# Supplementary Material – Safire

**Project Title:** Safire – The Harassment Saver  
**Team Members:** Aditya Sharma, Anamika Raj, Pari Gupta, Sugndha Agarwal  
**Team Name:** Smooth Operators

## Table of Contents

1. Introduction
2. Reference Documents & Research Papers
3. Tutorials, Guides & Resources Followed
4. Code Snippets / Algorithms
5. Sample Data & Database Details
6. Screenshots & UI Designs
7. Future Work / Planned Enhancements
8. References & Resources

## 1. Introduction

This supplementary file provides additional materials, resources, and references used during the development of Safire. It includes research papers, tutorials, detailed screenshots, sample data, code snippets, system diagrams, and suggested future enhancements.

## 

## 2. Reference Documents & Research Papers

| Document / Paper | Summary / Relevance |
| --- | --- |
| *“NLP for Hate Speech Detection”* | Techniques for detecting harassment using tokenization, sentiment analysis, and classification. |
| *“Browser Extension Security & Privacy”* | Guidelines for implementing safe extensions handling sensitive user data. |
| *“AI Ethics in Moderation”* | Defines ethical boundaries and user consent policies for automatic message detection. |

## 

## 3. Tutorials, Guides & Resources Followed

| Resource | URL / Notes |
| --- | --- |
| Plasmo Extension Documentation | https://www.plasmo.com/docs – Used to build the Chrome extension. |
| Puppeteer Automation Guide | https://pptr.dev/ – For automated screenshot capturing for evidence. |
| Google Gemini AI Documentation | https://ai.google.dev – AI moderation integration. |
| React + Tailwind CSS Tutorials | Various YouTube/blog tutorials – UI for dashboard and extension. |

## 4. Code Snippet / Algorithms

**Code Snippet:**

router.post('/detect-harassment', async (req, res) => {

const cacheKey = JSON.stringify(req.body);

const cached = await redisClient.get(cacheKey);

if (cached) return res.json(JSON.parse(cached));

for (const server of servers) {

try {

const response = await axios.post(`${server.url}/detect`, req.body);

await redisClient.set(cacheKey, JSON.stringify(response.data), { EX: 3600 });

return res.json(response.data);

} catch { continue; }

}

res.status(503).json({ error: "All servers failed" });

});

**Algorithm:**

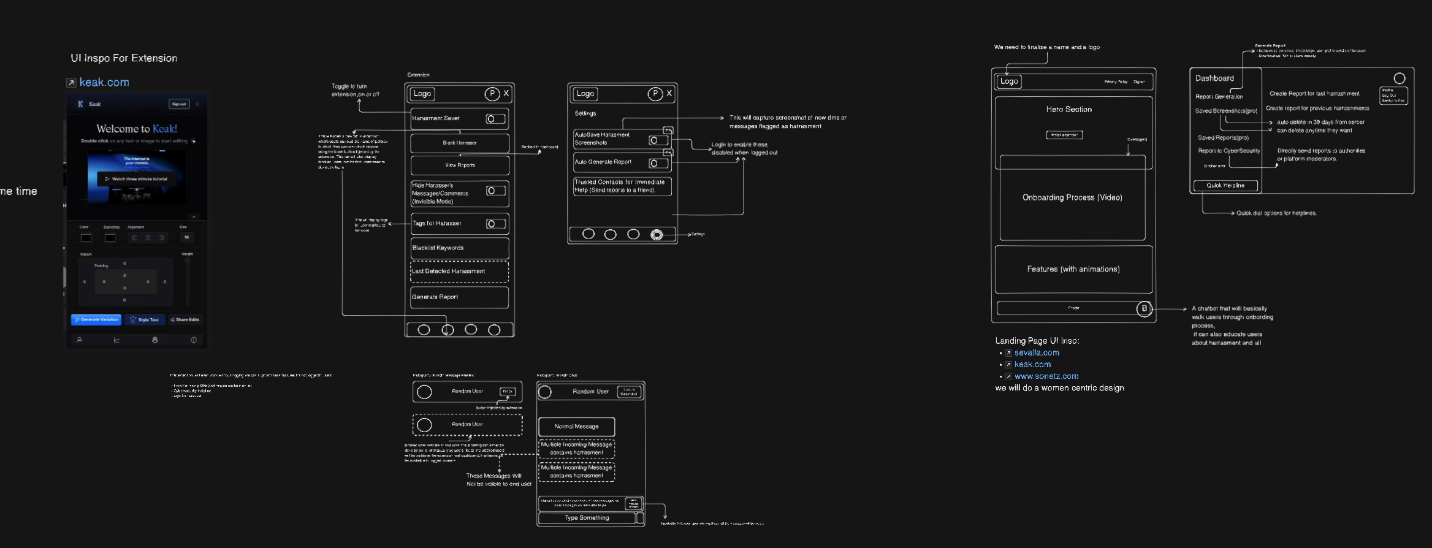
1. **Receive Request: /detect-harassment with { message, platform }.**
2. **Check Cache: Return cached result if available.**
3. **Select Server: Use load balancing (weighted, least connections, fastest response).**
4. **Circuit Breaker: Skip unresponsive servers.**
5. **Send to AI Server: Get harassment analysis JSON.**
6. **Cache & Respond: Store result in Redis and send to client.**
7. **Fallback: Return error if all servers fail.**

## 5. Sample Data & Database Details

* **Database used:** MongoDB
* **Collections:** users, messages, reports, hidden\_users
* **Sample Document – messages:**

{  
 "\_id": "msg001",  
 "sender": "John Doe",  
 "recipient": "Jane Smith",  
 "message": "You are incompetent",  
 "timestamp": "2025-09-05T10:15:00Z",  
 "flagged": true  
}

## 6. Screenshots & UI Designs

**Initial Wireframes:** -  


**Extension in Action:** -

A screenshot of a chat

AI-generated content may be incorrect.

**Dashboard UI:** - Screens screenshot of a computer

AI-generated content may be incorrect.

## 7. Future Work / Planned Enhancements

* Expand extension support to Instagram, WhatsApp, Twitter, and Teams.
* Add voice/video harassment detection.
* Implement multi-language support.
* Enhance community-driven moderation with reporting and collaborative blocking.
* Optimize AI model for lower latency and reduced false positives.

## 8.. References & Resources

* Plasmo Docs – https://www.plasmo.com/docs
* Google Gemini AI – https://ai.google.dev/
* React Docs – https://reactjs.org/docs/getting-started.html
* Tailwind CSS – https://tailwindcss.com/docs