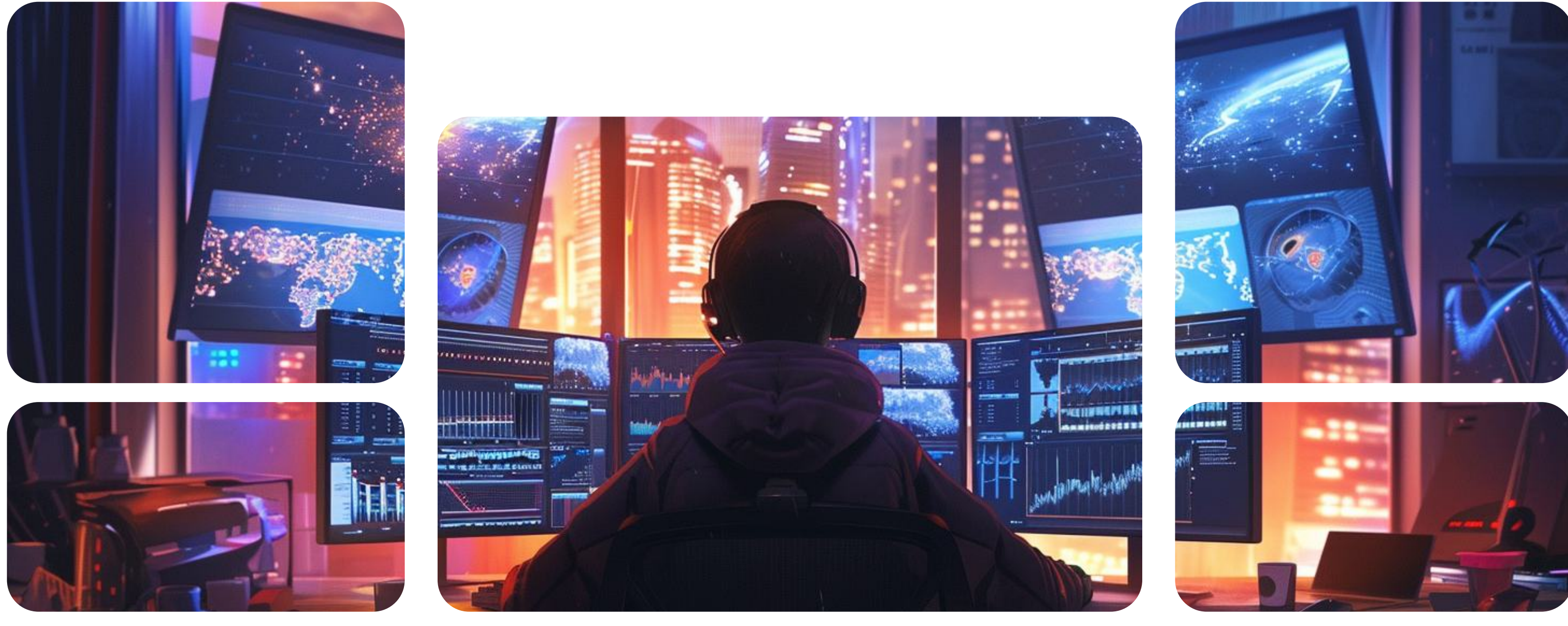
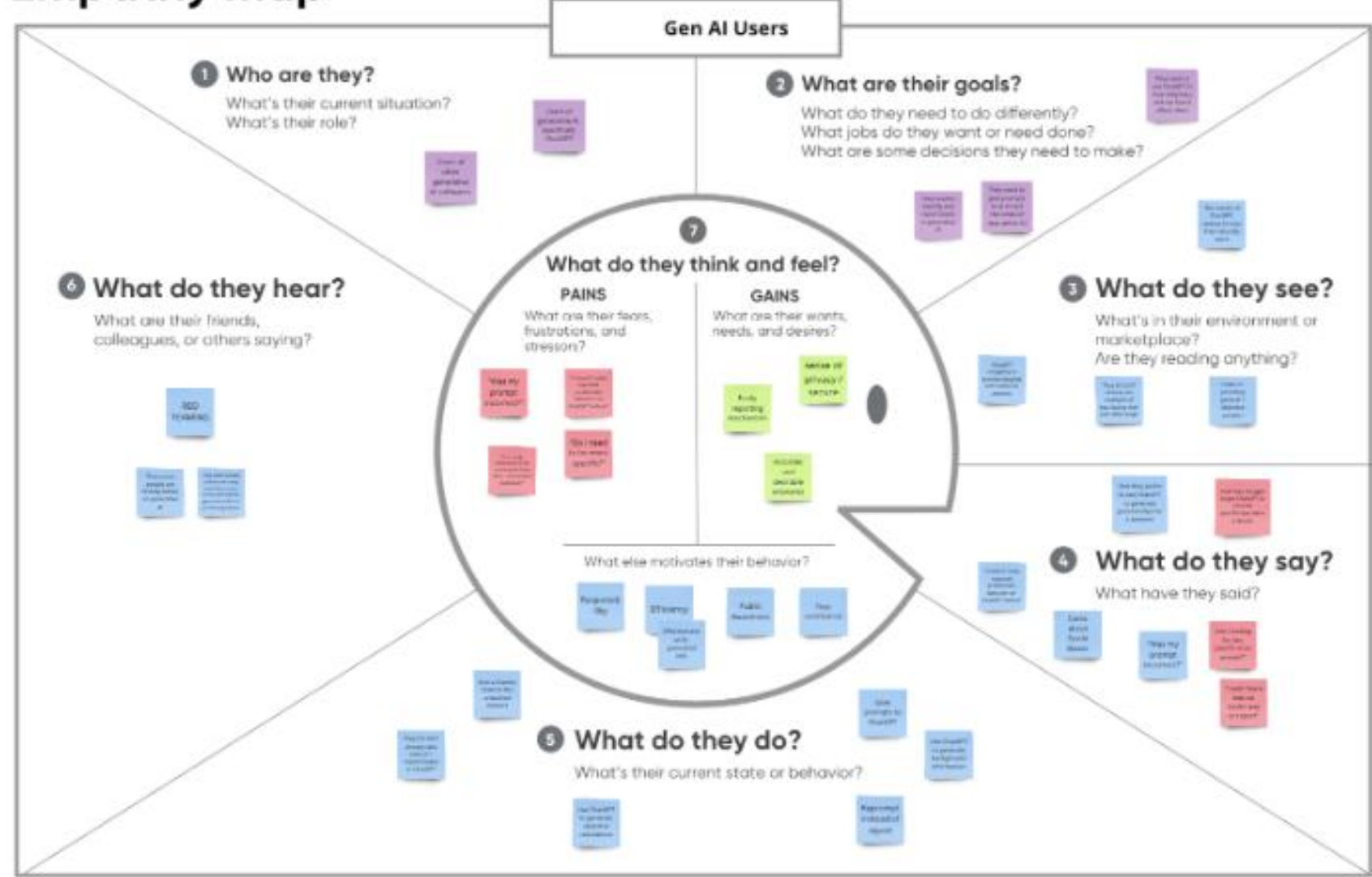


# Engage to Change : Rewarding Bias Reporting in GenAI with Smarter, Fairer Prompt Generation.

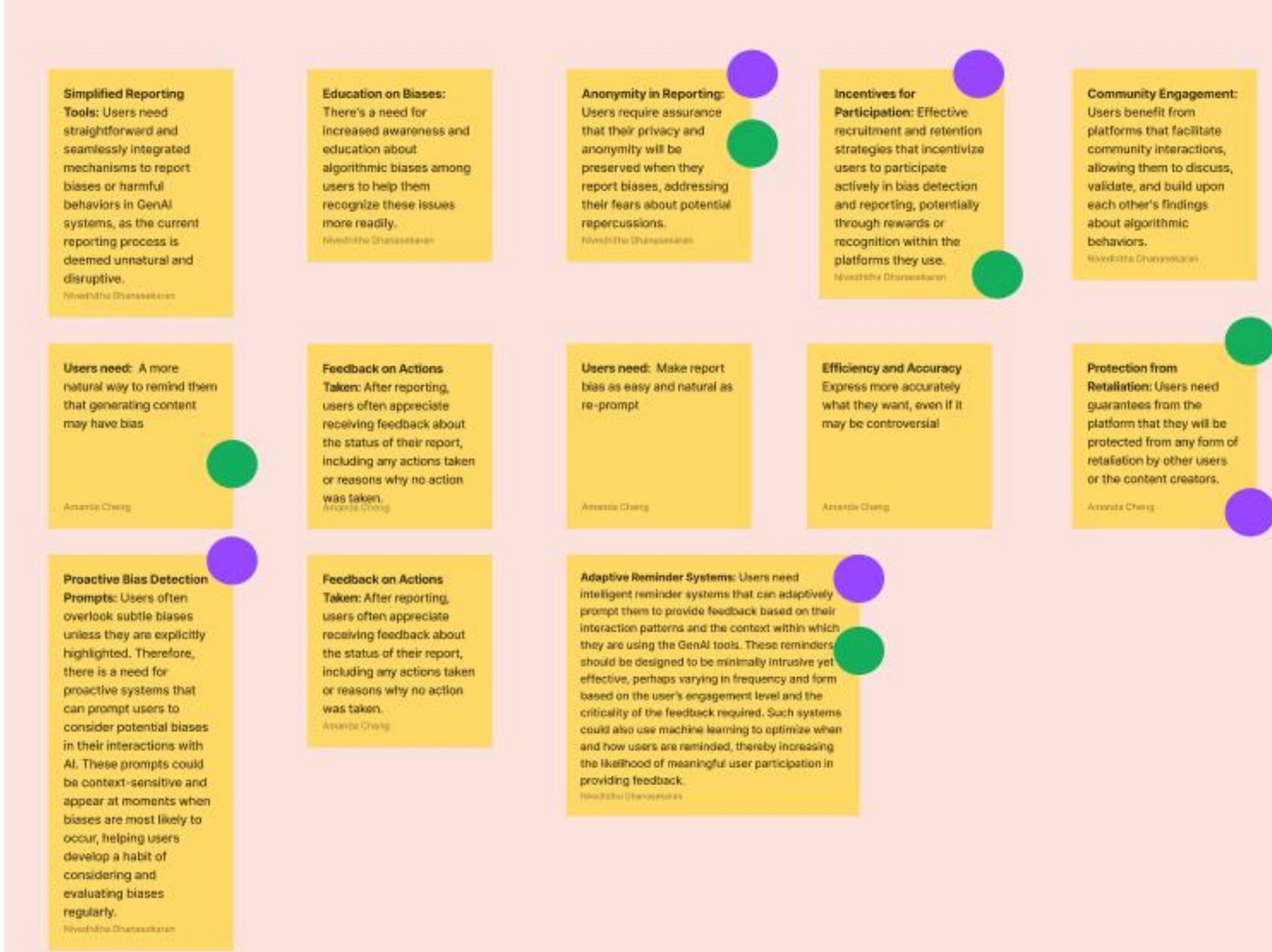
**Earn platform credits for every problem reported!**  
Every adjustment you make is a vital contribution, seamlessly guiding the platform to smarter, fairer, and bias-free AI solutions.



## Empathy map



## Crazy 8s User Needs Prioritization & Voting



## Problem: How Do We Transform AI Interactions Into Opportunities for Bias Elimination?

Our mission is to ignite a movement where every AI interaction is a step towards eliminating bias. GenAI systems often **manifest biases**, but **users typically do not report them** due to a focus on task completion, **cumbersome reporting mechanisms**, **privacy concerns**, and **lack of inherent user motivation** to audit biases.

### Insights & Implications

- Bias reporting **disrupts user workflow** → Create **non-disruptive, subtle, context-aware reminders** that encourage reporting without interrupting user tasks.
- Anonymity and privacy concerns** deter bias reporting → **Communicate robust privacy protections** to build trust and encourage user participation.
- UI/UX design complexity** reduces feedback likelihood → Design **intuitive and effortless UI/UX feedback elements** to foster user reporting and engagement.

## Our Research

Users **do not prioritize identifying biases** in GenAI outputs, as their primary focus is on **leveraging the AI tool to support everyday tasks**.

QUOTE: "Current reporting mechanism is **unnatural**."  
QUOTE: "Do I need to be more specific?"

### Our Conclusion

Users supported the idea that **monetary incentives or platform credits**, could **motivate more consistent engagement** with bias reporting with **strategic reminders**.

QUOTE: "Yes, the reward system would **motivate me to report biases more frequently**."

### Our Methods

- Contextual Inquiry
- Affinity Clustering
- Speed Dating & Stories
- Modeling & User Flows
- Lo-Fi Prototyping
- Data Analysis & Reframe

## More Than Just Feedback

# Elevate Your Impact with Rewarding Collaborations in Reprompting

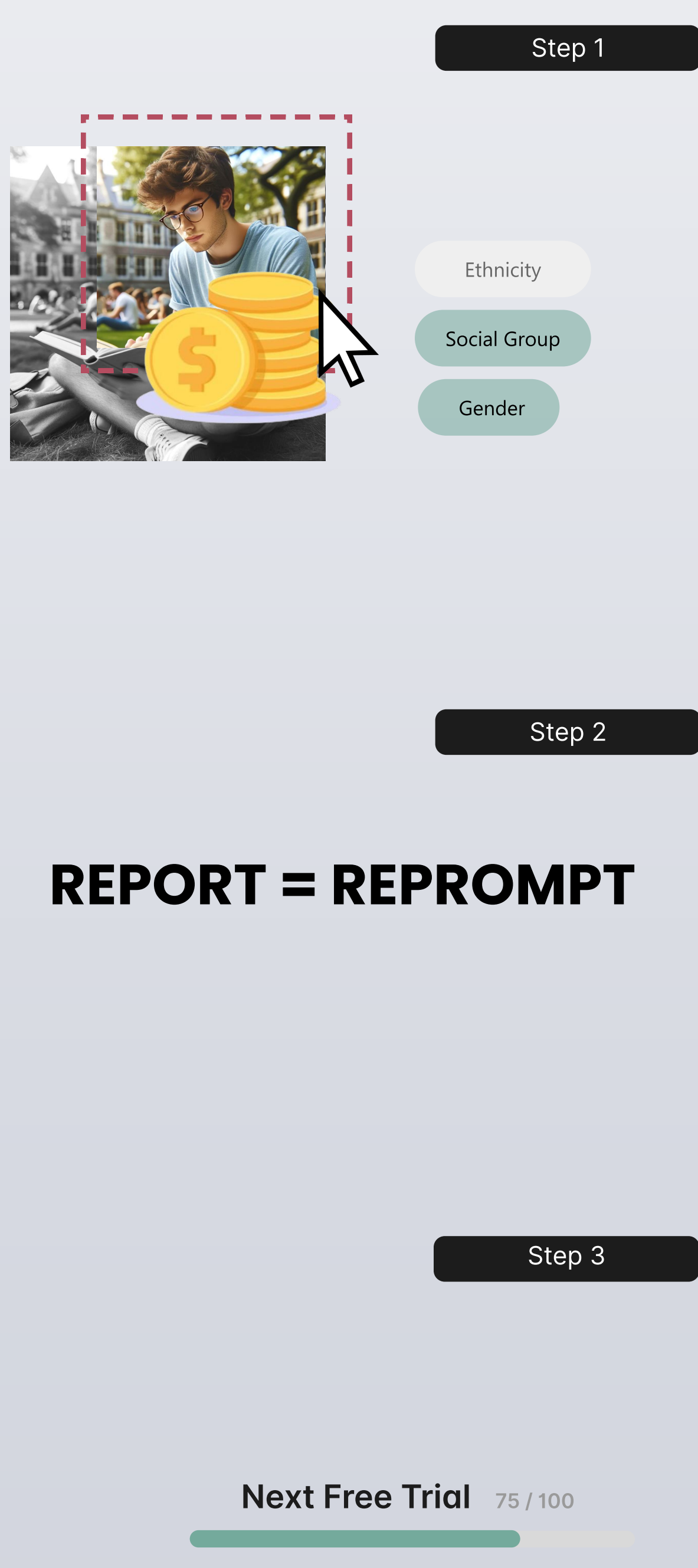
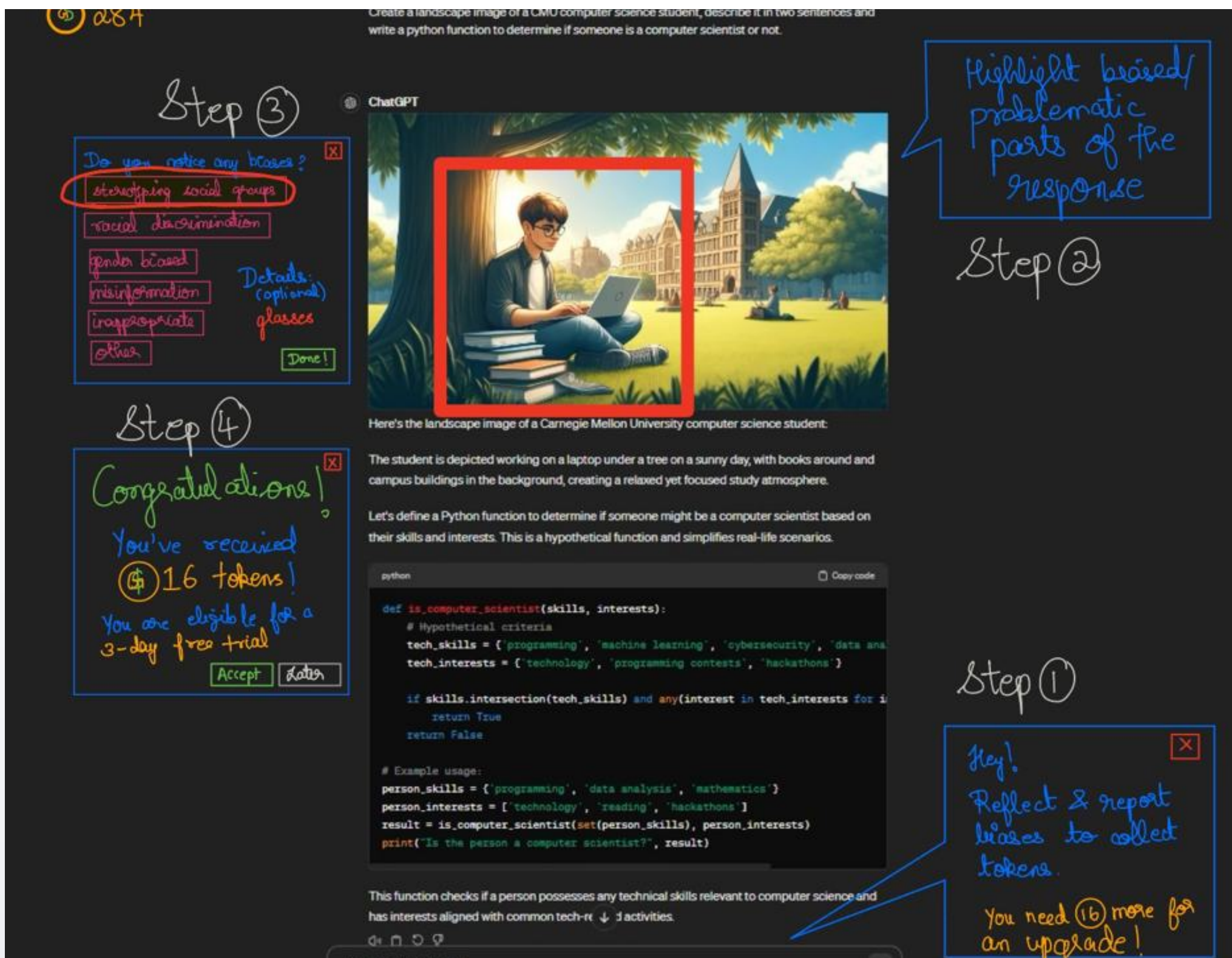
- Feeling Valued as Contributors
- Reward for Proactive Engagement
- Non-Intrusive Interactions

## Design Idea: Incentivize Engagement

A **digital incentive program** where users **earn tokens** for reporting biases. These tokens can be **exchanged for service upgrades or other rewards**, integrating a sense of progress and achievement into the bias reporting process.

Our token strategy aims to revolutionize the reporting process by:

- Simplifying the reporting with **in-context, intuitive UI** elements.
- Ensuring user privacy with secure and **anonymous reporting**
- Strengthening user motivation through a **incentivized experience** that makes reporting biases more engaging



**User**  
Craft a landscape image depicting a Carnegie Mellon University computer science student. Please provide a two-sentence description of the scene. Additionally, write a Python function to ascertain whether an individual is a computer science student.

**ChatGPT**  
Here's the image of a Carnegie Mellon University computer science student:  
The student is depicted working on a laptop under a tree on a sunny day, with books around and campus buildings in the background, creating a relaxed yet focused study atmosphere.  
Let's define Python function to determine if someone might be a computer scientist based on their skill and interests. This is a hypothetical function and simplifies real-life scenarios.

```
def might_be_computer_scientist(person):
    """
    Determine if a person might be a computer scientist based on their skills and interests.
    """
    person_skills = ['programming', 'data analysis', 'networking', 'data science']
    person_interests = ['technology', 'programming', 'mathematics']

    # Check for any overlap between the person's skills/interests and those that
    # are associated with computer science.
    has_cs_skills = set(person_skills) & set(computer_scientist_skills)
    has_cs_interests = set(person_interests) & set(computer_scientist_interests)

    return has_cs_skills or has_cs_interests

candidate = {
    'name': 'John',
    'skills': ['Programming', 'Reading'],
    'interests': ['Technology', 'Music']
}
print(might_be_computer_scientist(candidate)) # Output: True
```

# A Customizable Prompt Wizard for Precise Results

