Google Summer of Code

Social Smart Street

GSoC 2021
“Improving user experience of frontend and Enabling user personalization”

Project Proposal
By
Harsh Mishra

Table of Contents

About Me

Student Information

Name:	Harsh Mishra
Email:	harshm17172612@gmail.com, harsh.19je0353@ece.iitism.ac.in
TimeZone:	New Delhi, India (+5:30 GMT)
Github:	MiHarsh
LinkedIn:	harsh-mishra-5384b11a0
Portfolio:	https://miharsh.github.io/
Resume:	Link
Address:	Gorakhpur, Uttar Pradesh , India Pin- 273209

University Information

University:	Indian Institute of Technology (ISM) Dhanbad
Major:	Electronics and Communication Engineering
Current year:	2nd
Expected Graduation date:	August 2023
Degree:	B.Tech

Answers of listed questions

Que. For which projects have you worked? (For each project: write a few sentences describing what the project does; mention the programming languages used; describe your contributions; include a link to the project's code repository).

I have a keen interest in Machine Learning and Web Development. So my projects evolve around these fields.

1. **Pytorch Fanatics** : A Computer Vision Library for Classification Tasks
   Link : Github
   Tech Stacks : Python , Pytorch library
   Description : Pytorch , being an awesome library, requires a huge amount of code starting from dataloading ,training models to predicting results. I have created a library which includes a bunch of helpful functions starting from data loading to LR Finder, Early Stopping, Logger, Saver, etc.

2. **Medivision** : A Virtual diagnoser of diseases using medical scans
   Link : Github
   Tech Stacks : Python, Flask, HTML, CSS, Firebase
   Description : An AI solution for early diagnosis of seven diseases namely Alzheimer, Brain Tumor, Breast Cancer, Covid -19, Eye blindness, Pneumonia and Skin Cancer. I have used publicly available data from Kaggle. You can use this project by uploading images on the project's website. It is deployed via a website.

3. **Fake News Detection :** Data Analysis and Model Proposal
   Link : Github
   Tech Stacks : Python, Pytorch library
   Description : Did data analysis on the Fake News Detection KDD competition sponsored by Microsoft. Also proposed a model which helped our team secure 2nd position.

4. **Winter Of Code** : Implementation of Basic Machine Learning Algorithms
   Link : Github
   Tech Stacks : Python
   Description : Implemented basic machine learning algorithms such as linear regression, logistic regression, KNN , Neural Networks etc .

Que. Which relevant courses have you attended at the University? How was your performance?

I am currently a second-year undergraduate student at the Indian Institute of Technology (ISM) Dhanbad, India. I am pursuing a 4-year B.Tech course in Electronics and Communication Engineering. My current CGPA is 9.06. Additionally, I have attended an extra course on Data Structures and Algorithms and Computer Programming, in which I got an A grade.

Que. Why are you the best person to execute this proposal? (Demonstrate that you satisfy the requirements for the chosen idea.)

- I am keenly interested in open-source projects and have been passionately working on them. I am a quick learner and can quickly adapt to situations; moreover, I have good experience working as a team.
- I like to engage with real-world software engineering out of my coursework.
- Web technologies are fascinating to me - I think to implement my idea, I'll have to work on JS a lot.
- As an ML enthusiast, I'm also curious about the models deployed (e.g., fake news detection, clickbait detection, etc.) and would love to contribute to that.

Que. How much hours will you work per week on the project? Other commitments(eg exams, projects, jobs , thesis etc . How will you prevent them from affecting your GSoC performance ?)

I expect to work full time on the project *i.e.* 40 or more hours a week. I have no other commitments in the summer.

Background work and Programming Skills

I am a second-year student of IIT(ISM) Dhanbad, India. I'm pursuing a degree in Electronics and Communication. I work on Ubuntu 20.04 LTS. I use VS Code for development and vim for SSH sessions. I am proficient in C, Python, and Javascript.

Python easily lets me convert my ideas into code. I like it mainly because it is an interpreted language that gives you the freedom to do many things dynamically. Prototyping anything in Python is very easy and requires less man-work than any other programming language.

Since I started programming as a Machine Learning enthusiast, I used Pytorch and Flask, which made me familiar with Python.

I always loved to integrate ML and provide it publicly for everyone to use. I started using Javascript and its library jQuery to a large extent. It helps to manipulate websites as per our wish.

I like Javascript because of its asynchronous behavior.

I have recently found Chrome Extensions to be a great platform to create mini software projects for personal use - thus learning the web by doing it. The ability to do DOM manipulation of real websites when browsing is amusing (e.g., remove youtube comments).

I know how to use Git and Github.

If I am stuck, I go to Google and always come back with a solution.

Project Information

Organisation: AOSSIE

(Australian Umbrella Org for Open-Source Projects)

Project : Social Street Smart

Possible mentor: Utsav Shukla, Pranav Goyanka

Social Street Smart, being a chrome extension, provides a quick way for people to use it because it doesn't require any software or interpreter to run, so everything goes smoothly on the Chrome browser.

Social media has now become prone to fake news, spreading hate speech and disinformation. There must be a way to check and prevent this spread. Social Street Smart provides a better way to have a check. The current features are supported on major social media platforms such as Twitter, Facebook, etc.; it gives a review on clickbait, hate speech, and profanity words. This motivates us strongly to make it more user-friendly, hence enhancing a large community's daily lives.

Abstract

Social Street Smart extension provides some basic functionalities like labeling social media posts as clickbait, fake news, etc., on three platforms: Twitter, Facebook, and Reddit. Still, it needs new features to create a significant user impact. This project aims to develop a Chrome Extension to make the Internet a safer and more effective service for the users, hence needs lots of user-friendly features, which makes browsing social media a safe experience.
For this project, I would improve upon existing features, add a few new and enable user interaction for the first time.
During the GSoC period, I would add new features while ensuring a **solid, bug-free, non-glitchy smooth**, **flexible**, and **easy to use and maintain** code architecture.

Objective / Idea

I am focusing on improving user experience, providing a wide range of utilities, and documentation of the codebase for helping other open-source enthusiasts contribute easily.

I will also add few features,

- Improve the front-end of the settings page, make it more interactive.
- Options to Hide/Unhide,Report, Feedback.
- API for Disinformation in images.

In my view, the relevant skills required would be Python, javascript, Flask, and some knowledge of Machine Learning.

Implementation/Approach

Proposed Features :

1. Hide/Unhide content :

As of now, the chrome extension only flags posts as toxic or clickbaits, but it doesn't automatically hide them.

We should add a toggle on the settings page to enable auto-hiding of all toxic or clickbaits. There will be two switches, each for hiding toxic and clickbait posts.

How will it help ?
It will help users see relevant contents only.

How will I implement it ?
We'll store two keys using the chrome local storage API to represent these two switches' state - `auto_hide_clickbaits` and `auto_hide_toxic`.

Once the site is loaded and the clickbait/toxic check response has been obtained, we'll immediately check if the auto_hide_* keys are enabled and auto hide the posts instead of labelling them.

Some concerns in existing implementations

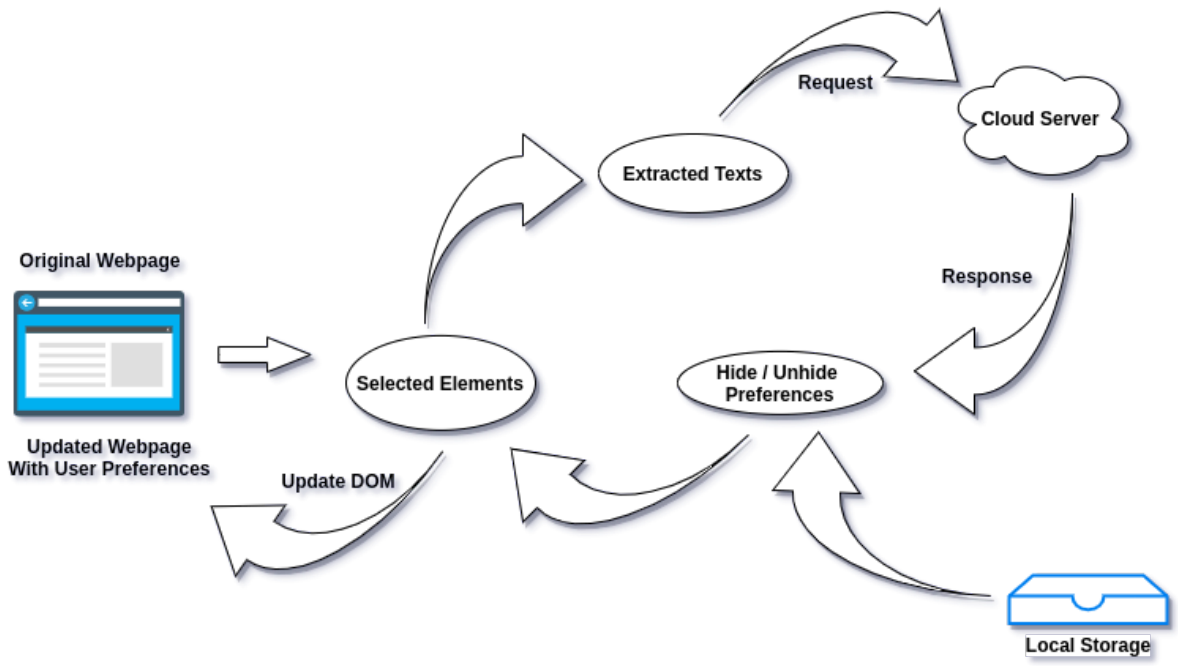
Although these are not part of my proposed implement, I just want to highlight some thoughts that I have on the current implementation of the project -

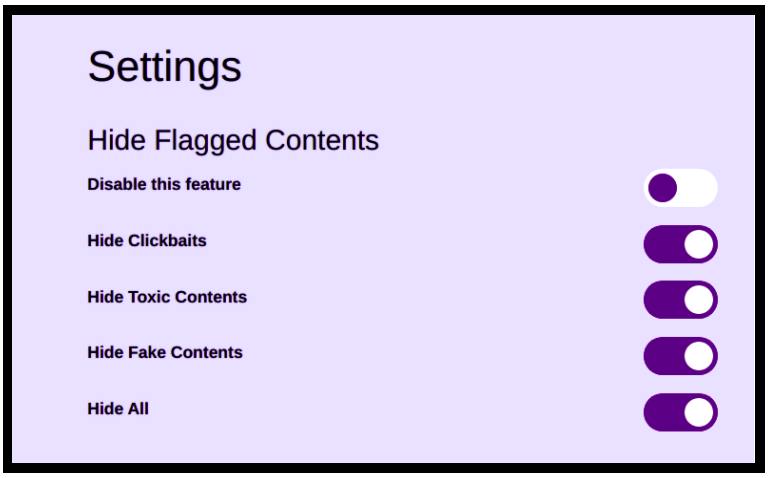
- Repeated requests on scrolling of newsfeed

As much as I understand, when a user scrolls their Twitter/Facebook newsfeed, we send the clickbait/toxic check request to the server for all the loaded posts. However, we can store some local cache (like adding an HTML attribute) for the posts we have already checked, and send check requests only for the newly loaded posts. I could be wrong about the fact that we are sending repeated requests for loaded posts on scrolling of the newsfeed.

- Too many requests on a large page

Since we send a GET request to the server for every post, this could easily overwhelm the server computing resources if multiple users are using it simultaneously. To scale this API properly, we should convert the endpoint to receive 1 bulk POST request from the user containing all the post details as an array and respond with an array. This could reduce the network requests on the server by a magnitude of 10 or more, assuming 10 posts are loaded on average for a user.





Workflow And Prototype For Hide/Unhide Contents

2. Report a post as Fake, Offensive

ML models are prone to errors. We always need an enormous corpus to train them on. As of now, users cannot provide any feedback for the results produced by our model. But this will be the start of the user's interaction with our product.
Having this feature will help us get better and updated results.

Problem 1 :

1. Need a Huge Corpus to train.
2. Needs manual scraping else use publicly available dataset.
3. Hence needs an automated way to store data.

Solution :
I'll create an API :

1. Report posts as fake. (End-point 1)
2. Report posts as toxic/offensive. (End-point 2)

Through API, we can store data in the database. Further, we can use that data to update(train) the deployed model. Hence data scraping becomes automated*.

* I'll store the count of reports. We can have a threshold for marking a post as fake or toxic. The one having counts greater than the threshold will be automatically labeled. Later we can mix it with our training corpus.

Problem 2 :

From the above proposal, if a user reports content, the user doesn't see the Fake/Toxic label. Why? Because the model hasn't been updated yet. How will they get a personalized experience?

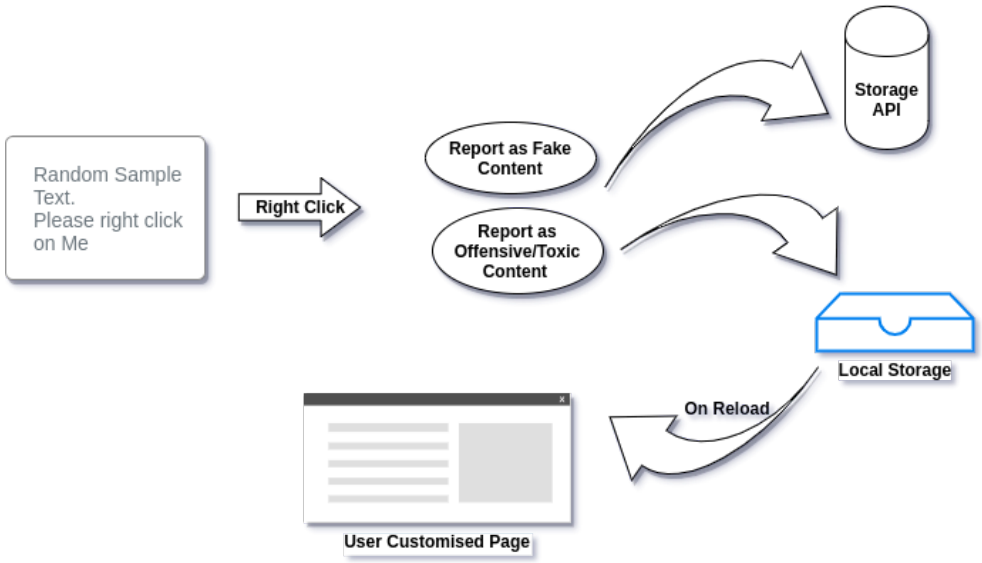
Solution :

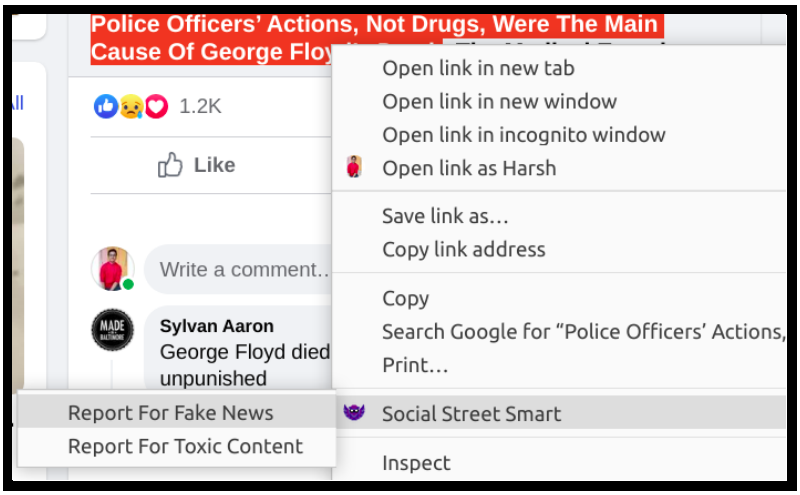
Until the model has been updated, we can personalize it for every user by storing the reported content in user chrome local storage. When an API request is to be sent, we can check for local cache first, and if nothing is found, we can send an API request. This would further reduce requests for the same content as the user encounters again.

/* How to reset the stored data after some time? * /

How will it help?

Help users constantly get updated results, provide an easy feedback system to users.





Workflow and Prototype For Report

3. Add Labelling of posts as Fake news, just as clickbait and toxic comments.

Use the existing Fake news API for this feature. (*Discussion ongoing at issue link*)The fake news API is currently available with a right-click, but adding this just as a label (as in Clickbait) will add more visibility to this feature.

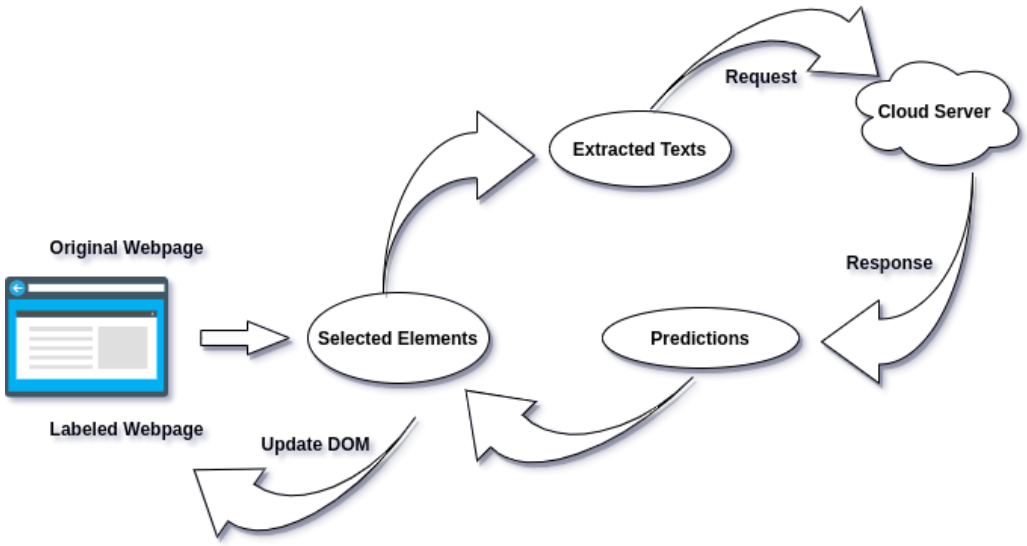
As proposed in **Feature 1**, this will also need its toggle for auto-hide.

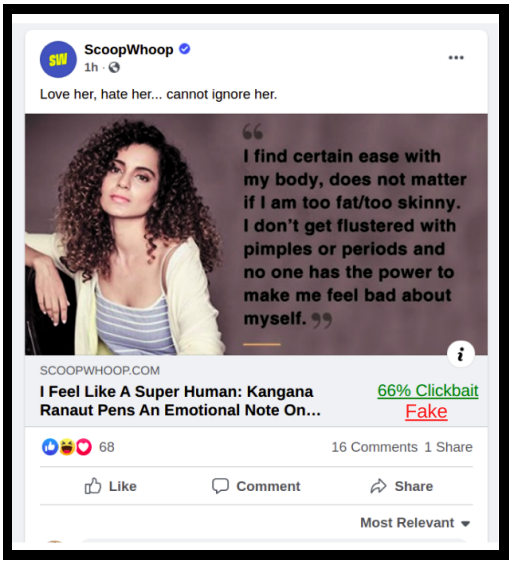
How will it help?

Will display a Fake News label if it's most probable, users don't need to suspect, and then follow the trivial process of right-clicking and then checking, hence increasing user experience.

How will I implement it?

We'll use the Fake news API deployed, send a request to the server, get predictions, and based on the same, we'll modify the DOM accordingly. For the auto-hide feature, we'll implement it the same way we did for Feature 1.





Workflow and Prototype for Fake News Support

4. Disinformation in images

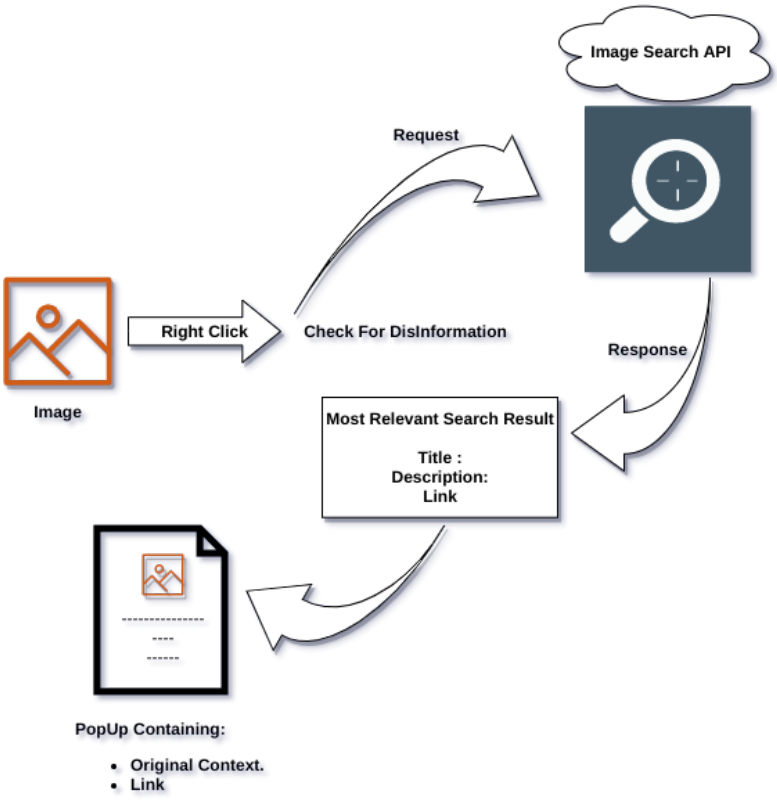
Public manipulation using Image Misinformation has now become a common threat. Adding some wrong contexts on an Image and then sharing widely on social media made it very easy for ordinary users to get manipulated. Adding this support would enable the user to get the correct context, hence having a check on such spreads.

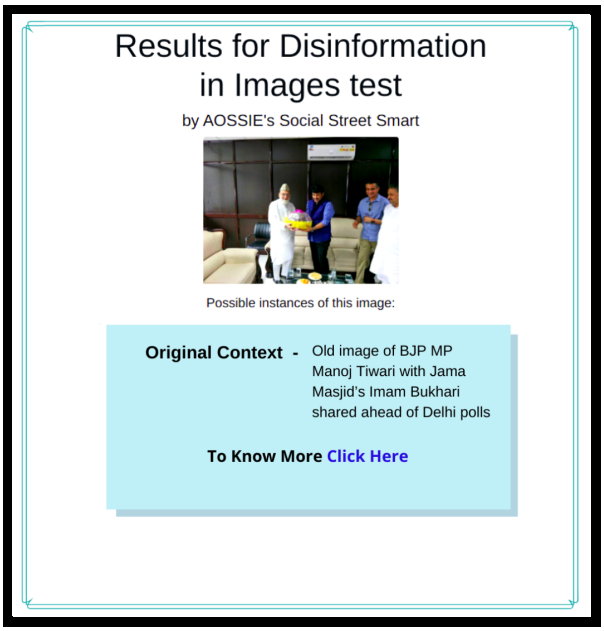
How will it help?

The user can right-click on the suspicious image and get the correct context; hence it will help control Disinformation propagation.

How will I implement it?

As of now, we have the Disinformation in images option on the right-click. ~~This functionality is not yet in working mode.~~ *(Discussion ongoing at issue link)*
.I am proposing to use google image search API and get the first search description, which will help get the image's main context. Also, we could provide a link to the primary context for user engagement.
(*Discussion related to API Keys ongoing at* *issue link*)





Workflow and Prototype For Disinformation In Images

Timeline (Tentative)

From my past experiences, I can say that things can change in a minimal time. For example, some bug comes up with a high priority level which needs to be fixed before any more work. However, having a timeline is always suitable for any project. This, at least, helps in keeping a tab on how much you are lagging.

Pre-GSoC period

Before the coding begins in June second week, there is a lot of homework for me to do. I have to revise core concepts of Javascript and other relevant tech stacks to be used. As the existing Codebase is not very vast, so it is good that most of my time won't be spent reading the code/documentation before proceeding.

Community Bonding

During this period, I shall study the existing Codebase. I shall discuss new approaches regarding different feature implementations. I will be keeping notes too! I shall also discuss the further plans regarding the scalability and other relevant topics regarding the same. Also, I will be active on Gitter and actively communicate with other people in the organization.

The plan for GSoC 2021 will be :

I am planning to split the project into three phases :

- Phase 1: Improve Existing Frontend: To create a great user experience.
- Phase 2: Implementation and testing new Features: To help in the smooth functioning of the application.
- Phase 3: Project Setup Cleanup and Documentation: Help new contributors grasp quickly and contribute.

Timeline

Period	Task
Week 1	• Open issues for all the tasks to be done. • Update UI for Settings Page.

[June 7 - June 13]	1. Add Switches for Hide/Unhide.
Week 2 [June 14 - June 20]	• Add Fake News Labeling support for 1. Twitter 2. Facebook 3. Reddit • Add CI/CD Tests. 1. For each platform.
Week 3 [June 21 - June 27]	• Implement auto-Hide / unhide Option for : 1. Twitter (clickbait ,toxic and Fake News). 2. Facebook (clickbait,toxic and Fake News). 3. Reddit (clickbait,toxic and Fake News).
Week 4 [June 28 - July 4]	• Create an API to store reported data. 1. create two endpoints (fake,toxic/offensive). • Add Report option on context menu. • Documentation of the API.
Week 5 [July 5 - July 11]	• Integrate Report API with frontend. 1. Add CI/CD Tests for API 2. Add CI/CD Tests for Integration.
Mid Evaluation [July 12 to July 16]	
Week 6 [July 16 - July 25]	• Integrate Report UI with local storage. • Add CI/CD Tests. • Add Documentation.
Week 7 [July 26 - August 1]	• Support For Image Disinformation 1. Improve UI. 2. Implement using Google Image Search API. 3. Add Documentation.
Week 8 [August 2 - August 8]	• Integrate Google Image Search API with frontend. • Add CI/CD Tests.

Week 9 [August 9 - August 15]	• Add Documentation 1. Fake News Labeling Support. 2. Auto Hide/Unhide. 3. And rest CI/CD Tests.
Week 10 [August 16 - August 22]	• Bug fixing period. • Buffer time for any critical blockers. • Any further enhancements.
Final Evaluation [August 23 - August 30]	

End-Term evaluation

Goals: Wrap up, Bug fixes

By this time, I will ensure that none of the above implementations has introduced any bugs and are complete by documentation and testing. I will extend this period to my **Future Work** by writing blogs explaining full features and preparing for a major release.

I will update my progress weekly to my mentors and incorporate their feedback and suggestions as and when required. I am optimistic enough about completing the project on time. I have also kept a buffer week before final evaluations for any critical blockers.

Tentative Spare Tasks

The above timeline has been made with keeping in mind the maximum time any task could take and the worst case scenario of having many bugs. Most probably, I will do the work before the end of the GSoC period and hence I would love to work on the below-mentioned task. I would be working on these irrespective of GSoC selection.

- **Improve existing models.**

There is always a scope for improving machine learning predictions by fine-tuning the model, changing the model, or increasing the training data. I'll try to add new existing public data and train the same model. Changing the model may be an option, but we must think of scalability too. Using BERT or RoBERTa will surely increase the accuracy, but the scalability might not be.
This is also a task I look forward to be doing perpetually and being involved in the community.

Post-GSoC Plans

I'm **not** applying for GSoC under any other organization this year since I am motivated by Social Street Smart's journey, and I want to be an integral part of it.

I've already mentioned additional enhancements, which I would love to do within the GSoC time, and if time constraints pose an issue, I will do after that. I'll also be looking forward to contributing to Shift News Origin functionality to Frontend. It was another mini-project that fascinated me on the ideas page. Due to limited Google Search API requests, this functionality is not scalable. Hence it needs to be shifted to Frontend, and users may use their API keys, which will make this easier to use.

News Origin Functionality will help people in several ways, the primary one being Reliability. It will help you to check the authenticity of the news. It will locate the news segment's source, and based on the origin, you can decide if the news is reliable or just picking up steam due to social media and deliberately created to spread fake information.

The sheer idea of functionality that helps users affirm rumors, or any genuine segment, is an exciting and groundbreaking milestone. In not much time, it'll become an integral part of a user's everyday life. Therefore, having support for it will help Social Street Smart become more beneficial to the world.

I would love to work on this (in my spare time) if the project is not done this time in Google Summer of Code. In case the project is completed this time, I'll try to provide bug fixes or any other enhancement for it since I want to contribute to the project in some way.

Along with that, I'll always be a part of AOSSIE and will be following (as well as contributing towards) its development. I will always be available to make changes to Project Social Street Smart and expand its feature set as and when need be.

Deliverables

- A new look of Social Street Smart UI with new features and user-friendly design.
- A new Gitlab Wiki to provide the user and developer documentation.
- Beginner-friendly readme for easy installation with detailed steps.

Other Deliverables

- **Contribution to Social Street Smart:** Apart from the deliverables mentioned above, I would also get more involved in the Social Street Smart ecosystem by contributing to it and helping other contributors.
- Non-coding tasks: There are several miscellaneous non-code tasks that I would like to take up to give back to the community , such as mentoring.
- Adding more new technologies, improving the tool with innovations into the project.

Future Scope

This project will provide a foundation step for making Social Street Smart more user-friendly and scalable.

I wish to remain as an active maintainer of Social Street Smart repository. I will also address all the future bugs encountered and feature additions.

Previous Open Source Contributions

PRs :

1. Added auto discovery of .test files [Open] https://github.com/python/mypy/pull/10108
2. Fixed out of memory error while using update_bn in SWA (Pytorch) [Merged] https://github.com/pytorch/pytorch/pull/52654
3. Added Support for Twitter [Merged] [https://gitlab.com/aossie/social-street-smart/-/merge_requests/81
4. Added Support for Facebook [Open] https://gitlab.com/aossie/social-street-smart/-/merge_requests/82
5. Added Test for Facebook HateSpeech [Open] https://gitlab.com/aossie/social-street-smart/-/merge_requests/83
6. Ported client side API calls from sync to async [Open]https://gitlab.com/aossie/social-street-smart/-/merge_requests/84
7. Fixed lookup time for website info [Open] https://gitlab.com/aossie/social-street-smart/-/merge_requests/85

The Motivation for GSoC

I am a typical geek who loves programming and enjoys problem-solving and making side projects a part of hobby coding. Also, I am an active contributor to a university-level open-source community, CyberLabs, where I have worked on various open-source projects.
The pride in the feeling that my code will cause an impact in the lives of millions of people who will use it is unparalleled. Moreover, it allows me to grow as an individual and learn how to work in an enormous community team. I have also been active in introducing people to the world of open source and getting them involved with various open-source projects and communities.

Achievements:

1. Secured Bronze Medal in Inter IIT-Tech Meet 2021 with a team of Ten.
2. Received Kaggle Expert Tag.
3. Secured 2nd Rank in Movie Genre Classification Competition, Kaggle.
4. Secured 2nd Rank in Fake News Detection KDD, Kaggle.
5. Secured 6th Rank in Melanoma Skin Cancer Detection, Kaggle.
6. Secured 20th Rank in Age Detection Competition.

7. Secured 28th Rank in Wind Speed Detection Competition, DataDriven.
8. Secured 31st Rank in LYFT autonomous Vehicle motion prediction, Kaggle with a team of two.
9. Secured 43rd Rank in Toon Emotion Detection Competition (HackerEarth)
10. Secured 138th Rank in MOA competition Kaggle.