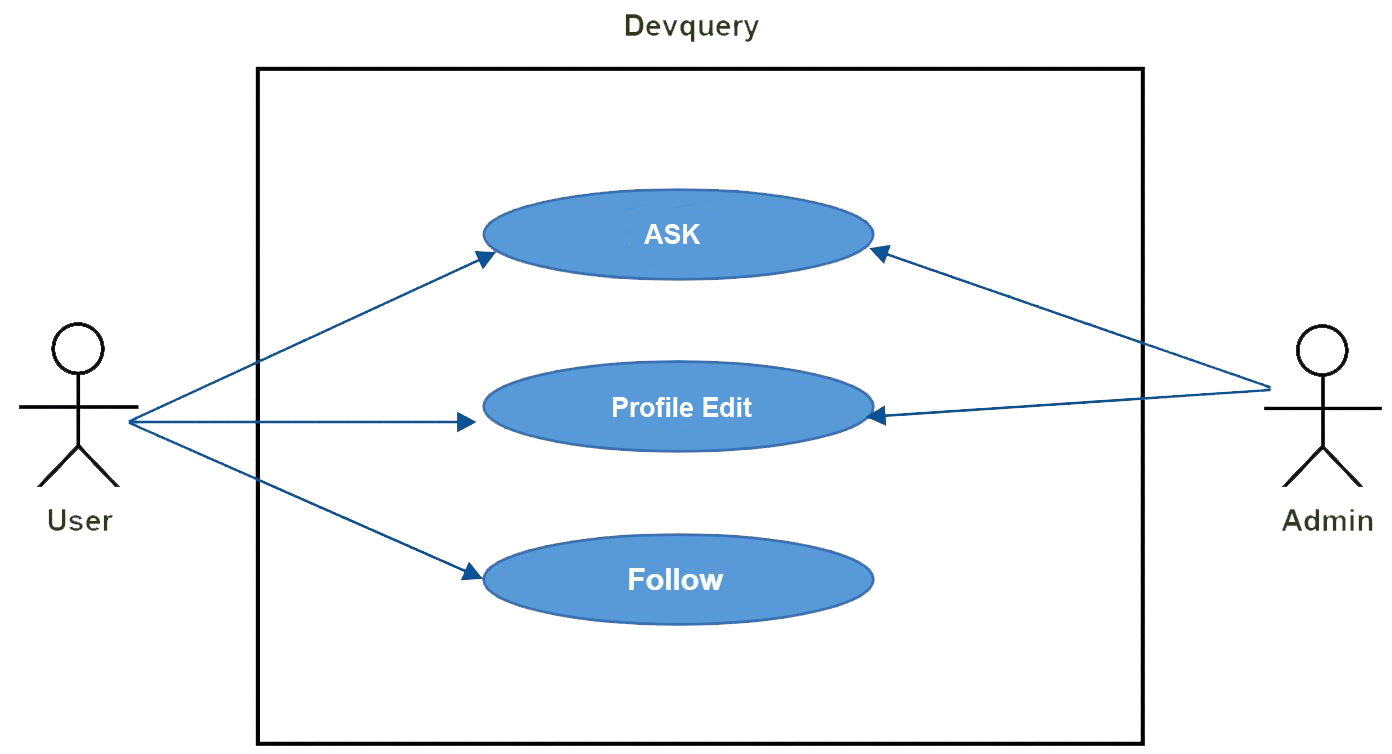


**USE CASE DIAGRAM**

* Rules of Use Case Diagram:



* Diagram:



* Use Case Descriptive Form:
  + - Use case name and number
    - Primary Actor
    - Stakeholders and Interests
    - Pre-conditions
    - Success Scenario
    - Alternate Scenario
    - Post-conditions

**Use case 1: Ask Question**

* **Primary actor**: User
* **Stakeholders and Interests**:
* **Use**r: Ask any question related with developer in Devquery platform
* **Admin**: Check the Question is Spam or not.
* **Success Scenario:**
  + 1. Now user can ask question in his timeline.
    2. User can reply to comment in their post.
    3. User can comment in their own post.
    4. User also can like or dislike the question.
    5. User can like or dislike any comment.
* **Alternate Scenario:**

**\*a: any time system can fail:**

* + 1. User restart the system, logs in and request recovery of prior state.

**1(a) Spam may come into comments:**

1. User can report for those spam message. Admin will handle this.

* **Pre-conditions:** 
  + 1. User must have a user id.
    2. User must have to log in with his or her user id.
* **Post-conditions:** 
  + 1. Post will be save in database with his or her user id and password.

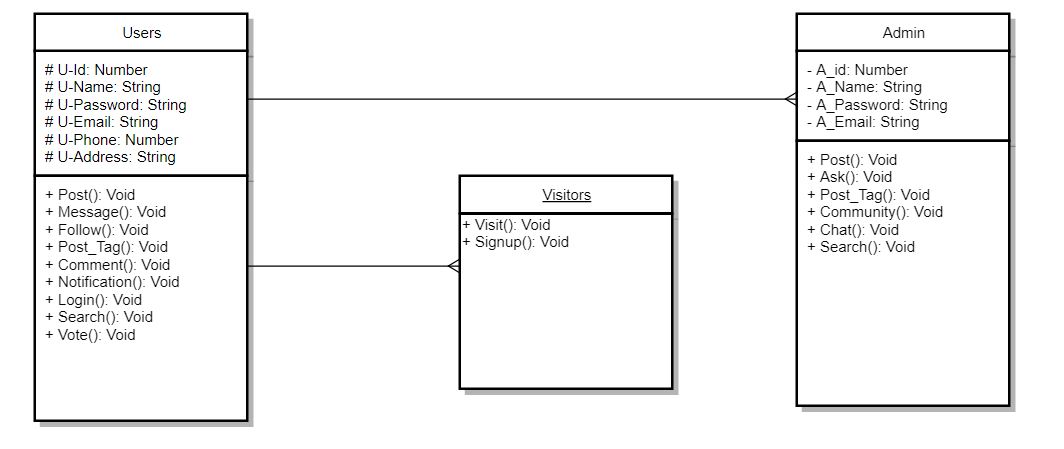
**Use case 2: Profile Edit**

* **Primary actor**: User
* **Stakeholders and Interests**:
  + - **Use**r:
      * + User can replace his or her old name, password and email by new name, password and Email.
        + User will able to change profile pic
* **Admin**: Update database and check the user is fraud or not by asking old password.

* **Success Scenario:** 
  + 1. User replace old name, password and email by new name, password and email.
    2. All process will be happen if the old password is right.
* **Alternate Scenario:**
  + - **\*a: Any time system can fail:** User restart the system, logs in and request recovery of prior state.
    - **2(a): Wrong password:**
      * **Solution:** Retype the correct password
* **Pre-conditions:**
  + 1. User have to visit Devquery’s website
    2. Must be log in with email and password
    3. Have to go to his or her profile
    4. Click on edit profile
* **Post-conditions:** 
  + 1. User new Name, Email, Password save in database.
    2. Now user will be identify by new name, password and email.

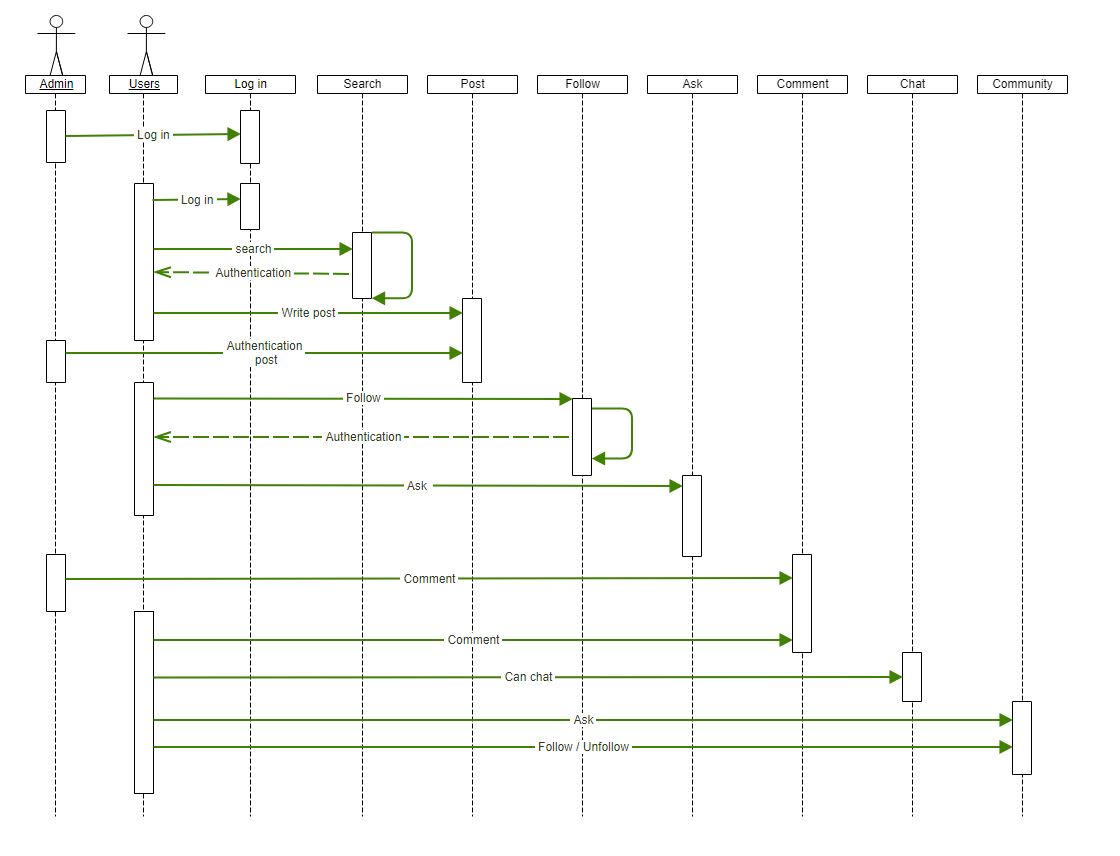
**Class Diagram**

* The class diagram simply describes the attributes and the operations of a class.
* Class name has to be noun. Such as: user, seller, buyer, customer, product, admin etc.
* There could be a relationship in inheritance.
* Relationship between two classes
* Events such as Palenque’s description
* Tangible things such as duster which has functions and attributes such as length, width and height.
* Attributes and methods can be,
  + - Private –
    - Protected #
    - Public +
* Diagram:



**Sequence Diagram**

* Also known as Interaction Diagram.
* The diagram is used to describe some types of interactions among the different elements in the model.
* The diagram shows,
  + - Dynamic behavior of a system.
    - Message flow in the system.
    - Structural organization of the objects.
    - Interaction among objects.
* Diagram:



**State Diagram**

* A state diagram simply describes the state of a system or machine.
* State diagram describes the flow of control from one state to another state.
* Things to be identified,
  + Objects
  + States
  + Events
* Diagram:

