ChatGPT UI/UX ENHANCEMENT – MILESTONE 3 REPORT

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

Introduction: Project Goals & User Needs Summary

The **goal of this HCI project** is to enhance the user interface and experience of the ChatGPT conversational AI, addressing real pain points identified through user research. In **Milestone 1**, we proposed taking the widely-used ChatGPT system and improving its UI/UX by identifying gaps via needfinding (surveys and interviews) and then prototyping solutions. We targeted existing ChatGPT users and aimed to resolve common frustrations without altering the core AI functionality. Key **user needs** uncovered (primarily from a survey of ~30 users, with interviews to validate findings) included:

1. **Better navigation in long chats:** Users struggle to scroll through lengthy conversations to find past answers. *81%* of survey respondents rated a quick navigation sidebar as *important* or *Very Important*, indicating a strong need for jump-to-message functionality.
2. **Prevention of accidental message sends:** The Enter-to-send mechanism caused frequent mistakes. *63%* of users reported accidentally sending messages “Frequently” or “Very Frequently,” expressing frustration when trying to add a newline. Users requested a clearer way to add line breaks or a confirmation before sending.
3. **Ability to save or export conversations:** ChatGPT lacks an easy way to save chats. An overwhelming *88%* considered an “export to PDF/Word” feature *Valuable* or *Very Valuable*, showing that users want to keep important dialogues for reference or sharing.
4. **Support for multi-tasking / multi-chat:** Many users wish to handle multiple chats or view information side-by-side. About *68% recommended* a split-screen interface for simultaneous chats, to compare answers or refer to one conversation while working in another.
5. **UI personalization & visual clarity:** Some users suggested interface customizations (themes, fonts) and richer response formatting (e.g. colored text for sentiment, emojis) to improve clarity. While these were “nice-to-have” ideas (61% favored theme options), they were lower priority than the functional issues above.

These findings guided our design. In summary, the **primary needs** we chose to address are:

1) efficient navigation of long chat history,
2) prevention of unintended actions (message sends),
3) saving/sharing chat content,
4) multi-chat multitasking. Secondary considerations (visual enhancements, personalization) were noted but given lower priority due to time constraints and the survey indication that core functionality issues were more pressing.

We had given a detailed analysis in Milestone 2 regarding the survey that was undertaken. However, we did not mention the interview. The details of the interview are given in the next paragraph.

Follow-up Interviews Based on Survey Responses

After analyzing the survey results, we realized there were a few areas that needed more context — especially around how users actually experience issues like multitasking, sharing message, navigating longer chats etc. To dig deeper, we conducted short interviews of around 10 minutes with a few people from the original survey group.

The interviews helped us better understand what users really cared about. Some of the most common things that came up were the need to view two chats side by side, the issue of accidentally sending a message before it's ready, and how useful it would be to quickly jump back to a specific part of a long conversation. A few users also mentioned wanting a way to quickly get a sense of the tone or sentiment of a response, and an easier way to save or share important chats

We've shared the interview recordings (with consent) at the link below : **Interview Recordings**

Needfinding Insights: (From survey of 159 users and follow-up interviews)

Need	% Affected	Priority
Difficult navigation in long chats	81%	High
Accidental message send	63%	High
Export chat to PDF	88%	High
Split-screen/multitasking	68%	High
Sentiment clarity/readability	High (qualitative feedback)	Medium-High
Theme customization	Only ~6% dissatisfied	Low

Prioritized Features & Design Decisions

Based on the needfinding insights, we **prioritized a set of features** that directly address the above user needs. We focused on feasibility and impact, considering **constraints** (development time, technical complexity in modifying ChatGPT's web UI via a browser extension) and **trade-offs** (choosing simpler implementations that still solve the core problem). The final design decisions and features implemented are:

1. **Quick Navigation Sidebar:** A collapsible "Your Messages" sidebar that lists all user prompts in a chat, allowing one-click jump to any point in a long conversation. This addresses the navigation pain point (no more endless scrolling). (Need addressed: long chat navigation).

2. **Improved Message Input (Prevent Accidental Send):** We redesigned the message input behavior so that *Enter* keypress adds a newline instead of sending the message, and a deliberate *Ctrl+Enter* (or Cmd+Enter on Mac) is required to send. (Need addressed: error prevention in message composition).

3. **Export Chat to PDF:** A one-click "Export as PDF" button was added to the main UI. This generates a nicely formatted transcript of the conversation, including text, code blocks, and images, which the user can save. (Need addressed: saving/sharing conversations).

4. **Split-Screen Multi-Chat Mode:** An "Activate Split Screen" option that opens two side-by-side ChatGPT panels, allowing the user to view or work in two chats at once. This directly enables multitasking and comparing answers. We included a prominent "Exit Split Screen" button for clarity. (Need addressed: multi-tasking with multiple chats).

5. **Sentiment Visualizer (Colored Text Responses):** As a proof-of-concept for enhanced visual feedback, we added an *"Analyze Sentiment"* toggle button on each AI response. When pressed, the AI's reply text is highlighted green for positive tone or red for negative tone (using a simple sentiment analysis word list). Pressing it again resets the text to normal. (Need addressed: making information easier to parse at a glance).

Features Deferred/Not Implemented: Some requested improvements were left out due to scope. Notably, **UI theme personalization** (custom colors or layouts) was not pursued in this phase because survey feedback indicated current themes were mostly satisfactory and we prioritized functional fixes. Likewise, ideas like voice input or richer editing of messages were considered out of scope for our timeline. We focused on the five features above as they promised the highest impact on usability given our resources.

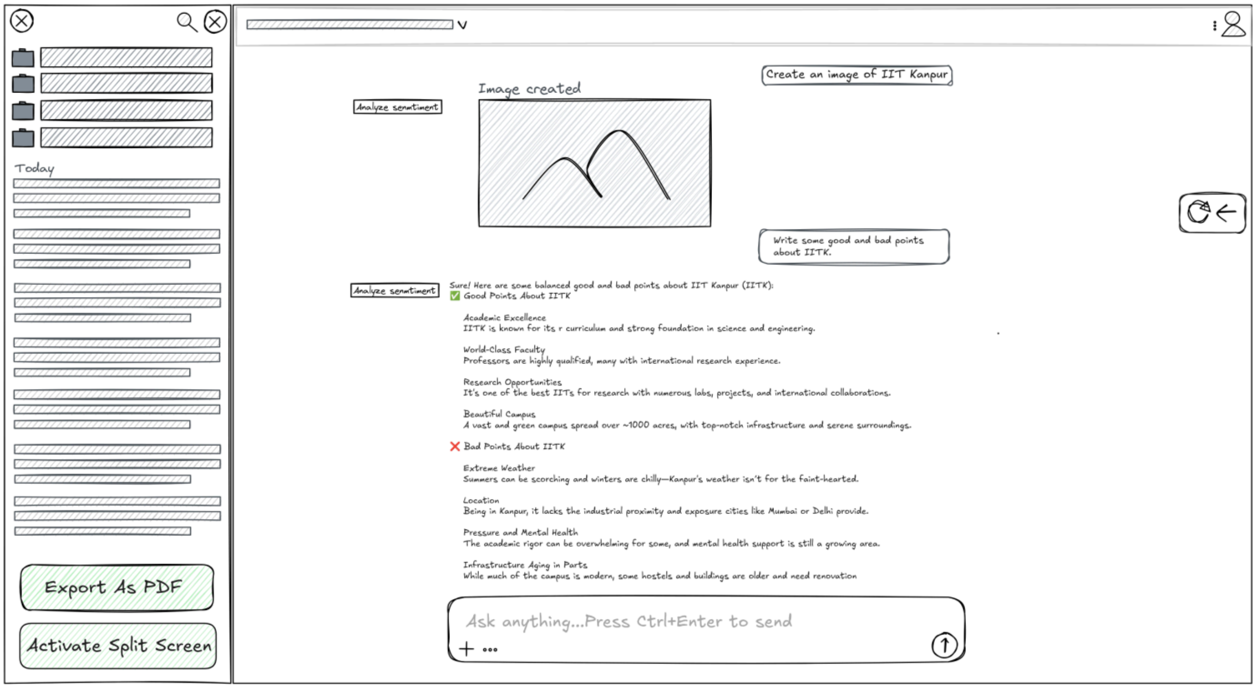
From User Needs to Prototype Features

Each **user need was translated into specific prototype features or UI interactions** as follows:

1. *Long chat navigation need* → **Quick Navigation Sidebar**
2. *Accidental send need* → **Revised input behavior**
3. *Saving conversations need* → **Export to PDF**
4. *Multi-tasking need* → **Split-Screen mode**
5. *Desire for visual clarity* → **Sentiment highlighting**

In the next section, we present the prototype's implemented features, with implemented solution's screenshots illustrating how these needs manifest as design solutions in our interface.

We created initial mid-fidelity prototypes for our features:



Create an image of IIT Kanpur

Image created

Write some good and bad points about IITK.

Sure! Here are some balanced good and bad points about IIT Kanpur (IITK):
✅ Good Points About IITK

Academic Excellence
IITK is known for its r curriculum and strong foundation in science and engineering.

World-Class Faculty
Professors are highly qualified, many with international research experience.

Research Opportunities
It's one of the best IITs for research with numerous labs, projects, and international collaborations.

Beautiful Campus
A vast and green campus spread over ~1000 acres, with top-notch infrastructure and serene surroundings.

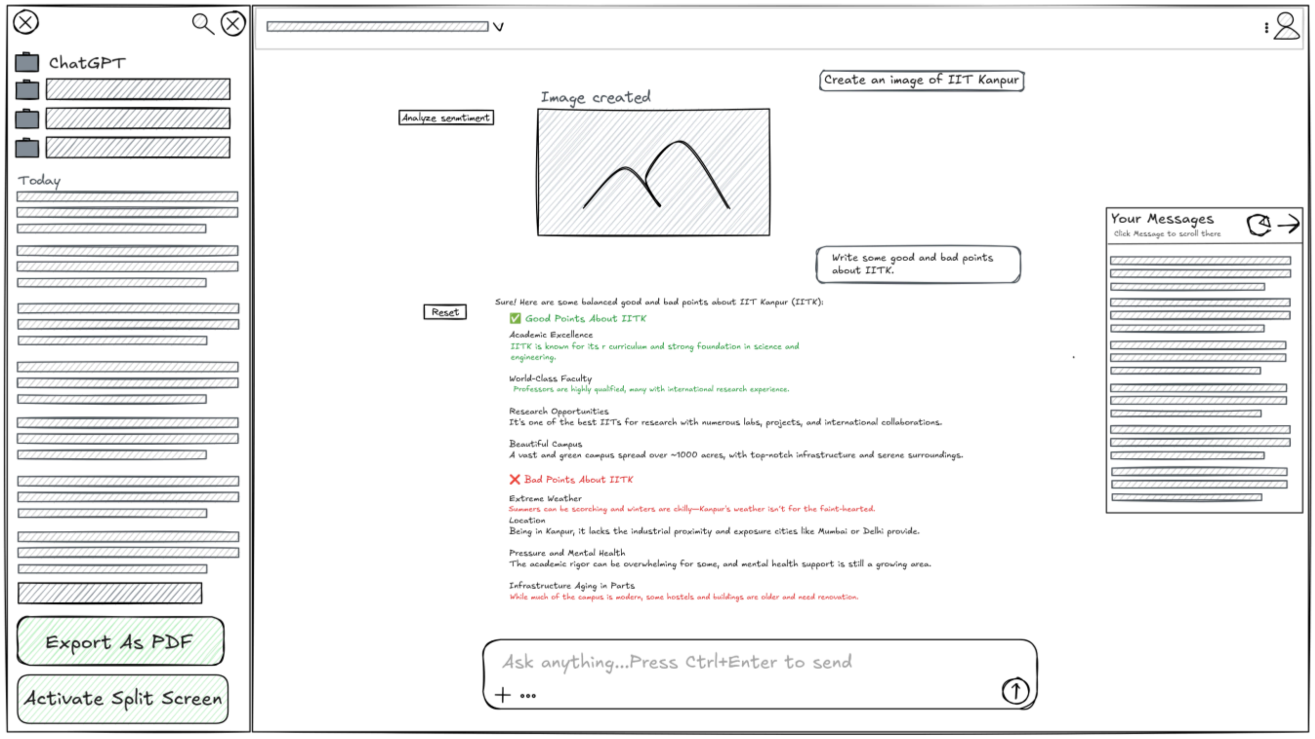
❌ Bad Points About IITK

Extreme Weather
Summers can be scorching and winters are chilly—Kanpur's weather isn't for the faint-hearted.

Location
Being in Kanpur, it lacks the industrial proximity and exposure cities like Mumbai or Delhi provide.

Pressure and Mental Health
The academic rigor can be overwhelming for some, and mental health support is still a growing area.

Infrastructure Aging in Parts
While much of the campus is modern, some hostels and buildings are older and need renovation



Create an image of IIT Kanpur

Write some good and bad points about IITK.

Sure! Here are some balanced good and bad points about IIT Kanpur (IITK):

✅ Good Points About IITK

Academic Excellence
IITK is known for its r curriculum and strong foundation in science and engineering.

World-Class Faculty
Professors are highly qualified, many with international research experience.

Research Opportunities
It's one of the best IITs for research with numerous labs, projects, and international collaborations.

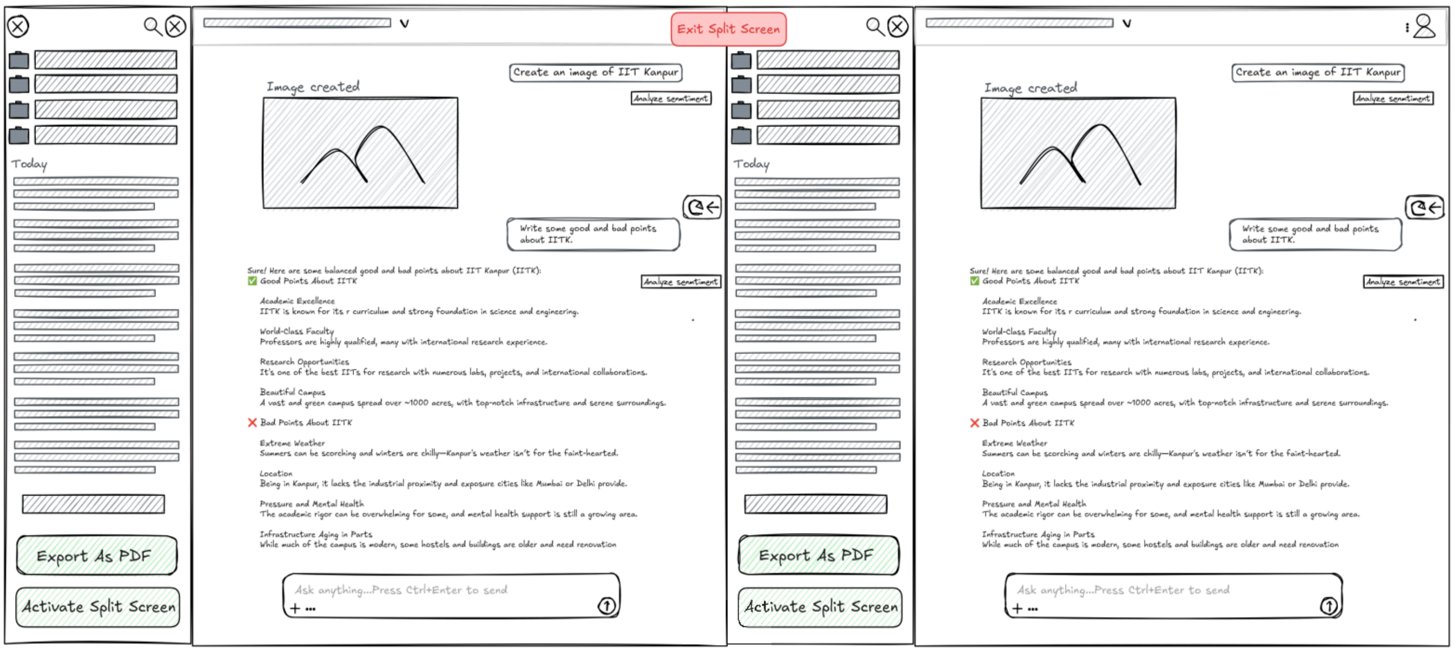
Beautiful Campus
A vast and green campus spread over ~1000 acres, with top-notch infrastructure and serene surroundings.

❌ Bad Points About IITK

Extreme Weather
Summers can be scorching and winters are chilly—Kanpur's weather isn't for the faint-hearted.
Location
Being in Kanpur, it lacks the industrial proximity and exposure cities like Mumbai or Delhi provide.

Pressure and Mental Health
The academic rigor can be overwhelming for some, and mental health support is still a growing area.

Infrastructure Aging in Parts
While much of the campus is modern, some hostels and buildings are older and need renovation.



Create an image of IIT Kanpur

Write some good and bad points about IITK.

Sure! Here are some balanced good and bad points about IIT Kanpur (IITK):
✅ Good Points About IITK

Academic Excellence
IITK is known for its r curriculum and strong foundation in science and engineering.

World-Class Faculty
Professors are highly qualified, many with international research experience.

Research Opportunities
It's one of the best IITs for research with numerous labs, projects, and international collaborations.

Beautiful Campus
A vast and green campus spread over ~1000 acres, with top-notch infrastructure and serene surroundings.

❌ Bad Points About IITK

Extreme Weather
Summers can be scorching and winters are chilly—Kanpur's weather isn't for the faint-hearted.

Location
Being in Kanpur, it lacks the industrial proximity and exposure cities like Mumbai or Delhi provide.

Pressure and Mental Health
The academic rigor can be overwhelming for some, and mental health support is still a growing area.

Infrastructure Aging in Parts
While much of the campus is modern, some hostels and buildings are older and need renovation

Create an image of IIT Kanpur

Write some good and bad points about IITK.

Sure! Here are some balanced good and bad points about IIT Kanpur (IITK):
✅ Good Points About IITK

Academic Excellence
IITK is known for its r curriculum and strong foundation in science and engineering.

World-Class Faculty
Professors are highly qualified, many with international research experience.

Research Opportunities
It's one of the best IITs for research with numerous labs, projects, and international collaborations.

Beautiful Campus
A vast and green campus spread over ~1000 acres, with top-notch infrastructure and serene surroundings.

❌ Bad Points About IITK

Extreme Weather
Summers can be scorching and winters are chilly—Kanpur's weather isn't for the faint-hearted.

Location
Being in Kanpur, it lacks the industrial proximity and exposure cities like Mumbai or Delhi provide.

Pressure and Mental Health
The academic rigor can be overwhelming for some, and mental health support is still a growing area.

Infrastructure Aging in Parts
While much of the campus is modern, some hostels and buildings are older and need renovation

Prototype Implementation and user interface overview

We built a high-fidelity **interactive prototype** as a set of front-end modifications to ChatGPT's web interface (implemented via a Tampermonkey userscript for ease of integration and testing). All features are functional in a web browser, meaning our prototype is not just a static design but a working system overlay on ChatGPT. Below we describe each major feature, how it works, and provide UI **screenshots** and **user flow** examples. The design emphasizes maintaining ChatGPT's clean look while augmenting it with new controls. *In all figures, green buttons and new panels represent our additions.*

Here is a demo video of the features implemented :

Main Demo Video – All Features in Action

Note : Demo videos of individual features are also attached with the feature's description.

Github Link of the userscripts:

GitHub Repository – Userscript Source Code

Additional Programming details for all the implementations can be found here:

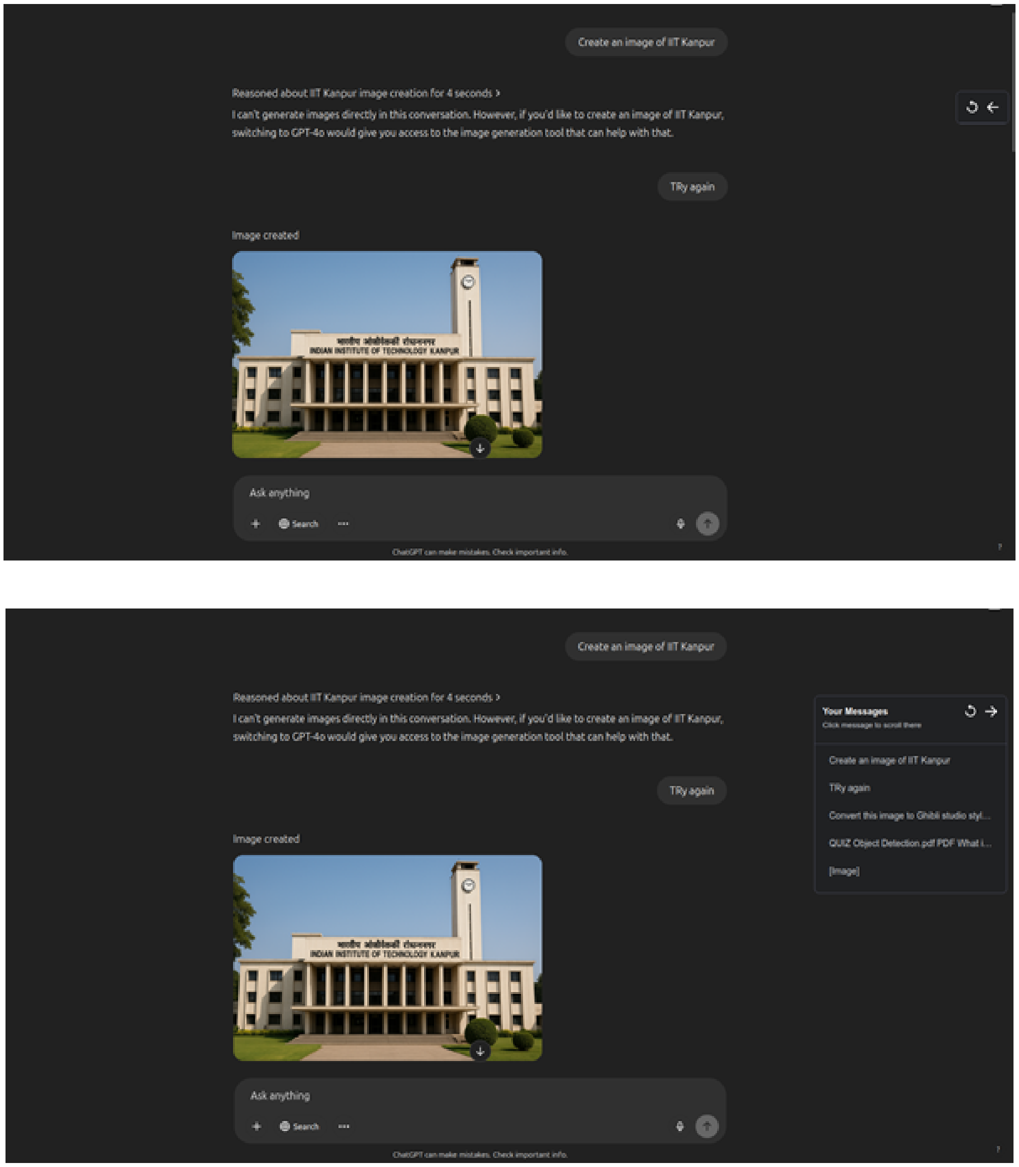
Programming Documentation – Implementation Details

Quick Navigation Sidebar (Chat Outline)

To help users navigate long chats, we implemented a **collapsible sidebar** on the right side of the chat that lists the **titles of all user prompts** in the conversation. When the sidebar is collapsed, only a subtle arrow icon is shown; when expanded, it displays a scrollable list of the user's messages (each entry is the first few words of a user prompt). The user can click any entry to instantly scroll the chat to that point.

The sidebar improves **information retrievability** by letting users jump to specific parts of a chat without endless scrolling. 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 scrollintoView behavior, minimalistic design)



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

भारतीय प्रौद्योगिकी संस्थान
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
Ask anything
Search
ChatGPT can make mistakes. Check important info.
Your Messages
Click message to scroll there
Create an image of IIT Kanpur
TRy again
Convert this image to Ghibli studio styl...
QUIZ Object Detection.pdf PDF What i...
[Image]

Figure 1: The Quick Navigation sidebar expanded on the right (titled "Your Messages"). Each user query in the chat is listed, enabling one-click navigation. In this example, the user's prompts like "Create an image of IIT Kanpur" and "Convert this image to Ghibli style…" are listed; clicking one scrolls directly to that part of the chat. A refresh icon (top right of panel) lets users re-index messages if needed, and the arrow icon collapses the sidebar.

No external dependencies; runs fully on the client-side and no performance degradation observed in chats.

Video Demo: Quick Navigation Sidebar

Export Conversation to PDF

To satisfy the widespread demand for saving chats, our prototype includes an **Export as PDF** feature. We added a prominently visible **“Export as PDF” button** to the left sidebar (below the chat history list), which stays available at all times in the interface. With one click, the entire chat history is compiled into a clean PDF document. The export respects the formatting of the conversation: user messages and AI responses are clearly distinguished, code blocks are properly formatted, and any images in the conversation

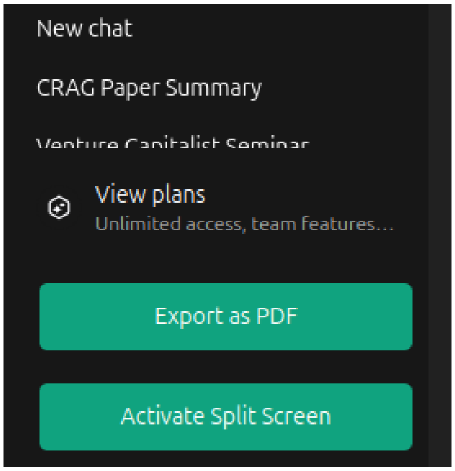


*Figure 2: New buttons added to the main sidebar. The **Export as PDF** button (top) allows the user to save the current chat. The **Activate Split Screen** button (bottom) toggles dual-chat mode. These buttons integrate into ChatGPT’s existing left menu for easy access.*

Video Demo: Export Chat to PDF

Full conversation structure (sender name, message blocks) is preserved with formatted HTML preview for clean printing or PDF save.

All done client-side — no external API used, safe and private.

Improved Message Input (Multi-line & Controlled Send)

Another critical improvement is the **message input box behavior**. We modified the interaction so that pressing **Enter or Shift+Enter** inserts a newline (line break) in the text, instead of sending the message. To send a message, the user can press **Ctrl+Enter** (Windows/Linux) or **Cmd+Enter** (Mac), or simply click the **Send** button as before.

The prior behavior of sending on a plain Enter led to anxiety and errors, especially during complex conversations.
The script does not modify core UI elements — only listens and acts on key events.

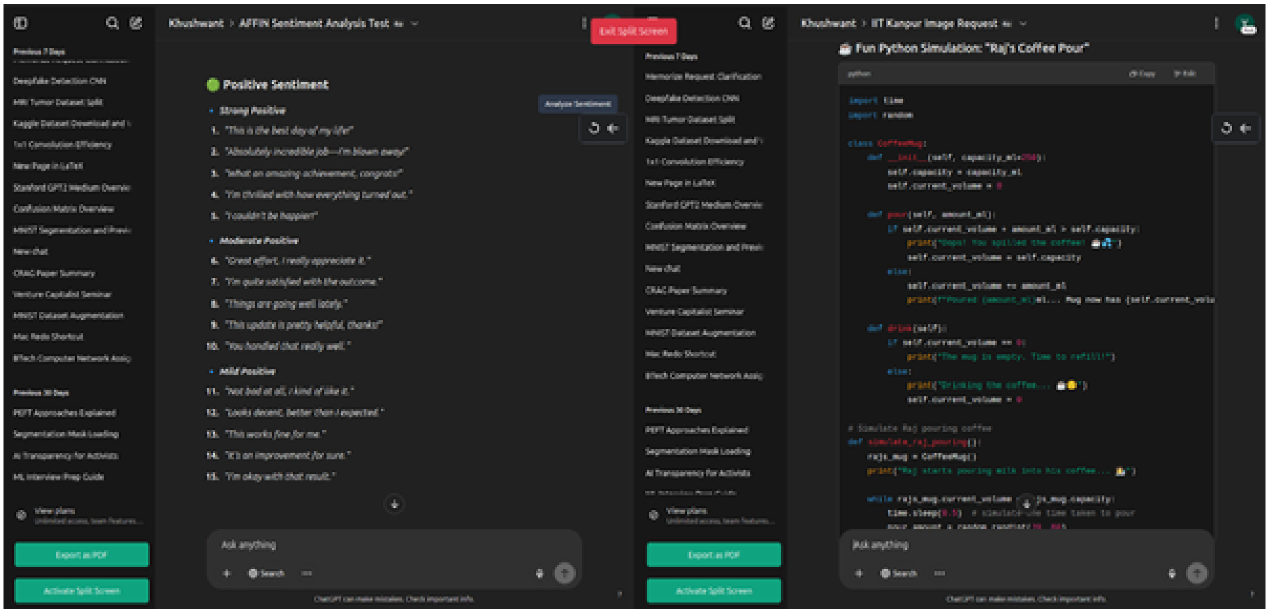
Split-Screen Dual Chat Mode

Perhaps the most technically ambitious feature of our prototype is the **Split-Screen mode**, which enables users to engage with two chats side by side. This is triggered by the **“Activate Split Screen”** button (visible in Figure 2, below the Export button). When activated, the interface enters a special layout: the screen splits into two panels, each showing a ChatGPT conversation interface. Two iframes of ChatGPT are overlaid on the main screen side-by-side. The user can have one chat on the left and another on the right, effectively doubling their ability to view or work with content at once and can cleanly restore to normal view, always accessible at top-center of dual screen layout as we planned in our initial prototypes.

This directly addresses the need for multitasking that many users expressed.

This is designed as a **non-intrusive, reversible** overlay that doesn't interfere with normal usage. Key UX principles observed:

- **Visibility & discoverability**: Activate button always in sidebar. Exit button available at top centre.
- **Control & freedom**: Activate and exit anytime. Does not reload or alter original tab — overlays everything for zero disruption
- **Consistency**: Looks and behaves like ChatGPT native controls.



Khushwant > AFFIN Sentiment Analysis Test
Exit Split Screen
Analyze Sentiment
Previous 7 Days
Deepfake Detection CNN
MRI Tumor Dataset Split
Kaggle Dataset Download and I
1x1 Convolution Efficiency
New Page in LaTeX
Stanford GPT2 Medium Overvie
Confusion Matrix Overview
MNIST Segmentation and Previ
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
Export as PDF
Activate Split Screen
Ask anything
Search
ChatGPT can make mistakes. Check important info.
Khushwant > IIT Kanpur Image Request
Memorize Request Clarification
python
Copy
Edit

🟢 Positive 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."*

Fun Python Simulation: "Raj's Coffee Pour"

```python
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 < rajs_mug.capacity:
        time.sleep(0.5)  # simulate the time taken to pour
```

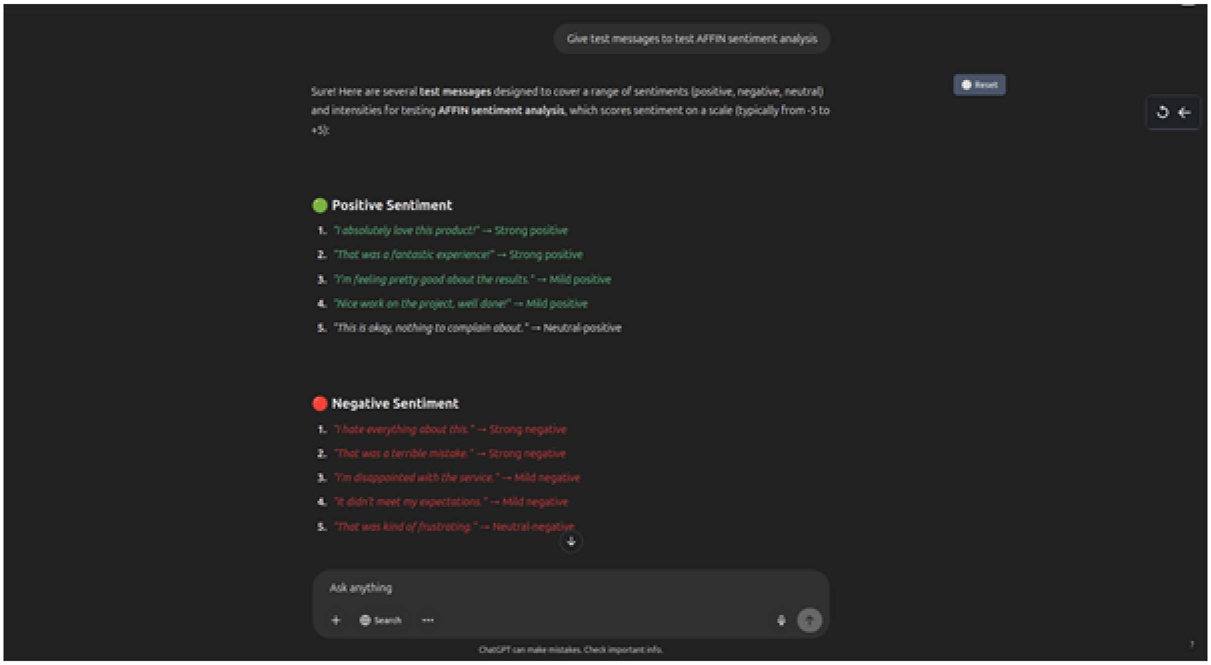
*Figure 3: **Split-Screen Mode** in action. The user has two chat sessions open side by side. The left panel shows a conversation ("AFFIN Sentiment Analysis Test"), and the right panel shows another ("IIT Kanpur Image Request") with a code output. Each panel is a fully functional ChatGPT interface (complete with its own scrollable chat history and input box). A red **"Exit Split Screen"** button at the top allows the user to return to the single-chat view.*

Video Demo: Split-Screen Chat Mode

Performance impact is minimal as iframe content loads in parallel.

Sentiment Analysis Highlight (Visual Tone Indicator)

As an **experimental feature** to improve the interpretability of AI responses, we added an *Analyze Sentiment* button that appears alongside each ChatGPT response. This small toggle button (labelled “Analyze” or “Analyze Sentiment”) allows the user to color-code the response text based on its sentiment polarity: positive phrases turn green and negative phrases turn red. It’s essentially a lightweight sentiment analysis overlay using a predefined lexicon (AFINN word list) to score the text. If the user clicks the button again (now shown as “Reset”), the colors are removed and the text returns to normal. This feature speaks to the user feedback about *adding visual elements for clarity* – for instance, quickly highlighting if the assistant’s answer contains more positive or negative language, which could be useful in scenarios like analyzing the tone of an essay, feedback, or just for curiosity.



Give test messages to test AFFIN sentiment analysis

Sure! Here are several **test messages** designed to cover a range of sentiments (positive, negative, neutral) and intensities for testing **AFFIN sentiment analysis**, which scores sentiment on a scale (typically from -5 to +5):

🟢 **Positive Sentiment**

1. *"I absolutely love this product!"* → Strong positive
2. *"That was a fantastic experience!"* → Strong positive
3. *"I'm feeling pretty good about the results."* → Mild positive
4. *"Nice work on the project, well done!"* → Mild positive
5. *"This is okay, nothing to complain about."* → Neutral-positive

🔴 **Negative Sentiment**

1. *"I hate everything about this."* → Strong negative
2. *"That was a terrible mistake."* → Strong negative
3. *"I'm disappointed with the service."* → Mild negative
4. *"It didn't meet my expectations."* → Mild negative
5. *"That was kind of frustrating."* → Neutral-negative

Figure 4: Sentiment Highlighting in use. The user asked for test messages to analyze sentiment, and upon clicking ***Analyze Sentiment****, the positive statements (above) turned green and the negative ones (below) turned red. A “Reset” button appears (upper right of the answer panel) to revert the text. This feature helps users quickly see emotional tone in ChatGPT’s output, addressing suggestions for colored text feedback.*

Video Demo: Sentiment Highlight

Fully client-side; no external API or network request involved. No impact on performance, as sentiment scoring is lightweight and only triggered on button press.

Project Progress vs Time

According to the original project timeline set in Milestone 1, our plan was: needfinding completed by Mar 21, prototype by Apr 4, and prototype evaluation by Apr 18. We are pleased to report that we are **on track with the timeline**. Key milestones achieved:

- **Needfinding (Completed by Mar 21):** We conducted the user survey in early March and collected responses well before the deadline. We also lined up a few user interviews. By Mar 21, we had analyzed the survey data (as presented in Milestone 2) and extracted the prioritized needs that directly informed our design. (Interviews were slightly delayed but a couple of them were done by end of March, largely confirming the survey findings – e.g., interviewees echoed frustration with accidental sends and desire for export capability.)
- **Prototype Design (Completed by Apr 4):** We began brainstorming solutions and sketching interfaces immediately after analyzing needs. By the beginning of April, we had low-fidelity sketches for each major feature and started technical prototyping. The core features – navigation sidebar, input tweak, export, and split view – were decided and initial code written. We had a medium-fidelity clickable demo around Apr 4, which we then polished.
- **Prototype Implementation (Apr 4 – Apr 18):** Over the two weeks leading up to Apr 18, we focused on building out the high-fidelity prototype with full functionality. As of this Milestone 3 report, **all major planned features have been implemented and integrated** in our prototype. We have tested them in our development environment (Firefox/Chrome with the userscript). A short video demo accompanies this report (and the screenshots above) to illustrate each feature in action. We also performed informal user testing within the team and with two external students to catch obvious bugs. Some minor issues were fixed (for instance, the PDF export initially had trouble with images, which we resolved; the sentiment analysis initially colored the entire message block which was overwhelming, so we switched to per-sentence coloring).
- **Pending Work (Apr 18 – Final Presentation):** We are entering the **evaluation phase**. According to the timeline, evaluation of the prototype is to be completed by Apr 18; realistically, we have designed the evaluation and will conduct it in the days following this report (given that the prototype was finished only very recently). We aim to complete user testing and gather results before the final presentation (finals week). Thereafter, we will still have a chance to iterate on the prototype based on feedback. Fortunately, since our implementation is modular, we anticipate being able to refine features (or disable/adjust any that cause issues) in response to user testing results.

In summary, by Milestone 3 we have a **fully functional first-version prototype** that embodies the design decisions grounded in user needs. We are on schedule and prepared to execute the evaluation and then present the final outcomes as per the course timeline

Evaluation Plan

With the prototype in hand, the next critical step is to **evaluate its usability and effectiveness**. Our evaluation plan is designed to answer specific questions about whether the new features indeed solve the user needs and how they impact the user experience. We will use a combination of qualitative and quantitative methods, focusing on usability testing with target users. Below we outline the evaluation in detail:

Methodology:

We will conduct a **usability study** in a controlled setting, using **scenario-based tasks** and **think-aloud** protocol. Participants will use the prototype to perform given tasks, while we observe and take notes. After tasks, we will collect subjective feedback via a short questionnaire and follow-up interview questions. Our approach combines quantitative measures (task success, time, error counts) with qualitative insights (comments, ratings).

1. **Study Design:** We plan a within-subject scenario: users will be given tasks to complete using our enhanced interface. If feasible, we might also have them try the same tasks on the original ChatGPT interface (for a comparative baseline) – though this depends on time and avoiding fatigue. Our primary focus is on our interface, but a comparison could highlight improvements (e.g., time saved with the sidebar vs. without). The study will be moderated (one of us will facilitate, and another will observe/record). We'll encourage participants to "think aloud" – i.e. verbalize what they are trying to do, if something is confusing, etc., to gather insight into their thought process.

2. **Tasks:** We will create concrete tasks that align with each key feature:
    a. **Find Information Task:** *"Here is a long chat (we'll preload a chat with ~20 Q&A). Please find the part where you asked about <specific topic> and tell us what answer ChatGPT gave."* – This task targets the navigation sidebar. We'll measure if they use the sidebar, how quickly they find the answer, and note any scrolling vs clicking.
    b. **Compose Message Task:** *"Ask ChatGPT to write a two-paragraph poem about spring. Make sure to go to a new line between the paragraphs in your message."* – This observes the input box behavior. We see if they try to press Enter for newline and note the outcome (with our feature, it should just work). If they press Ctrl+Enter accidentally, that might send early (we'll watch for any confusion).
    c. **Export Task:** *"You found the answer helpful and want to save this conversation. Go ahead and save it as a PDF."* – We watch how they use the Export feature: do they find the button quickly, do they manage to save the file. We'll ask them to open the PDF and check if it looks okay.
    d. **Multi-chat Task:** *"Now imagine you want to compare the answer from this chat with another chat where ChatGPT answered a related question. Here's another*

chat link… Could you compare the two responses side by side?" – This will prompt them to use split-screen. We'll see if they figure out to click "Activate Split Screen" (we can give a hint if needed like "Is there a way to view both?"). Once in split mode, we'll observe how they navigate and if they can effectively do the comparison or multi-task.

e. **General Q&A Task:** We might include a free-form task like "Using this interface, ask any question you like or continue a conversation as you normally would." This is to see general comfort and if any feature interrupts the normal flow. It also gives them freedom to explore, possibly trying the sentiment button out of curiosity.

3. **Participants:** We aim to recruit **5-7 participants** who are current ChatGPT users from our survey and interview(at least moderately familiar with the original interface). Given our project scope, 5-7 users is a reasonable number to get diverse feedback while being manageable to schedule. If time permits, we might go up to 8-10 for more robustness, but our initial target is 5 as a baseline. We will ensure a mix of users (for example, some who use ChatGPT daily, some less frequently) to see how intuitive the features are for different expertise levels. All participants will likely be fellow students or colleagues, which is acceptable for this study since our target user group is essentially "people who use ChatGPT for various tasks" – our recruits fit that profile. We will of course brief them and obtain consent, making it clear this is a prototype test and their feedback is valuable to improve it.

4. **Procedure:** Each session will last about 30 minutes. We'll begin with a quick introduction, asking about their prior experience with ChatGPT (to gauge familiarity). We'll then have them perform the tasks outlined. We will encourage think-aloud, but not force it if it disrupts them. We (the moderators) will observe silently, only intervening if the participant is very stuck or if a hint is needed to proceed (and we'll note where such interventions were needed). We will record the screen (with permission) to later measure task times and review any issues. After tasks, we will administer a short **post-test questionnaire** – this will include a **SUS (System Usability Scale)** or a custom Likert-scale set of questions on ease of use, and specific questions like "Did you find the navigation sidebar useful?" (rating 1-5) and similar for other features. Finally, we'll do a brief interview asking what they liked most, least, and any suggestions.

Justification in HCI Literature:

Our evaluation approach is grounded in standard HCI methods:

1. We chose a **usability study** because we have a working prototype and we want to observe actual use and identify usability issues. Given our features aim to improve

efficiency (e.g., faster navigation), a task-based test is appropriate to verify those improvements.

2. The number of participants (5-7) follows the well-known heuristic that a handful of users can reveal the majority of usability problems. If time allowed, more participants could be added for greater confidence, but within our resources this is a sensible number.
3. Using **scenario tasks** ensures ecological validity – we simulate real usage scenarios (like “find info from a previous chat” which is a realistic user goal). This ties the evaluation to actual user needs and tasks identified earlier, a recommended practice in interaction design (tasks should reflect real-world use cases).
4. We incorporate both **qualitative (think-aloud, interviews)** and **quantitative (timings, success rates)** measures.
5. We will also be mindful of the **learning effect**. If we do have them use the original interface vs the new, we might counterbalance order or just rely on their existing familiarity as baseline. However, our main interest is absolute performance and satisfaction with the new interface, so a direct comparison may not be strictly required; users’ memory of the old interface may suffice for them to judge if this feels better.
6. The **System Usability Scale (SUS)** or similar post-test questionnaire provides a standardized metric for overall usability perceived. It’s a 10-item questionnaire widely used in HCI evaluations to produce a usability score. While we won’t obsess over the exact score given our small sample, it helps to benchmark whether our interface is in a generally “good” range of usability and can highlight if any statement (like “I found the system unnecessarily complex”) gets disagree/agree patterns that indicate issues.

Post-Evaluation and Next Steps:

Finally, we will prepare to present these findings, along with a live demo of the prototype, in the final presentation. The evaluation results will help us tell a story of how user-centered design can tangibly improve an existing product like ChatGPT, and how each step from needfinding to prototyping to testing is crucial in that process.