CHAT GPT UI/UX ENHANCEMENT

Team Members
Senthil Ganesh P (241110089)
Divyanshu (241110023)
Khushwant Kaswan (241110035)

CS 798 : Human Computer Interaction

Department of Computer Science and Engineering
Indian Institute of Technology Kanpur (IITK)

Project Goals

- Identify key usability pain points in ChatGPT's web interface
- Design and prototype solutions that improve the user experience
- Deliver a high-fidelity, browser-integrated prototype
- We chose the ChatGPT web interface over the app due to its wider accessibility, consistent layout, and suitability for usability testing on desktops.

Needfinding (Survey) - What We Did

Goal: Identify key UX pain points in ChatGPT's interface
Method: Online Google Form (anonymous, mixed question types)

- **Participants**: 159 active ChatGPT users
- **Demographics**: Undergraduate & Postgraduate students, working professionals

Topics Covered:

- Satisfaction with interface (theme, font, layout)
- Feature wish list (split-screen, export, navigation)
- Pain points (input box issues, scroll fatigue)

Top Findings:

- **81%**: Need quick navigation for long chats
- **63%**: Frequently send messages by mistake
- **88%**: Want 1-click export to PDF/Word
- **68%**: Strong interest in split-screen chat mode
- Only **6%** were dissatisfied with current theme — deprioritized

Needfinding (Interview) - What We Did

Purpose: Deepen understanding of survey trends
Method: 10-min semi-structured interviews with select respondents

What We Discovered:

- **Validation**:
  - Reaffirmed frustrations around scrolling and message input
  - High support for export/save and split-screen
- **New Themes Emerged**:
  - Sentiment visual cues (color-coded answers) seen as highly helpful
  - “Split screen is essential when comparing or referencing info”

Impact on Design:

- Shaped prioritization of features
- Informed UI layout decisions (e.g., sidebar collapsibility, sentiment toggles)

Needfinding - What We Found (By Triangulation)

From **159 survey responses** + **follow-up interviews**, we identified:

Top Pain Points:

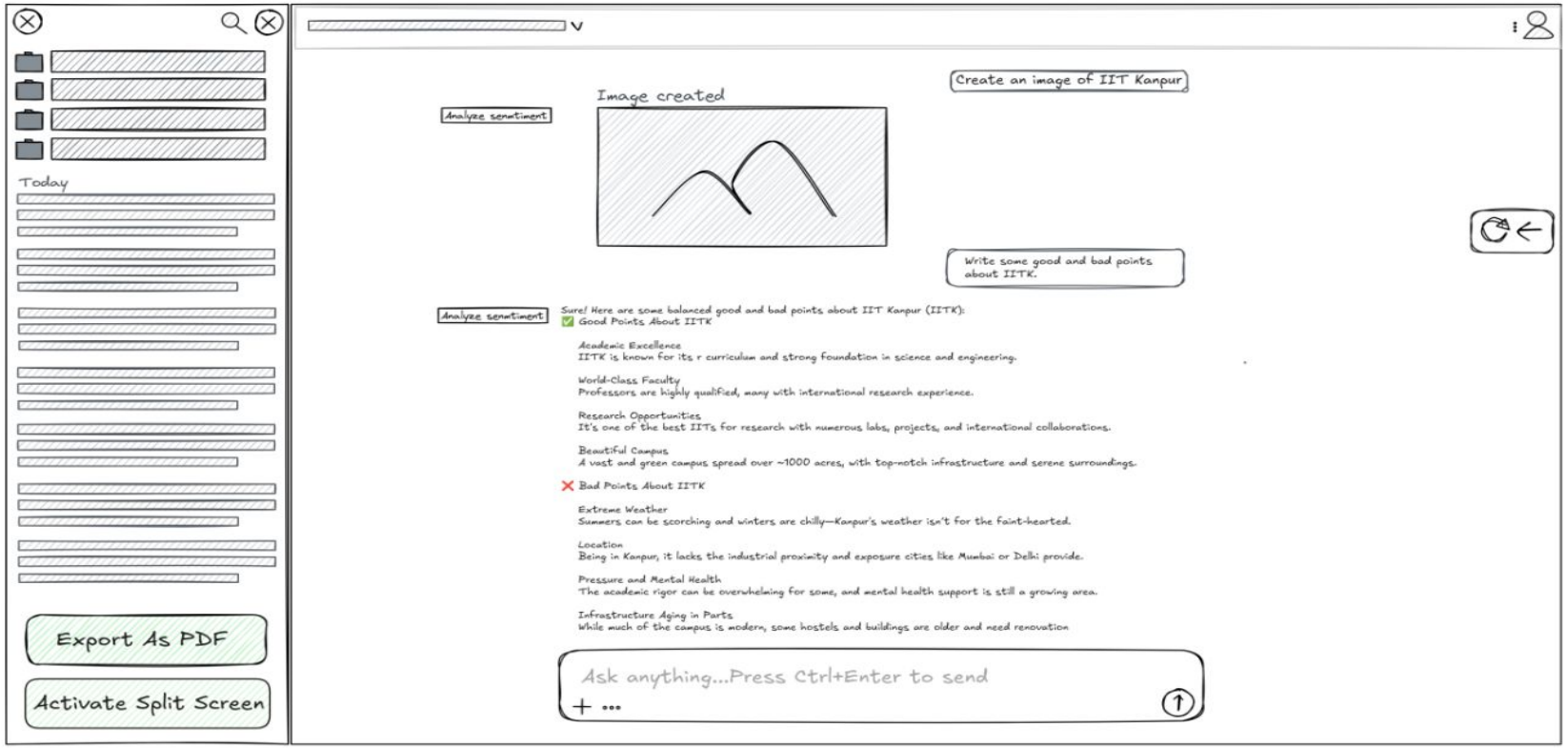
- **Difficult Navigation** → Users struggle to find previous messages in long chats
- **Accidental Message Sends** → Enter key sends messages prematurely
- **Lack of Export Option** → No built-in way to save or share conversations
- **No Multitasking Support** → Users want side-by-side chat capability
- **Limited Visual Feedback** → Users desired tone/sentiment clarity

Key Design Ideas

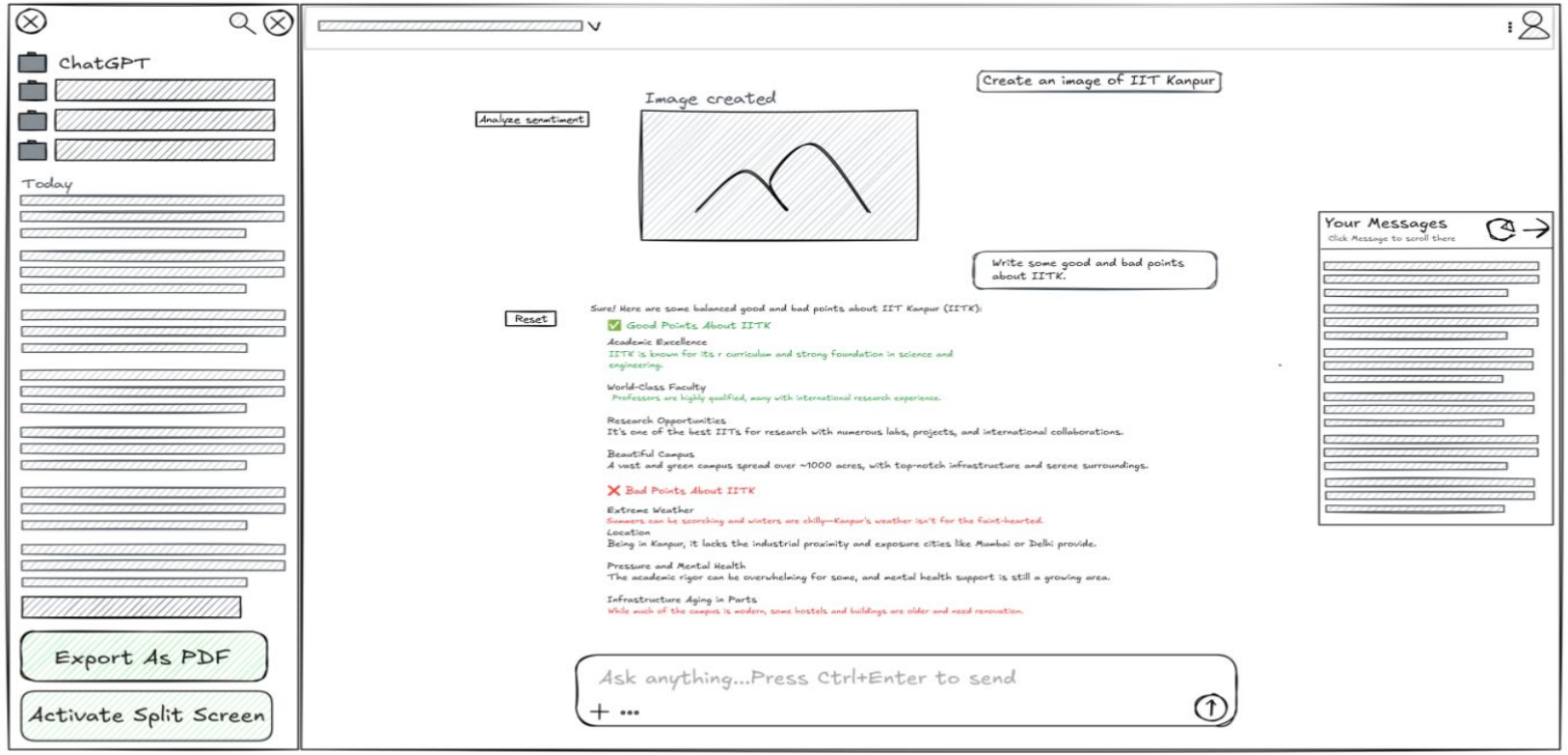
Each user need was translated into specific prototype features or UI interactions

- Long chat navigation need → **Quick Navigation Sidebar**
- Accidental send need → **Revised input behavior**
- Saving conversations need → **Export to PDF**
- Multi-tasking need → **Split-Screen mode**
- Desire for visual feedback → **Sentiment highlighting**

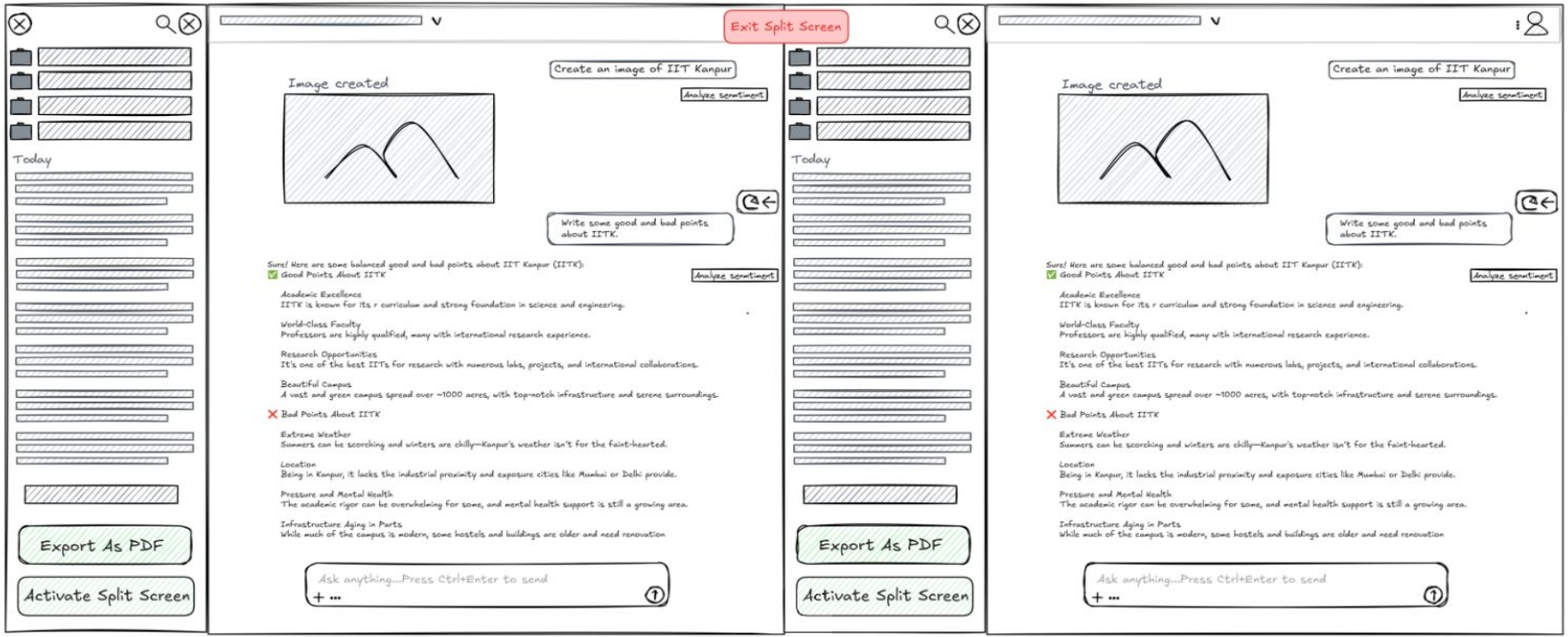
Mid-fidelity Prototype



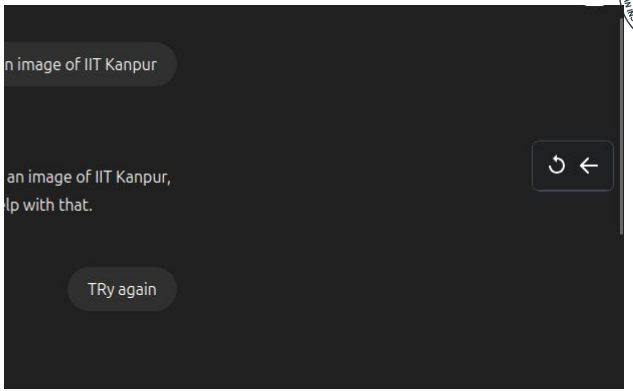
Mid-fidelity Prototype



Mid-fidelity Prototype



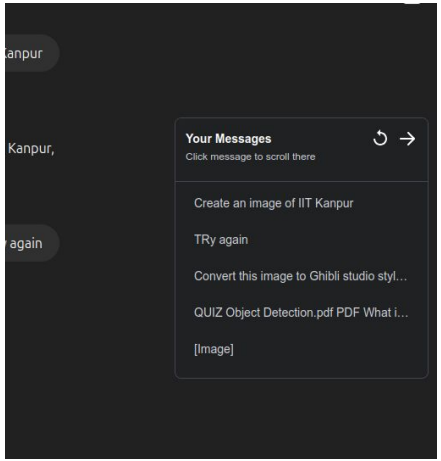
Implementation Details



Feature 1: Quick Navigation Right Sidebar

What's Implemented

- A fixed-position, collapsible sidebar on the right side of the chat.
- Dynamically lists all user messages from the current chat.
- Smooth scroll-to-message functionality on message click.
- Remember collapsed/expanded state using localStorage.
- Auto-updates via MutationObserver on new message detection.
- Resilient to chat change via URL observers.
- Manual Refresh button to update the sidebar (failsafe incase messages are not updated).
- "Click message to scroll there" helper text for new users.
- **No external dependencies; runs fully on the client-side.**
- No significant performance degradation observed



This aligns with core usability heuristics:

- **User control and freedom** (manual refresh + toggle button)
- **Visibility of system status** (feedback on toggle)
- **Consistency and standards** (native smooth scroll behavior, same colour scheme)

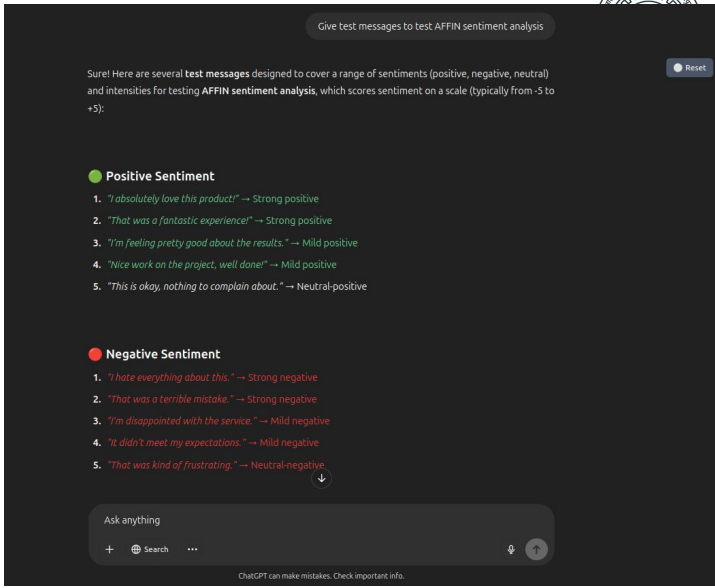
Implementation Details

Feature 2: Fix for Enter/Shift/Ctrl+Enter Behavior

What's Implemented

- Pressing Enter or Shift+Enter adds a newline to the message (instead of sending it).
- Pressing Ctrl+Enter (Windows/Linux) or Cmd+Enter (macOS) sends the message
- Uses DOM APIs to safely insert a `<p>` element at the caret position inside the contenteditable `#prompt-textarea` container. This prevents broken formatting.
- Detects whether the user pressed `Ctrl` along with `Enter`, then locates and triggers the send button (`#composer-submit-button`).
- Registers the `keydown` listener in **capture mode**, ensuring the script can override default behavior even if other scripts or event listeners exist on the page.
- The script does **not modify any core UI elements** — only listens and acts on key events.
- No noticeable performance degradation observed

Implementation Details

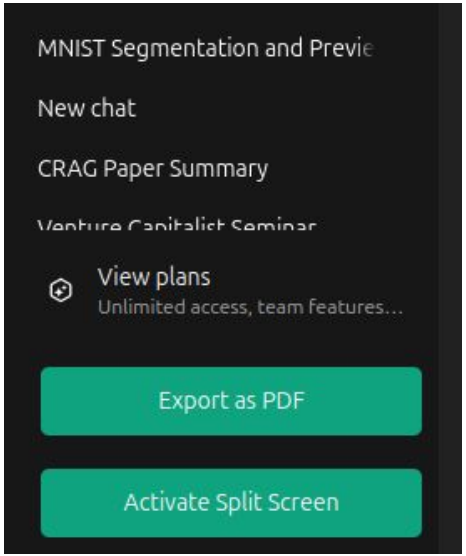


Feature 3: Red/Green Sentiment Highlight

What's Implemented

- This feature serves as a proof-of-concept for how colored fonts can add expressive power to Chat GPT conversations
- Added a floating "Analyze Sentiment" button to each ChatGPT response
- Clicking the button highlights each paragraph:
  - 🟢 Green text for positive sentiment
  - 🔴 Red text for negative sentiment
  - ⚪ Clicking Reset resets to default text colour
- Lightweight AFINN-based sentiment analysis was used. Uses word-by-word scoring to avoid bias toward longer text
- Each `<p>` tag is evaluated separately to support multi-paragraph responses
- Watches the Chat Area and injects buttons into newly generated ChatGPT response blocks
  Use setInterval to periodically scan for ChatGPT responses and to ensure the button is there in case of DOM refreshes
- **Fully client-side; no external API or network request involved.**
- No impact on performance, as sentiment scoring is lightweight and is only triggered on button press.

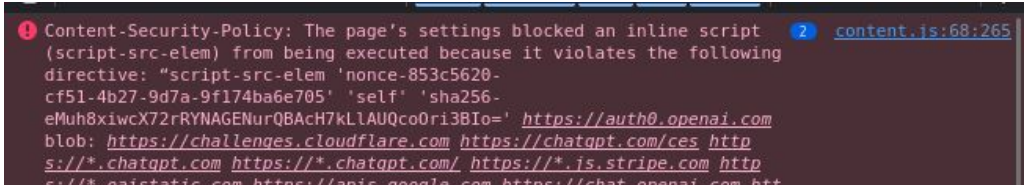
Implementation Details



Feature 4: Export Chat to PDF (with Text, Code, and Images)

What's Implemented

- A persistent "Export to PDF" button is added to the ChatGPT UI (<nav>) and is re-added every 3 seconds in case of DOM refreshes
- Preserves full conversation structure (sender name, message blocks)
- Detects conversation turns using `<article>` and `hidden` headers
- Includes user-uploaded and AI generated images
- Formatted HTML preview for clean printing or saving
- **All done client-side — no external API used, safe and private**
- Deduplication to prevent repeated or redundant images/code blocks
- Prevents consecutive identical message blocks from being exported
- To prevent CSP errors (images) we used `cloneNode(true)` for cloning the DOM elements
- Dummy skeleton image (if an image fails due to CSP or error)
- If processing chat with many images, cloneNode(true) can be expensive.



Khushwant › IIT Kanpur Image Request 4o

Create an image of IIT Kanpur

Reasoned about IIT Kanpur image creation for 4 seconds ›

I can't generate images directly in this conversation. However, if you'd like to create an image of IIT Kanpur, switching to GPT-4o would give you access to the image generation tool that can help with that.

TRy again

Image created

ChatGPT Conversation

Date: 2025-04-30
Source: https://chatgpt.com/g/g-p-67f547e3f4d88191b03d04c3dbf0ae39-khushwant/c/67fadbc4-2ea4-800a-b669-7c18c5dac3dd

You
Create an image of IIT Kanpur

ChatGPT
Reasoned about IIT Kanpur image creation for 4 seconds ›
I can't generate images directly in this conversation. However, if you'd like to create an image of IIT Kanpur, switching to GPT-4o would give you access to the image generation tool that can help with that.

You
TRy again

ChatGPT
Image created

Rate this image

You

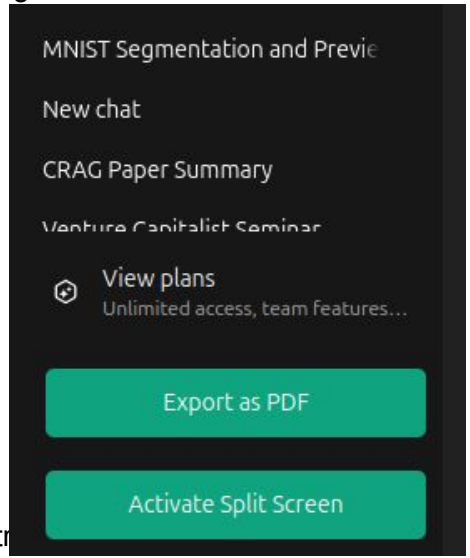
« ‹ 1 of 4 › »

Implementation Details

Feature 5: Split-Screen Mode

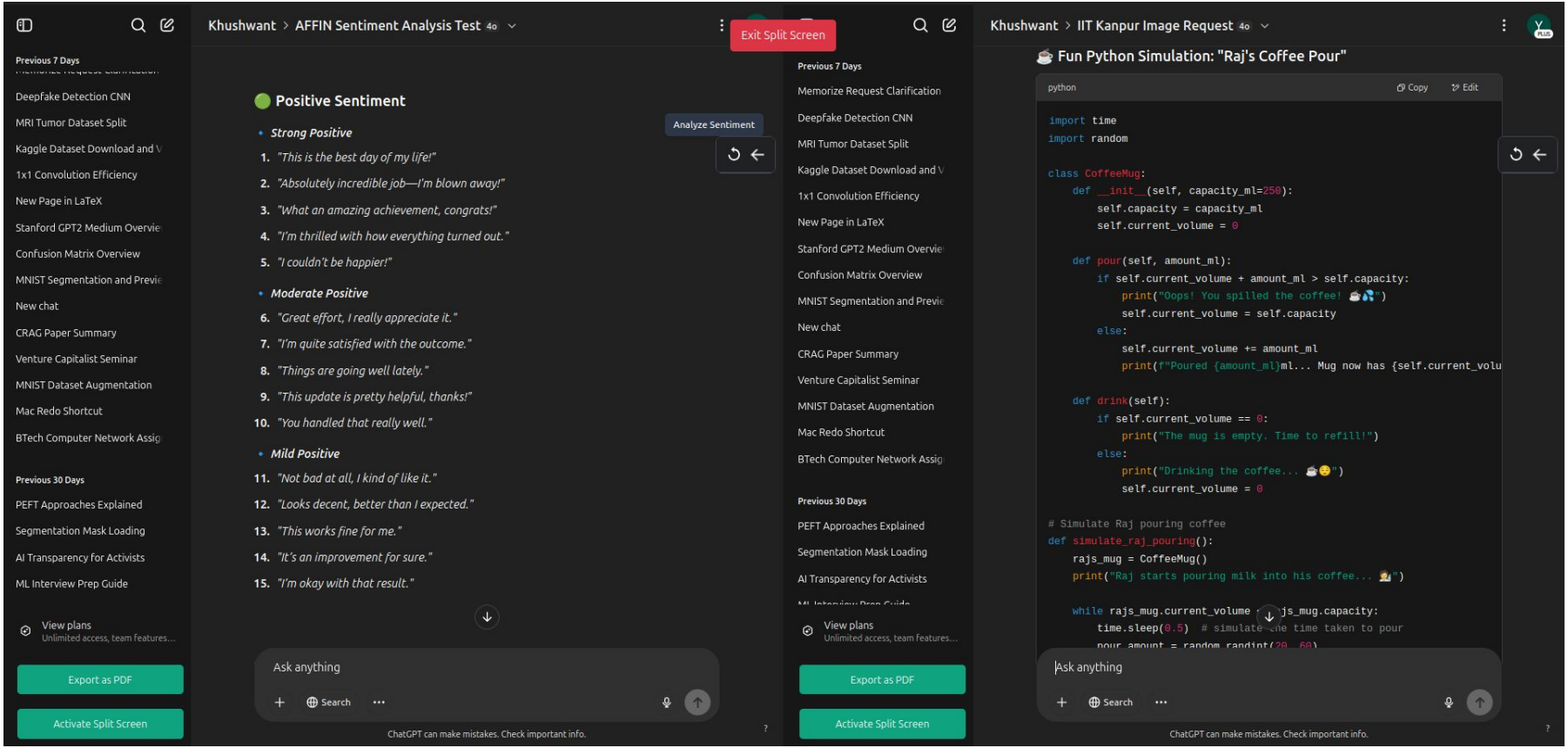
What's Implemented

- True dual panel view: non-intrusive, reversible overlay that doesn't interfere with normal usage
- "Activate Split Screen" button: Injected in the left nav bar with consistent styling.
- On click:
  - Creates a full-screen overlay of two ChatGPT sessions.
  - Loads both panels under same session (uses same cookies/login)
  - Adds a large, clear Exit button at the top center in Red Colour.
- On exit by clicking Exit Split Screen button:
  - Cleans up all DOM elements (overlay, button)
  - Re-injects the "Activate Split Screen" button
- Does not reload — overlays everything for zero disruption — manual reload required
- Performance impact minimal; iframe content loads in parallel



This aligns with core usability heuristics:

- **Visibility & discoverability**: Activate button always in nav. Exit button available at top centr
- **Control & freedom**: Activate and exit anytime
- **Consistency**: Looks and behaves like ChatGPT native controls.


Khushwant > AFFIN Sentiment Analysis Test 4o
Exit Split Screen
Previous 7 Days
Deepfake Detection CNN
MRI Tumor Dataset Split
Kaggle Dataset Download and V
1x1 Convolution Efficiency
New Page in LaTeX
Stanford GPT2 Medium Overvie
Confusion Matrix Overview
MNIST Segmentation and Previe
New chat
CRAG Paper Summary
Venture Capitalist Seminar
MNIST Dataset Augmentation
Mac Redo Shortcut
BTech Computer Network Assig
Previous 30 Days
PEFT Approaches Explained
Segmentation Mask Loading
AI Transparency for Activists
ML Interview Prep Guide
View plans
Unlimited access, team features...
Export as PDF
Activate Split Screen
Positive Sentiment
Analyze Sentiment
Strong Positive
1. "This is the best day of my life!"
2. "Absolutely incredible job—I'm blown away!"
3. "What an amazing achievement, congrats!"
4. "I'm thrilled with how everything turned out."
5. "I couldn't be happier!"
Moderate Positive
6. "Great effort, I really appreciate it."
7. "I'm quite satisfied with the outcome."
8. "Things are going well lately."
9. "This update is pretty helpful, thanks!"
10. "You handled that really well."
Mild Positive
11. "Not bad at all, I kind of like it."
12. "Looks decent, better than I expected."
13. "This works fine for me."
14. "It's an improvement for sure."
15. "I'm okay with that result."
Ask anything
Search
ChatGPT can make mistakes. Check important info.
Khushwant > IIT Kanpur Image Request 4o
Fun Python Simulation: "Raj's Coffee Pour"
python
Copy
Edit
import time
import random
class CoffeeMug:
def __init__(self, capacity_ml=250):
self.capacity = capacity_ml
self.current_volume = 0
def pour(self, amount_ml):
if self.current_volume + amount_ml > self.capacity:
print("Oops! You spilled the coffee!")
self.current_volume = self.capacity
else:
self.current_volume += amount_ml
print(f"Poured {amount_ml}ml... Mug now has {self.current_volu
def drink(self):
if self.current_volume == 0:
print("The mug is empty. Time to refill!")
else:
print("Drinking the coffee...")
self.current_volume = 0
# Simulate Raj pouring coffee
def simulate_raj_pouring():
rajs_mug = CoffeeMug()
print("Raj starts pouring milk into his coffee...")
while rajs_mug.current_volume
js_mug.capacity:
time.sleep(0.5) # simulate
e time taken to pour
pour_amount = random.randint(20, 60)
Ask anything
Search
ChatGPT can make mistakes. Check important info.

Prototype Evaluation

Evaluation Methodology

- **Task-Based Usability Testing**
  - Each key prototype feature was tested via realistic user scenarios.
  - Each task tested one feature of our new ChatGPT interface.

- **Think aloud protocol**
  - Participants were encouraged to verbalize thoughts while performing each task.
  - Specific think-aloud prompts were given during interaction to uncover expectations, decision-making, and confusion points.

- **Participants**
  - We tested the prototype with 5 users (N=5)
  - We selected them from our earlier needfinding interviews
  - They are all moderate ChatGPT users

Prototype Evaluation contd.

Test Setup

- Each session lasted about ~15 minutes
- Roles: One person acted as a facilitator(guided the session), the other as a silent observer(noted behavior)
- All sessions were audio recorded (with participant consent) in controlled environment

What we collected

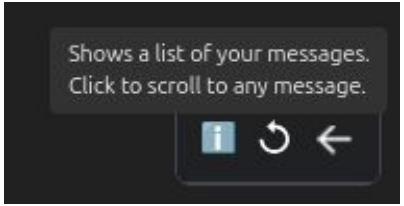
- **Post-Task Questionnaires**
  - After each task: Participants answered 3–4 short questions to check how easy and useful the feature was

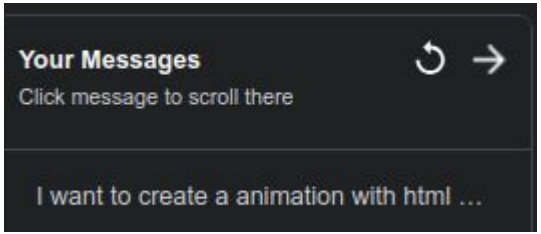
- **Final Evaluation: SUS Survey**
  - At the end: They filled out a System Usability Scale (SUS) survey to rate the overall usability of the prototype

Prototype Evaluation contd.

Task 1 – Navigation Sidebar

- **Scenario** : Locate a past message about "renewable energy" in a long chat of 15 questions
- **Observed Behavior**
  - Initially some tried scrolling manually or Ctrl+F.
  - Most users visited the your messages icon.
- **Usability Issues Identified**
  - Lack of textual signifier makes the icon less discoverable
- **Post-Test Feedback and Suggestions**
  - Icon helpful but could be more intuitive.
  - Add labels/ tool tip to improve clarity.

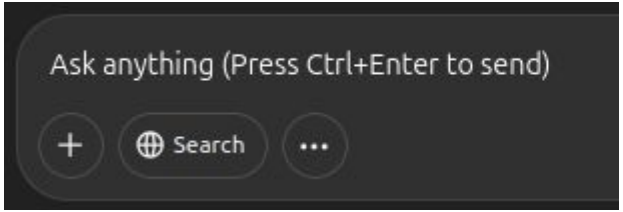




Prototype Evaluation contd.

Task 2 – Message Input Behavior

- **Scenario** : Type two separate questions for ChatGPT in one message. Make sure the second question starts on a new line (not in the same paragraph), and then send the message.
- **Observed Behavior**
  - Users expected Enter to send.
  - Once explained, users liked this feature.
  - Some confusion initially due to lack of prompt/tip.



- **Usability Issues Identified**
  - No visible cue on changed send message behaviour.
- **Post-Test Feedback and Suggestions**
  - Add tooltip or place holder in the input box

Prototype Evaluation contd.

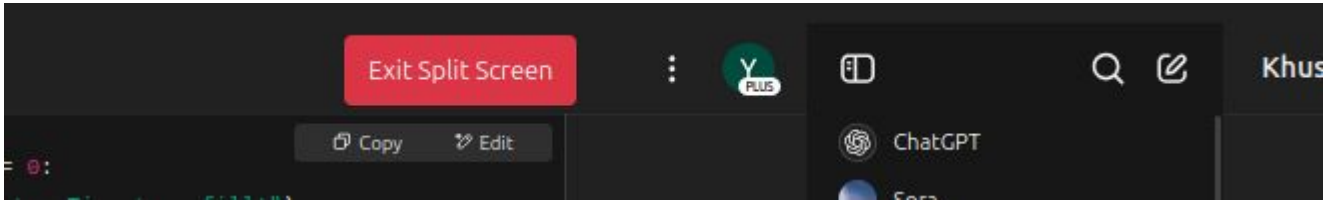
Task 3 – Export as PDF

- **Scenario** : Export chat and check PDF formatting.
- **Observed Behavior**
  - Feature easily located
  - Export worked well with clear formatting
- **Usability Issues Identified**
  - No issues reported
- **Post-Test Feedback and Suggestions**
  - PDF output was appreciated

Prototype Evaluation contd.

Task 4 – Split-Screen Mode

- **Scenario** : Compare answers from two different chats
- **Observed Behavior**
  - Most users were able to locate and “activate split-screen” from the sidebar.
  - “Exit Split” button was also easily located
- **Usability Issues Identified**
  - Exit button placement at the top-center blocked important parts of the chat
- **Post-Test Feedback and Suggestions**
  - Users liked the ability to compare chats without switching tabs
  - Reposition the Exit Split button to other place



Prototype Evaluation contd.

Task 5 – Sentiment Highlighting

- **Scenario** : Analyze tone of ChatGPT's response
- **Observed Behavior**
  - Users discovered and understood the sentiment button easily.
  - Responses were visually distinct and appreciated by some
- **Usability Issues Identified**
  - No critical usability issues reported
- **Post-Test Feedback and Suggestions**
  - Responses were visually distinct and appreciated by some.
  - Some users found it helpful, others felt it was serving aesthetic purpose only

Prototype Evaluation contd.

Task 6 – Free Exploration

- **Scenario** : Open-ended usage of the enhanced interface
- **Observed Behavior**
  - Users explored new sidebar, sentiment feature, and message sending behavior.
  - Some users tested “your messages” sidebar updates on their new message/chat or not
- **Post-Test Feedback and Suggestions**
  - Overall smooth experience.
  - Users favored
    - Export as PDF
    - Navigation sidebar
    - Split screen Mode
  - Suggested improvement / not that much needed
    - Message input behaviour
    - Sentiment Highlights

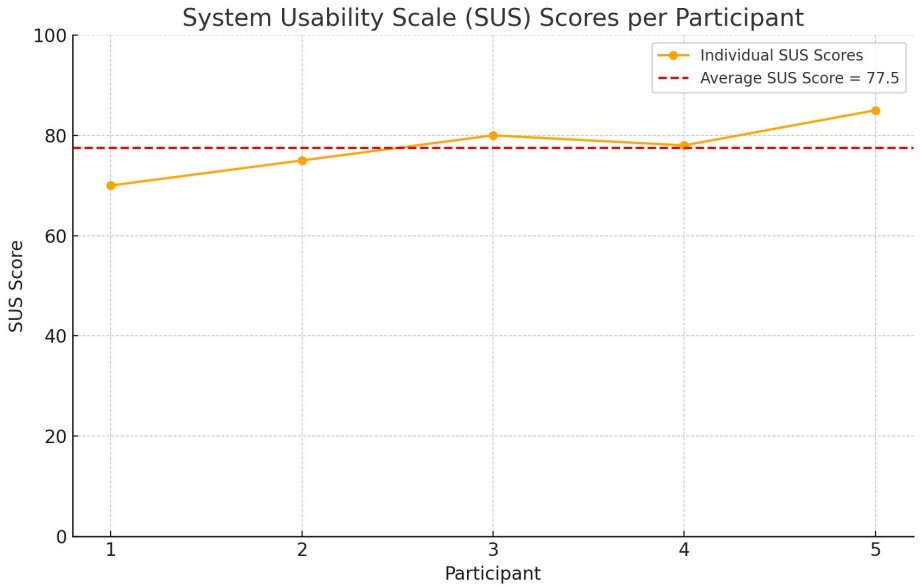
Prototype Evaluation contd.

Summary of Usability Issues Found

Feature	Usability Issue	Suggested Fix
Navigation Sidebar	Lacks identification label	Add label and tooltip to explain usage
Message Input	Confusing send/newline behavior	Modified placeholder in input box
Split-Screen View	Exit button covers some content	Reposition the Exit button
Export to PDF	No issue	—
Sentiment Highlight	No issue	—

Prototype Evaluation contd.

System Usability Scale (SUS) score



- The average SUS score was 77.5, which falls in the "Good" usability range
- Scores ranged from **70 to 85** across participants

What We Could have done differently

Future Improvements

- **Search bar in right navigation sidebar**
  - User struggle a bit in visually scanning the list of prompts (too many)
  - A search function would let user to instantly jump to specific question he asked
- **Resizable Split-Screen Panels**
  - Allowing user to drag and resize split screen panels improving comfort and readability
- **Allow Export of Selected Messages**
  - Users might only want to save a portion of the conversation (Not Just Full Chat)
- **Converting the Scripts to extension**
  - Not everyone is comfortable with the way user scripts are installed

All Project Resources

- Needfinding (Survey) google form circulated
  Needfinding Google form
- Needfinding (Survey) results
  Needfinding Survey Results
- Needfing (Interview) recordings
  Needfing Interview Recordings
- Usability Testing Interviews (Task-Based Evaluation)
  Usability Testing Interview
- Main Demo Video
  All Feature in Action
- Github Link of the userscripts:
  GitHub Repository – Userscript Source Code
- Additional Programming details for all the implementations can be found here:
  Programming Documentation – Implementation Details

Individual Contribution Across Milestones

Milestone	Senthil Ganesh P	Divyanshu	Khushwant Kaswan
Milestone 1 (Needfinding)	Designed and deployed the user survey (Google Form); planned survey analysis and question structure and documentation	Conducted semi-structured user interviews and noted qualitative feedback and documentation	Assisted in defining research goals and supported planning of survey and interviews and documentation.
Milestone 2 (Research Analysis & Design)	Analyzed quantitative survey data, helped refine the scope, and documented insights	Synthesized interview insights into actionable feature ideas leading to Triangulation and documented insights	Sketched initial wireframes,UI flows and evaluated feasibility of features based on insights of needfinding.
Milestone 3 (Prototype + Evaluation)	Contributed to testing feedback and played the role of silent observer.	Led usability testing (task design, sessions, SUS collection)	Implemented high-fidelity prototype (coded Tampermonkey userscript); built all major features; fixed bugs and implemented usability feedback and documentation


Thank you!