

# CODE SNIPPET RECOMMENDER

Group 13:

Karan Bhatt - 2022202003

Deepak Singh Shekhawat - 2022202008

Pritha Ghosh - 2022201067

Devesh Nandan - 2022201080





## OBJECTIVE

People who are not usually proficient in writing codes from scratch, tend to struggle for long stretches of time to write a piece of code, which is just a small part of a bigger assignment.

Therefore, in these situations, it would be very helpful if these small snippets of codes were available in a single platform for people to look them up. This project aims to do that.



# PROJECT OVERVIEW

- We have used the Python web development framework Flask for our project.
- We have a simple UI which has two options on screen, one is to contribute by uploading a code snippet, and the other one is search for code snippets serving a certain purpose.
- The user clicks on the button corresponding to their requirement and then they are redirected to another page accordingly.
- If the user intends to upload a particular code snippet, the user is redirected to the sign-up page where the user either signs up or registers if they are a first time user.



# PROJECT OVERVIEW

- After logging in, the user is redirected to a form.
- The form asks the user for the language in which the code snippet is written, the keywords describing the code snippet, and finally the snippet itself.
- If the user intends to upload a particular code snippet, the user is redirected to the sign-up page where the user either signs up or registers if they are a first time user.
- After logging in, the user is redirected to a form.



# PROJECT OVERVIEW

- The form asks the user for the language in which the code snippet is written, the keywords describing the code snippet, and finally the snippet itself.
- If the user wants to rate a particular snippet according to their experience and satisfaction, the user will have to register if they are a new user or login if they are already a registered user.
- To search for code snippets, the user need not register or log in. They can simply search in the search bar to view the desired snippets.



# TOOLS AND SYSTEM REQUIREMENTS

**Programming language:** Python, HTML, CSS, Javascript

**Framework:** Flask

**Database:** MongoDB



# PROJECT ARCHITECTURE

## User Authentication:

- Create Account
- Sign in
- Sign out

## Search:

- Keywords based searching
- Upvote/Downvote code snippets
- Copy code to clipboard

## Upload Code Snippet:

- Contribute code snippet by filling a form
- View contributed code snippets
- Delete uploaded code snippets

```
graph TD
    subgraph System
        Registration
        Login
        Upload_code[Upload code]
        Available_code[Available code snippets]
        Fill_form[Fill form]
        Search
        Uploaded_code[Uploaded code]
        keywords
        Delete_code[Delete code]
        Vote
        Use_code[Use code snippet]
    end

    Upload_er((Upload er))
    User((User))

    Upload_er --> Registration
    Upload_er --> Login
    Upload_er --> Upload_code
    Upload_er --> Fill_form
    Upload_er --> Uploaded_code
    Upload_er --> Delete_code

    User --> Login
    User --> Search
    User --> keywords
    User --> Use_code

    Registration -.->|<<Include>>| Login
    Login -.->|<<Extend>>| Upload_code
    Login -.->|<<Extend>>| Vote
    Upload_code -.->|<<Include>>| Fill_form
    Fill_form -.->|<<Include>>| Uploaded_code
    Available_code -.->|<<Include>>| Search
    Search -.->|<<Include>>| keywords
    Search -.->|<<Include>>| Use_code
    Search -.->|<<Extend>>| Copy_clipboard[Copy code to clipboard]
    Copy_clipboard -.->|<<Extend>>| Uploaded_code
    keywords -.->|<<Extend>>| Delete_code
    Use_code -.->|<<Extend>>| Search
    Vote -.->|<<Include>>| Search
```





# FUNCTIONALITIES AND LOGIC

- At first, user lands on **search page** from where he/she can directly search for code snippets(without signing in) by providing keywords.
- After the user searches for code, he/she is **redirected on the search results page**.
- On the search result page, all the **matching code snippets are shown** and user is given the option to copy the code to clipboard.
- Search results shows details like user who uploaded the code snippet, language, description, vote counts and the code itself



# FUNCTIONALITIES AND LOGIC

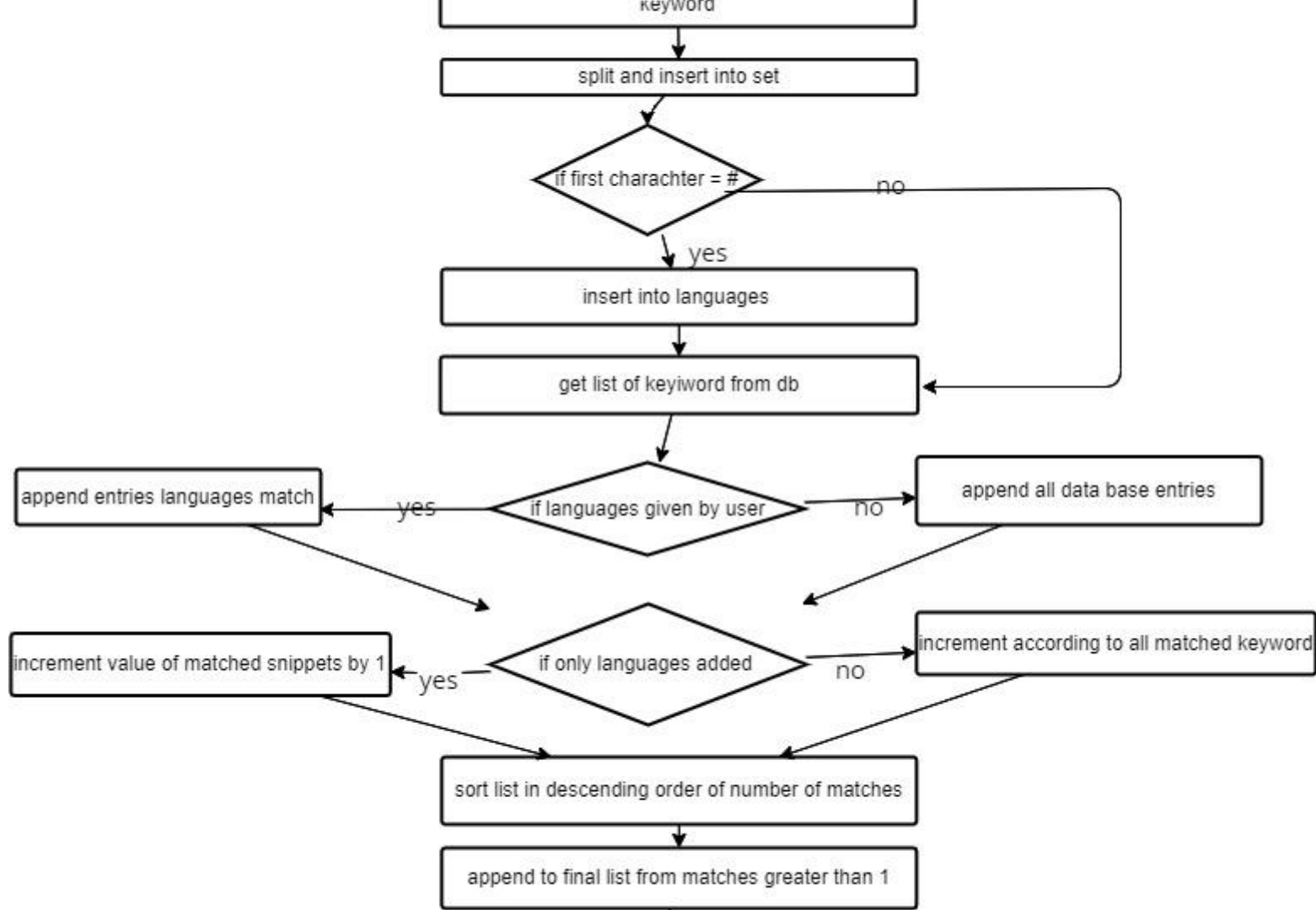
- When the user searches by entering a particular type of code snippet, the search takes into consideration the matching of the keywords first, followed by the number of upvotes received by a particular code snippet.
- The uploader can delete their own snippet by clicking on the delete button
- Also there are options to Upvote or Downvote the code snippets, but user needs to be logged in for that functionality.
- User can create account and log in to upload code snippets.



# FUNCTIONALITIES AND LOGIC

- After logging in, user is provided with the form to upload code snippets.
- All the code snippets uploaded by a user is shown on the dashboard page itself.
- User can delete the uploaded code snippet from the dashboard.
- After logging in, there is search bar on the dashboard page from where user can search for code snippets and also upvote/downvote it.

Next slide shows the flowchart for the keyword based search:





# DATABASE DESIGN

Mongodb has been used for the database requirement of our project.

## User:

- `_id`
- `name`
- `email`
- `password`

## Code Snippet:

- `_id`
- `user_email`
- `language`
- `keywords`
- `description`
- `code snippet`
- `votes`

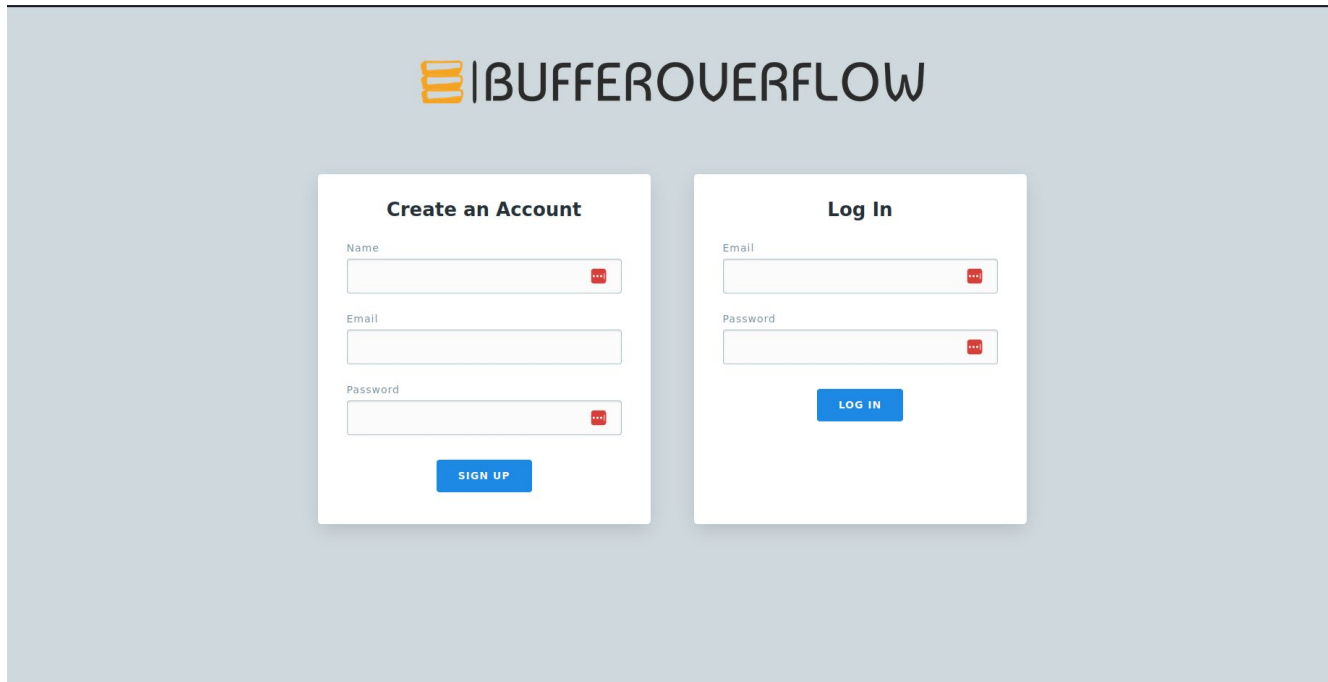


# RESULTS AND SCREENSHOTS





# RESULTS AND SCREENSHOTS



The screenshot displays the Buffer Overflow web application interface. At the top center is the logo, which consists of a yellow icon of three stacked horizontal bars followed by the text "BUFFEROVERFLOW" in a bold, black, sans-serif font. Below the logo are two white rectangular forms with rounded corners, set against a light gray background. The left form is titled "Create an Account" and contains three input fields: "Name", "Email", and "Password". Each input field has a small red "x" icon on its right side. A blue "SIGN UP" button is positioned at the bottom of this form. The right form is titled "Log In" and contains two input fields: "Email" and "Password", each also featuring a red "x" icon. A blue "LOG IN" button is located at the bottom of the "Log In" form.

**BUFFEROVERFLOW**

**Create an Account**

Name

Email

Password

**SIGN UP**

**Log In**


Email

Password

**LOG IN**



# RESULTS AND SCREENSHOTS

 BUFFEROVERFLOW

Hi Karan, [SIGN OUT](#)

Search snippet here...

SEARCH SNIPPET

**Language:** python  
**Description:** while loop range 10 python  
**Votes:** 3  
**Code Snippet:**

```
i = 0
while(i < 10):
    pass
    i += 1
```

DELETE

**Language:** c++  
**Description:** for loop range 100 c++  
**Votes:** 0  
**Code Snippet:**

```
for(int i = 0; i < 100; i++)
{
    // do something
}
```

DELETE

**Language:** c++  
**Description:** while loop range 50 c++  
**Votes:** 6

**Language:** python  
**Description:** for loop range 10 python  
**Votes:** -1

### Submit Code Snippet

Language

Keywords


Description

Code Snippet





# RESULTS AND SCREENSHOTS


 BUFFEROVERFLOW

loop

SEARCH SNIPPET

SIGN OUT

**User:** a@b.com  
**Language:** c++  
**Description:** while loop range 50 c++  
**Votes:** 6


**Code Snippet:** 

```
int i = 0;
while(i < 50)
{
    // do something
    i++;
}
```

UPVOTE

DOWNVOTE

**User:** a@b.com  
**Language:** python  
**Description:** while loop range 10 python  
**Votes:** 3


**Code Snippet:** 

```
i = 0
while(i < 10):
    pass
    i += 1
```

UPVOTE

DOWNVOTE

**User:** a@b.com  
**Language:** c++  
**Description:** for loop range 100 c++  
**Votes:** 0


**Code Snippet:** 

```
for(int i = 0; i < 100; i++)
{
    // do something
}
```

UPVOTE

DOWNVOTE

**User:** a@b.com  
**Language:** python  
**Description:** for loop range 10 python  
**Votes:** -1

**Code Snippet:** 



**Thank You**