**A Project Report on**

**AI Fusion**

submitted in partial fulfillment for the award of

**Bachelor of Technology**

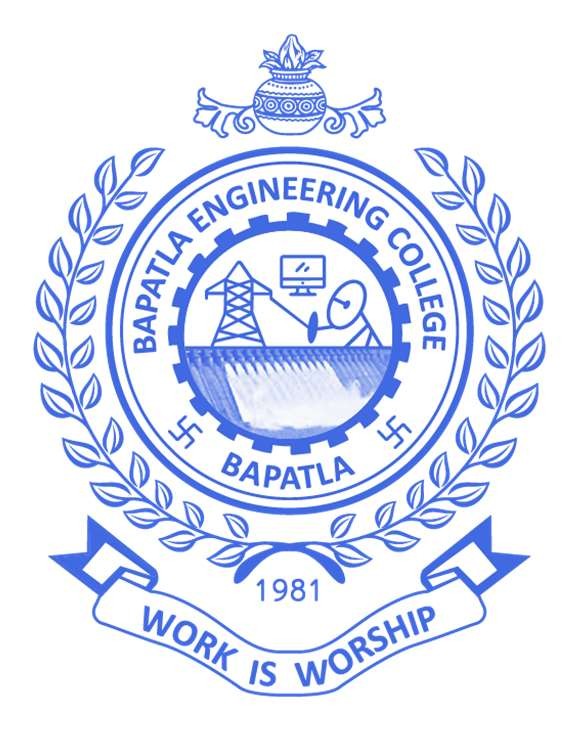
in

**Computer Science and Engineering**

by

M. Kavya(Y21ACS504) K.Nagaraju (Y21ACS487)

M. SaiKiaran (Y21ACS50478) K.V.N.B Srikanya(Y21ACS463)



Under the guidance of

Y.Saranya,M.Tech

Assist Professor.

Department of Computer Science and Engineering

**Bapatla Engineering College**

(Autonomous)

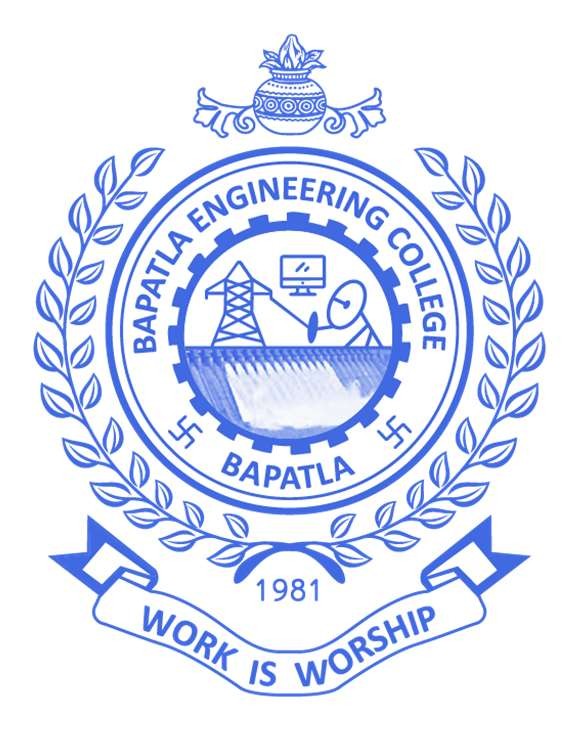
(Affiliated to Acharya Nagarjuna University)

**BAPATLA – 522 102, Andhra Pradesh, INDIA**

**2023-2024**

**Department of**

**Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the project report entitled **Artificial Intelleigence Fusion** that is being submitted by M.Kavya (Y21ACS504), K.Nagaraju (Y21ACS487), M.SaiKiran (Y21ACS478) and K.V.N.B Srikanya(Y21ACS463) in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering to the Acharya Nagarjuna University is a record of bonafide work carried out by them under our guidance and supervision.

Date:

**Signature of the Guide Signature of the HOD Y.Saranya Dr. M. Rajesh Babu**

**Assist Professor Assoc. Prof. & Head**

**DECLARATION**

We declare that this project work is composed by ourselves, that the work contained herein is our own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

M.Kavya (Y21ACS504)

K.Nagaraju(Y21ACS487)

K.Saikiran(Y21ACS478)

K.Srikanya(Y21ACS463)

**Acknowledgement**

We sincerely thank the following distinguished personalities who have given their advice and support for successful completion of the work.

We are deeply indebted to our most respected guide **Y.Saranya**, Assist Professor, Department of CSE, for her valuable and inspiring guidance, comments, suggestions and encouragement.

We extend our sincere thanks to **M. Rajesh Babu**, Assoc. Prof. & Head of the

Dept. for extending his cooperation and providing the required resources.

We would like to thank our beloved Principal Dr. **Dr.N.Rama Devi** for providing the online resources and other facilities to carry out this work.

We would like to express our sincere thanks to our project coordinator

**Dr**. **P.Pardhasaradhi**, Prof. Dept. of CSE for his helpful suggestions in presenting this document.

We extend our sincere thanks to all other teaching faculty and non-teaching staff of the department, who helped directly or indirectly for their cooperation and encouragement.

M.Kavya(Y21ACS504)

K.Nagaraju(Y21ACS487)

K.Saikiran(Y21ACS478)

K.Srikanya(Y21ACS463)

**Table of Contents**

[**List of Figures** viii](#_Toc195113213)

[**Abstract** ix](#_Toc195113214)

[1 **Introduction** 1](#_Toc195113215)

[1.1 **Background** 1](#_Toc195113217)

[1.2 **Problem Statement** 3](#_Toc195113218)

[1.3 **Motivation** 4](#_Toc195113219)

[1.3.1 **Fragmented AI Ecosystem**: 5](#_Toc195113220)

[1.3.2 **Lack of Multi-Model Accessibility**: 5](#_Toc195113222)

[1.3.3 **Absence of In-Browser Integration**: 5](#_Toc195113224)

[1.3.4 **Need for Quick Interaction Features**: 5](#_Toc195113225)

[1.3.5 **Demand for Combined Visual and Text AI Tools**: 6](#_Toc195113226)

[**1.4** **Objective** 6](#_Toc195113227)

[1.5 **Significance** 7](#_Toc195113228)

[1.6 **Existing System** 8](#_Toc195113229)

[**Google Gemini (Formerly Bard)** 8](#_Toc195113230)

[**Claude by Anthropic** 9](#_Toc195113231)

[2 **Literature Review** 10](#_Toc195113232)

[2.1 **Multi-Model AI Integration** 10](#_Toc195113233)

[2.2 **Browser-Based AI Tools** 10](#_Toc195113234)

[2.3 **AI in Communication and Automation** 11](#_Toc195113235)

[2.4 **AI-Driven Image Generation** 11](#_Toc195113236)

[2.5 **Summary and Research Gap** 11](#_Toc195113237)

[3 **Proposed System** 12](#_Toc195113238)

[3.1 **Task** 12](#_Toc195113239)

[3.1.1 **Project Planning and Setup:** 12](#_Toc195113240)

[3.1.2 **Feature Design and Integration** 13](#_Toc195113241)

[3.1.3 **Module Development and Logic Building:** 13](#_Toc195113242)

[3.1.4 **API Integration and Backend Logic :** 13](#_Toc195113243)

[3.1.5 **Training and Evaluation:** 14](#_Toc195113244)

[3.1.6 **Deployment and Packaging:** 14](#_Toc195113245)

[3.1.7 **Documentation and Reporting:** 14](#_Toc195113246)

[3.2 **Dataset** 15](#_Toc195113247)

[3.3 **Input** 15](#_Toc195113248)

[3.4 **Output** 16](#_Toc195113249)

[4 **Technologies and Frameworks Used** 17](#_Toc195113250)

[5 **System Design** 19](#_Toc195113251)

[5.1 **Use Case Diagram** 19](#_Toc195113253)

[5.2 **Class Diagram** 20](#_Toc195113254)

[5.3 **Activity Diagram** 21](#_Toc195113255)

[5.4 **Sequence Diagram** 22](#_Toc195113256)

[5.5 **Collaboration Diagram** 23](#_Toc195113257)

[5.6 **State Chart Diagram** 24](#_Toc195113258)

[6 **Implementation** 25](#_Toc195113259)

[6.1 **Requirements** 25](#_Toc195113260)

[6.1.1 **Libraries** 26](#_Toc195113261)

[**Networking & API Calls** 27](#_Toc195113262)

[6.2 **Code** 28](#_Toc195113266)

[6.2.1 **GitHub Link** 28](#_Toc195113267)

[6.2.2 **Importing all necessary packages** 28](#_Toc195113268)

[6.3 **Multi-Model AI Communication** 29](#_Toc195113269)

[6.4 **Email Summarization and Auto Reply Logic** 29](#_Toc195113270)

[6.5 **PDF and HTML Content Extraction** 30](#_Toc195113271)

[6.6 **AI Routing and Prompt Engineering** 30](#_Toc195113272)

[6.7 **Dataset Handling and ML Extension (Planned)** 30](#_Toc195113273)

[6.8 **Frontend Architecture** 31](#_Toc195113274)

[6.9 **Model Switching Logic** 32](#_Toc195113275)

[6.10 **Image Generation Module** 32](#_Toc195113276)

[6.11 **Email Auto-Reply Feature** 33](#_Toc195113277)

[7 **Results** 34](#_Toc195113278)

[7.1 **Interface** 34](#_Toc195113279)

[7.2 **Output when user make a request** 35](#_Toc195113280)

[7.3 **Output when user make request to generate image** 36](#_Toc195113281)

[7.4 **Output when user make request to summarize the mail** 37](#_Toc195113282)

[7.5 **Output when user make request to compose the mail** 37](#_Toc195113284)

[8 **Conclusions** 39](#_Toc195113285)

[9 **References** 40](#_Toc195113286)

# **List of Figures**

[Figure 1 - Use case diagram 19](#_Toc195807599)

[Figure 2 - class diagram 20](#_Toc195807600)

[Figure 3 - Activity Diagram 21](#_Toc195807601)

[Figure 4 - Sequence Diagram 22](#_Toc195807602)

[Figure 5 - collaborative diagram 23](#_Toc195807603)

[Figure 6 - start chart Diagram 24](#_Toc195807604)

[Figure 7 - user interface 35](#_Toc195807605)

[Figure 8 - Result when user make request 36](#_Toc195807606)

[Figure 9 - Result when user make request to generate image 36](#_Toc195807607)

[Figure 10 - Result when user make request to summarization of mail 37](#_Toc195807608)

[Figure 11 - Result when user make request to summarization of mail 37](#_Toc195807609)

[Figure 12 - Result of requested mail 38](#_Toc195807610)

# **Abstract**

AI Fusion is a Chrome extension designed to centralize access to leading AI models like OpenAI’s GPT, Google’s Gemini, and Anthropic’s Claude. Instead of switching between separate platforms, users can interact with these models through a unified mini-window interface directly in their browser. This allows for seamless multitasking and faster, AI-assisted productivity during browsing, content creation, and research.

Built using React.js for the frontend and Django for the backend, AI Fusion ensures responsive performance and secure API handling. It includes features like email summarization, quick auto-reply buttons, and letter generation tailored to different scenarios—all aimed at enhancing communication and reducing manual effort.

The extension also supports image generation from text prompts, expanding its use cases to visual content creation. By combining text and visual AI tools in one interface, AI Fusion simplifies workflows and showcases the practical benefits of multi-model, multimodal AI integration in everyday tasks.

Keywords:AI-Integration,Multi-Model,System,Email,Auto-Reply,Chrome Extension, Image Generation.

# **Introduction**

## AI Fusion is a Chrome extension that integrates multiple AI models such as OpenAI’s GPT, Google’s Gemini, and Anthropic’s Claude into a single user-friendly interface. It enables users to interact with these models seamlessly without switching tabs or platforms. Built with React.js on the frontend and Django on the backend, it ensures smooth and secure performance. Key features include smart replies, email summarization, letter generation, and image creation. This project enhances productivity by embedding AI tools directly into the user’s browsing workflow.

## **Background**

As AI becomes increasingly vital in daily digital tasks, users often need to rely on several platforms for different models and features. This fragmented access disrupts efficiency and limits real-time productivity. Existing tools rarely offer multi-model integration or in-browser flexibility. AI Fusion was developed to overcome these limitations by centralizing powerful AI tools into one extension. It reflects the need for faster, smarter, and more accessible AI interaction in modern web environments

With the rise of AI as a daily assistant in communication and creative tasks, there is a growing demand for tools that can provide versatile support without overwhelming the user with complexity. Existing AI platforms often operate in silos, requiring users to open multiple tabs or applications to utilize different models. This fragmentation can be especially counterproductive in scenarios where quick switching or comparisons between model outputs is necessary. AI Fusion solves this challenge by embedding AI interaction directly into the browser via a Chrome extension, offering an uninterrupted, integrated experience.

Beyond text processing, users increasingly seek visual creativity powered by AI. AI Fusion incorporates image generation functionality, enabling users to convert descriptive prompts into visual content on the fly. Whether drafting professional emails, responding to messages with quick replies, summarizing conversations, or generating imagery, AI Fusion delivers a suite of smart tools through an intuitive interface. Built using React.js and Django, it ensures performance efficiency and modular expansion, setting a foundation for scalable, AI-powered browser experiences

This project builds upon the idea that the future of AI usability lies not in isolated tools but in unified platforms that offer flexible, real-time, multi-model interactions. AI Fusion is designed with adaptability in mind, opening up possibilities for integrating even more advanced models and multimodal capabilities in future iterations. By responding to both user needs and the evolving AI landscape, AI Fusion represents a step toward more intelligent and context-aware productivity environments.

This hybrid approach addresses several limitations of traditional Android malware detection methods. It can effectively handle high-dimensional feature spaces, capture complex interactions between permissions, and adapt to dynamic changes in malware behaviour. By leveraging the rich metadata provided by Android's permission system and the optimization capabilities of genetic algorithms, researchers can develop robust and adaptive detection models capable of identifying both known and previously unseen malware variants.

In summary, the growing reliance on AI tools for communication, creativity, and productivity highlights the need for more integrated and user-friendly solutions. While individual models like GPT, Gemini, and Claude offer powerful capabilities, their separation across platforms limits accessibility and efficiency. Existing tools lack the flexibility to support multi-model interactions, quick replies, and visual generation within a single interface. These limitations set the stage for AI Fusion—a unified Chrome extension that bridges these gaps by delivering a seamless, real-time AI experience directly within the user's browser. This project builds upon current trends while introducing a new direction for smart, consolidated AI usage in everyday digital environments.

## **Problem Statement**

As artificial intelligence becomes increasingly integrated into day-to-day activities, users are depending more on large language models (LLMs) for tasks such as writing assistance, email replies, summarization, and creative generation. However, most existing platforms are siloed, allowing access to only a single AI model at a time. This limits users' ability to compare results, choose the most contextually accurate response, or utilize model-specific strengths. The lack of a unified interface for interacting with multiple AI systems hinders productivity and creates friction in the user experience.

Furthermore, the typical workflow for accessing AI tools involves switching between different tabs, applications, or websites, each tailored to one specific model. This not only consumes time but also breaks user focus and introduces unnecessary complexity. Users often need to copy content across different interfaces, manage logins or APIs separately, and manually organize outputs from different models. Such inefficiencies create barriers for casual users and professionals alike who seek quick and reliable AI support integrated into their existing workflow.

In addition to model switching, users frequently engage in tasks that require a combination of capabilities—such as drafting an email, generating a summary, and creating accompanying visuals. Most current tools are not designed to support multimodal interactions (text + image) within a single interface, requiring users to resort to entirely different platforms for image generation or visual design. This separation further fragments the creative process and limits the practical usage of AI in real-time scenarios.

AI Fusion addresses these gaps by introducing an all-in-one Chrome extension that integrates multiple AI models along with support for image generation. The extension allows users to interact with their chosen model, receive contextual responses, and generate visual content directly within the browser. This unified platform minimizes disruption, simplifies access, and streamlines complex workflows. However, building such an integrated system presents challenges in terms of architecture, API management, and model switching—all of which are considered and tackled in the design of AI Fusion

## **Motivation**

Users often struggle with switching between multiple AI platforms to access different models, which disrupts workflow and reduces efficiency. There’s a growing need for a unified tool that combines various AI capabilities in one place. Current systems lack integration of both text-based and visual AI features. AI Fusion aims to streamline this experience through a compact browser extension. It provides seamless access to multiple models and tools, enhancing productivity and creativity.

### **Fragmented AI Ecosystem**:

### In today’s digital landscape, users rely on multiple AI platforms for writing, coding, designing, and communication. This fragmentation forces users to switch between tabs, manage multiple logins, and manually copy content. Such a setup consumes time and hampers productivity. It also complicates workflows for users who need to integrate outputs from different models. A unified solution can address these challenges effectively.

### **Lack of Multi-Model Accessibility**:

### Most existing tools are restricted to a single AI model, limiting flexibility and performance comparison. Users often have preferences or needs that vary across tasks, such as creative writing vs. technical responses. Without access to multiple models, users can’t leverage each model’s unique strengths. This limitation often leads to repetitive trial and error. AI Fusion solves this by allowing real-time model switching within one platform.

### **Absence of In-Browser Integration**:

Many AI services require users to open separate apps or websites, pulling them out of their main workspace. This disrupts concentration and slows down task execution. In-browser integration ensures quick access without navigating away from the task at hand. Embedding AI directly into the browser can support real-time assistance. This feature significantly improves user convenience and flow.

### **Need for Quick Interaction Features**:

Everyday communication tasks, such as replying to emails or summarizing content, demand fast responses. Most platforms don’t offer instant, AI-powered buttons for common replies or follow-ups. Users are left typing repetitive responses manually, which consumes time. AI Fusion introduces one-click smart replies and summaries to speed up interactions. These features enhance user responsiveness and efficiency.

### **Demand for Combined Visual and Text AI Tools:**

Text and image generation are typically handled by separate tools or platforms. This disconnect restricts creative processes where both mediums are required simultaneously. Users working on presentations, social media posts, or marketing content benefit from integrated visual and textual AI. A unified tool simplifies the creative workflow. AI Fusion merges these capabilities to deliver a complete AI-powered workspace.

## **Objective**

1. **Unified AI Access**

Provide a single platform that integrates multiple AI models like GPT, Claude, and Gemini for streamlined usage.

1. **In-Browser Operation**

Develop a Chrome extension to ensure users can access AI features without leaving their current workflow.

1. **Smart Email Features**

Implement quick-reply buttons, summarization, and letter generation tools to simplify email handling.

1. **Image Generation**

Incorporate AI-powered image creation to support visual content generation within the same interface.

1. **User-Friendly Design**

Ensure the interface is compact, intuitive, and responsive, making AI accessible even to non-technical users.

## **Significance**

1. **Enhanced Productivity**

AI Fusion reduces the time users spend switching between different AI tools.  
By providing multi-model access and smart automation features in one place, it boosts overall efficiency in daily digital tasks.

1. **Unified Multi-Model Experience**

The integration of GPT, Claude, and Gemini into a single extension empowers users with flexibility.They can choose the best model for each task, enabling higher quality outputs and better decision-making.

1. **Real-Time Assistance**

With in-browser access, users can interact with AI tools without disrupting their workflow.This leads to faster task execution, especially for tasks like email replies, content summaries, and ideation.

1. **Creative Empowerment**

Incorporating image generation adds a visual dimension to the user experience.  
It supports designers, marketers, and content creators in generating quick visual assets without leaving the browser.

1. **Accessibility for All Users**

AI Fusion’s user-friendly interface, built with React.js, ensures that even non-technical users can benefit.Its intuitive design and one-click features make powerful AI tools more accessible to a wide audience

## **Existing System**

**ChatGPT by OpenAI**

1. **Functionality:** Natural language generation, Q&A, content writing, summarization.
2. **Accuracy:** Very high (based on GPT-4, with benchmarks showing ~86% on MMLU tasks).

* **Drawbacks:**
  1. Works as a standalone platform, requiring manual navigation.
  2. No support for switching to other models within the same session.
  3. Limited browser integration without third-party tools.

## **Google Gemini (formerly Bard)**

1. **Functionality:** Conversational AI for search, summarization, and creativity.
2. **Accuracy:** High for Google-integrated queries, though lower on complex reasoning tasks (~75–80% depending on prompt).
3. **Drawbacks:**
   1. Model outputs can be inconsistent across different contexts.
   2. Less customizable and limited access in some regions.
   3. Lacks integration with multiple tools in one interface.

## **Claude by Anthropic**

1. **Functionality:** Ethical AI assistant focused on safe, context-aware responses.
2. **Accuracy:** Around 80–85% in general tasks (based on internal benchmarking).
3. **Drawbacks:**
   1. Limited access compared to other models.
   2. Currently not as widely available or embeddable in third-party apps.

**Stable Diffusion (Image Generation)**

1. **Functionality:** AI-powered image generation from text prompts.
2. **Accuracy:** High-quality outputs depending on model and training (realism varies by version).

* **Drawbacks:**
  1. Only handle images — no text or multitasking features.
  2. Require separate usage from text-based AI models.

**Summary of Drawbacks Across Systems**

1. No single platform provides multi-model support in one interface.
2. Lack of in-browser access and integration with user workflows.
3. Missing quick interaction tools like auto-reply, summarize, and visual creation in one place.

# **Literature Review**

## **Multi-Model AI Integration**

The growing interest in artificial intelligence has led to the development of several standalone AI models such as OpenAI's ChatGPT, Google's Gemini, and Anthropic's Claude. Each of these models exhibits strengths in specific domains. For example, ChatGPT is highly effective in creative writing and code generation, while Claude focuses on ethical awareness and context sensitivity. However, current research shows that no single model consistently outperforms others in every task. This has highlighted the need for platforms that can integrate multiple models, allowing users to select the best-suited model depending on their specific use case. AI Fusion addresses this need by offering an integrated platform where users can dynamically choose between models for better results.

## **Browser-Based AI Tools**

Recent studies in Human-Computer Interaction emphasize the value of integrating AI tools directly into users’ workflows through browser extensions. Tools like Compose AI and Grammarly provide helpful features such as text completion and grammar checking, but they are typically tied to a single AI engine and lack the flexibility to handle diverse user needs. Moreover, most browser-based tools don’t support multitasking features like AI image generation or model switching. AI Fusion bridges this gap by embedding multiple AI functionalities in a single, compact interface. Users can interact with multiple models without leaving their current tab, significantly improving productivity and user experience

## **AI in Communication and Automation**

AI has been increasingly applied to streamline digital communication, especially in email handling. For instance, Google’s Smart Reply and Microsoft’s Outlook AI offer pre-generated responses based on the email’s content. These systems have proven to reduce response time and improve communication efficiency. However, they provide limited customization and rely heavily on predefined formats. AI Fusion expands on this by enabling users to generate personalized replies, create formal letters, and access quick response buttons, all while selecting the preferred AI model. This allows for a much more tailored and flexible communication experience

## **AI-Driven Image Generation**

Generative AI models like DALL·E, Stable Diffusion, and Midjourney have transformed how users create visuals using natural language prompts. These systems are widely used in fields such as marketing, design, and entertainment. However, one major limitation is that they operate separately from other productivity tools, requiring users to switch between applications. AI Fusion integrates image generation directly into the same interface used for text-based AI interactions. This combined functionality makes it easy for users to produce both written and visual content in one place, promoting creative workflows and saving time.

## **Summary and Research Gap**

While existing AI systems offer powerful features, they are often isolated and lack integration. Few platforms allow users to access multiple models and tools in one interface. AI Fusion fills this gap by providing a unified, in-browser solution that enhances productivity and user experience.

# **Proposed System**

AI Fusion is a browser-based extension designed to unify access to multiple AI models like ChatGPT, Claude, and Gemini within a compact interface. It enables users to generate text, create images, summarize content, and automate email replies efficiently. Built with React.js and Django, it streamlines AI interactions directly inside the browser for enhanced productivity.

## **Task**

The primary task of the AI Fusion project is to create an interactive browser extension that enables users to access and interact with multiple AI models, generate text/image responses, and automate communication tasks like email replies—all from a compact UI..

### **Project Planning and Setup:**

1. Define project scope, objectives, and expected deliverables such as multi-model interaction, auto-reply, and summarization.
2. Set timelines, assign development tasks, and organize tools and frameworks (React.js for frontend, Django for backend).
3. Set up the development environment with Chrome extension boilerplate, API credentials for AI models, and version control using Git.

### **Feature Design and Integration**

1. Design a floating mini-window with modular components for model selection, text generation, image creation, and summarization.
2. Implement API handlers in Django to securely route requests to ChatGPT, Claude, Gemini, and image generation services.
3. Define frontend components in React for dynamic rendering of AI responses and interaction buttons.

### **Module Development and Logic Building:**

1. Develop submodules: Quick Reply (email responses), Letter Generator, Summary Panel, and Image Generator.
2. Implement logic for selecting response tone (formal/informal) and context-based content generation.
3. Add functionality for toggling between models and dynamically updating responses based on user selection.

### **API Integration and Backend Logic :**

1. Use Django views to handle different AI requests through REST APIs.
2. Ensure secure API usage and efficient request-response handling using JSON-based communication.
3. Handle rate limiting and model-specific formatting (e.g., Gemini vs GPT responses).

### **Training and Evaluation:**

1. perform unit and integration testing for each module (Quick Replies, Model Switching, Summary Generation).
2. Validate image generation with various prompts and confirm consistent rendering.
3. Debug and resolve UI/UX inconsistencies, backend connection errors, and cross-browser issues.Integration and Deployment:

### **Deployment and Packaging:**

1. Package the extension using Webpack and configure manifest.json for Chrome deployment.
2. Test compatibility on different browsers and integrate necessary permissions.
3. Deploy backend on a server or local instance depending on hosting constraints and security settings.

### **Documentation and Reporting:**

1. Prepare technical documentation detailing extension architecture, API flow, and module dependencies.
2. Write usage instructions for end-users including how to install, select AI models, and use different features.
3. Compile a full project report highlighting the motivation, system architecture, features, limitations, and future scope.

## **Dataset**

Unlike traditional machine learning models that require structured datasets for training, AI Fusion leverages pre-trained large language models and image generators that do not require local dataset training. Instead, it processes real-time input provided by users through the browser interface. This includes text-based prompts for generating responses, email content for summarization and auto-reply, and descriptive phrases for image generation.

The system dynamically sends this input to cloud-based AI models such as ChatGPT, Gemini, and Claude, which are already trained on vast corpora of data. These models return contextually appropriate and intelligent responses. For the image generation module, user input prompts are forwarded to pre-integrated image models (like DALL·E or Stable Diffusion APIs), which return generated visuals based on the descriptions.

## **Input**

1. **Text Prompts:** Free-form questions or commands for AI responses (e.g., “Explain quantum computing”).
2. **Email Content:** Used for generating replies or summarizing emails.
3. **Image Descriptions:** Short phrases for image generation (e.g., “A cat on a surfboard in space”).
4. **Model Selection Commands:** Choosing between GPT, Claude, or Gemini for personalized results.
5. **Quick Reply Buttons:** Single-tap inputs like “Yes,” “Thanks,” “Sorry,” for auto-generated responses.

## **Output**

1. **Textual Responses:** Generated by the selected AI model based on the user’s query or prompt.
2. **Summarized Emails:** Concise summaries of lengthy email threads.
3. **Auto-Replies:** Context-aware, quick responses to emails based on tone and intent.
4. **Generated Images:** Visuals based on natural language descriptions, returned by the integrated image model.
5. **Dynamic Interface Updates:** The mini window displays real-time output, toggling between text and image views.

# **Technologies and Frameworks Used**

The development of the AI Fusion browser extension involves a combination of modern web technologies and backend frameworks that ensure seamless integration with multiple AI models and efficient user interaction.

**Frontend-Js:**

React.js is used to build the dynamic and interactive user interface of the extension. It enables the creation of reusable UI components, allowing users to quickly select AI models, enter prompts, and view responses in a responsive mini-window format.

**Styling– Tailwind CSS**

Tailwind CSS provides utility-first styling for rapid and consistent UI development. It helps in designing a clean, compact, and user-friendly interface within the Chrome extension environment.

**Backend – Django**

Django is employed to manage backend operations, including user session handling, API request routing, and managing authentication tokens for third-party AI APIs. It provides a secure and scalable backend structure.

**API Integration**

AI Fusion communicates with multiple AI model APIs like OpenAI (GPT), Gemini, and Claude using RESTful HTTP requests. These external APIs handle natural language understanding, text generation, and image creation, which are then returned and displayed by the extension.

**Chrome Extension API**

Chrome’s Extension APIs are used to integrate the tool directly into the browser environment, enabling users to access AI features while browsing websites or reading emails without leaving their current page.

.

# **System Design**

## The basic architecture of the proposed model is designed to develop a robust, scalable, and user-friendly Chrome extension that integrates multiple AI models (such as GPT, Gemini, Claude) in a unified interface.

## **Use Case Diagram**

Use-case diagrams describe the high-level functionalities and interaction between users and the AI Fusion system. In this system, the primary user interacts with the extension to select AI models, enter queries, receive responses, generate images, and utilize email tools.

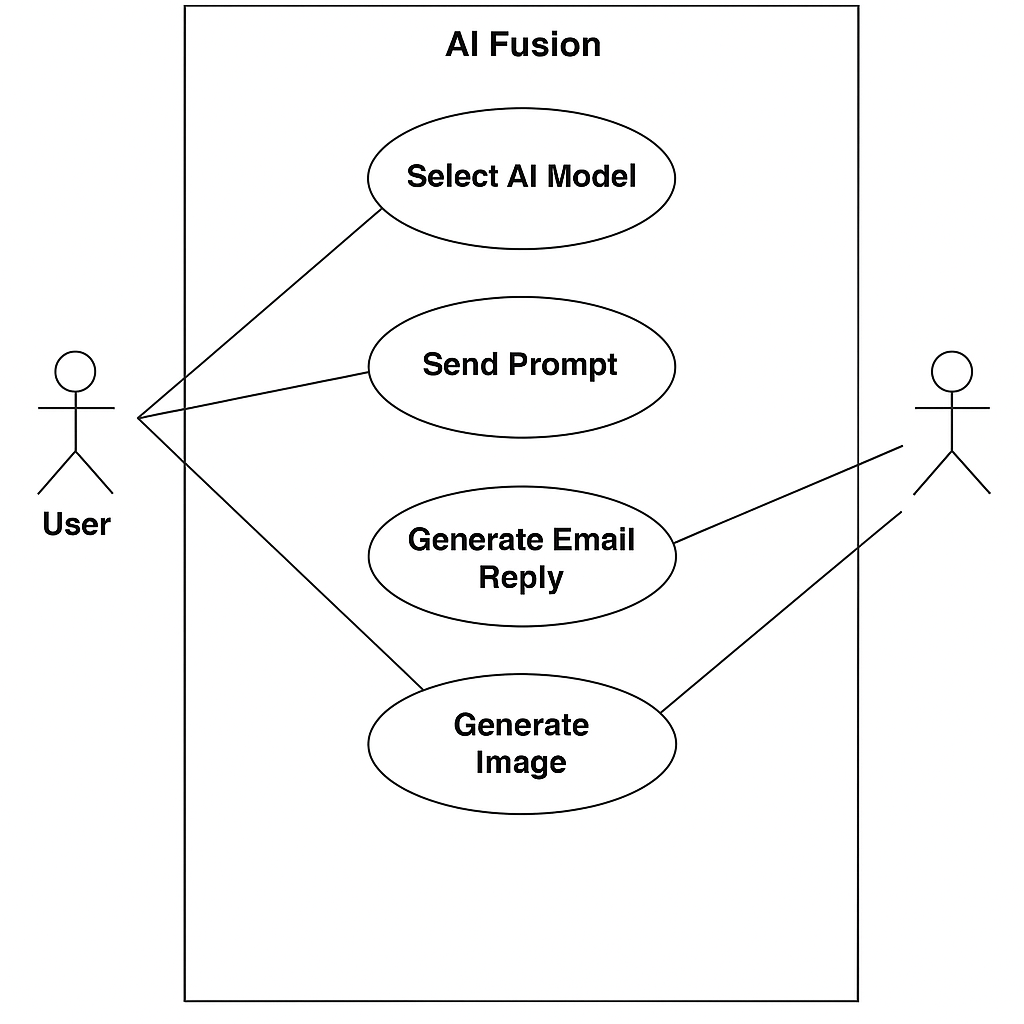


Figure - Use case diagram

## **Class Diagram**

The class diagram for **AI Fusion** represents the static structure of the Chrome extension system. It defines the key components such as classes, their attributes, methods, and the relationships between them. The diagram outlines how the frontend (built using ReactJS) interacts with backend services (developed in Django), including the AI model handler, image generation module, and email utilities.

Each class in the system—such as UserRequest, AIModelHandler, EmailAutoReply, and ImageGenerator—has specific responsibilities. For instance, the AIModelHandler manages API calls to various AI models (GPT, Gemini, Claude), while the EmailAutoReply class handles predefined response generation. The class diagram provides a blueprint for development and helps ensure modularity, reusability, and maintainability of the system.

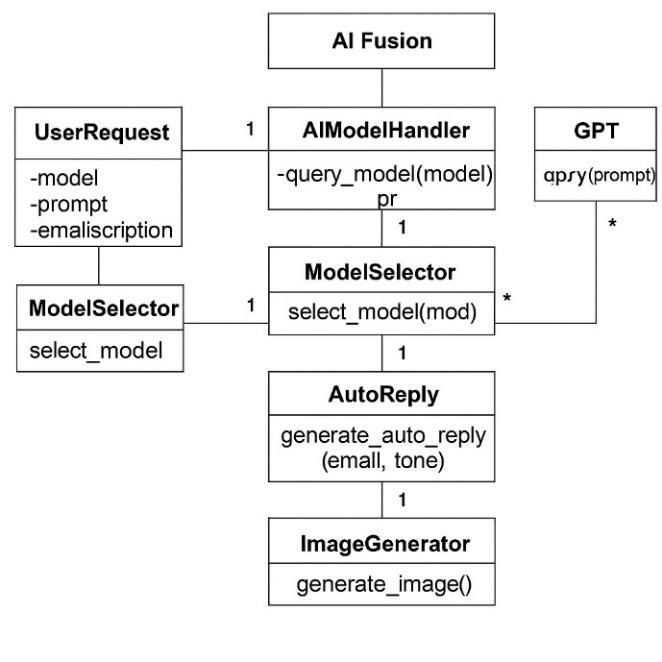


Figure - class diagram

## **Activity Diagram**

The activity diagram of **AI Fusion** visually represents the workflow involved in interacting with the AI-powered Chrome extension for multi-model integration. It outlines the sequence of operations from the user's interaction to the system's intelligent response. In this context, the activity diagram showcases the dynamic behavior of the AI Fusion system by modeling key processes such as selecting an AI model (e.g., GPT, Claude, Gemini), uploading input or composing an email, generating replies, summarizing content, and displaying responses. The control flow can be sequential—such as user input leading to response generation—or branched—such as choosing different models or AI features. This diagram offers a clear visualization of how the extension's core components interact to deliver an efficient.

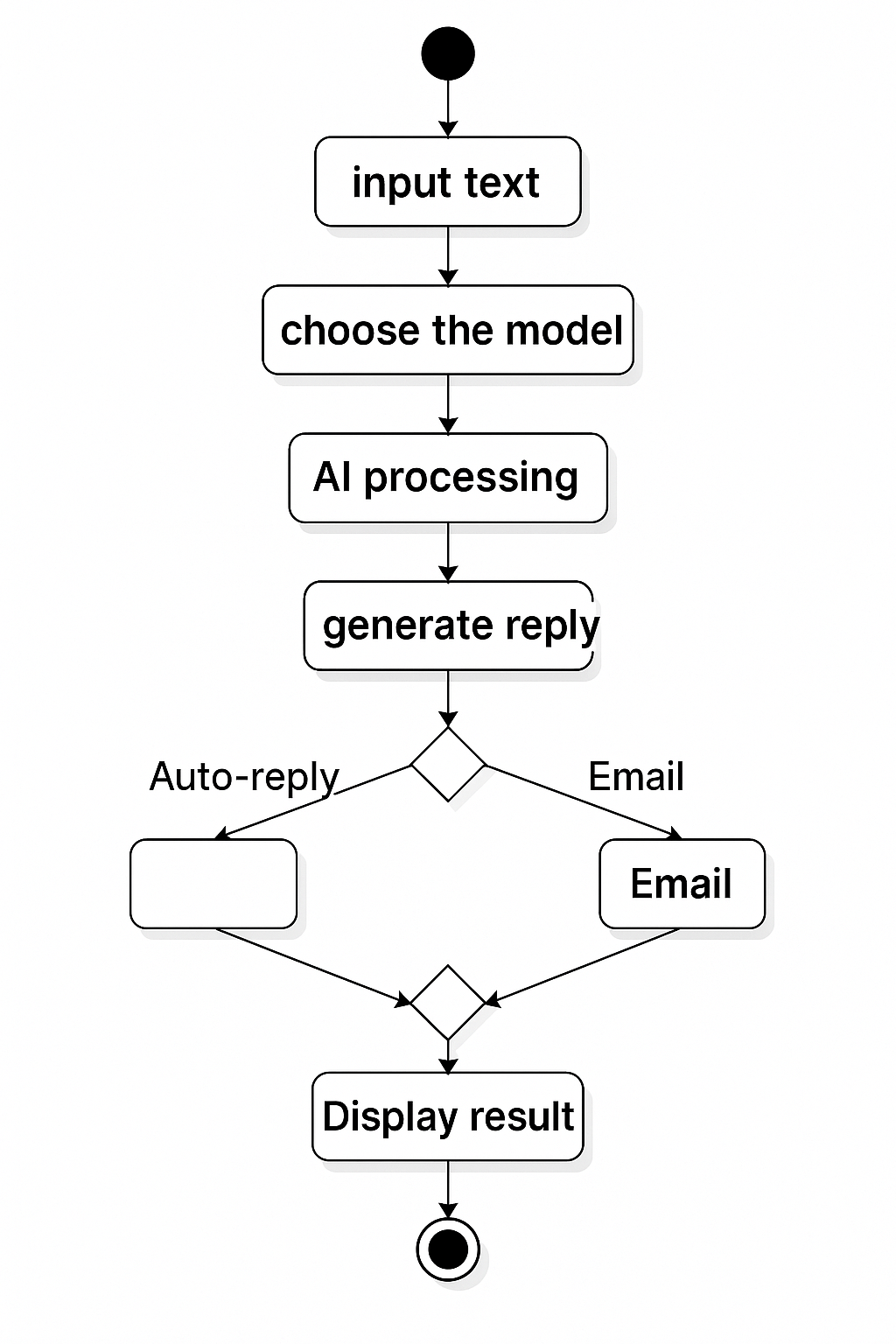


Figure - Activity Diagram

## **Sequence Diagram**

**A sequence diagram** is a type of interaction diagram that illustrates how—and in what order—various components of the AI Fusion system interact to accomplish a task. Figure 4.4 shows the sequence diagram representing the user interaction with AI Fusion. These diagrams are valuable for developers and stakeholders to visualize the dynamic behavior of the system, such as selecting AI models, sending user queries, and receiving responses. In the context of AI Fusion, it effectively outlines the flow between the user, the interface, and the integrated AI services, helping to document the process and streamline development.

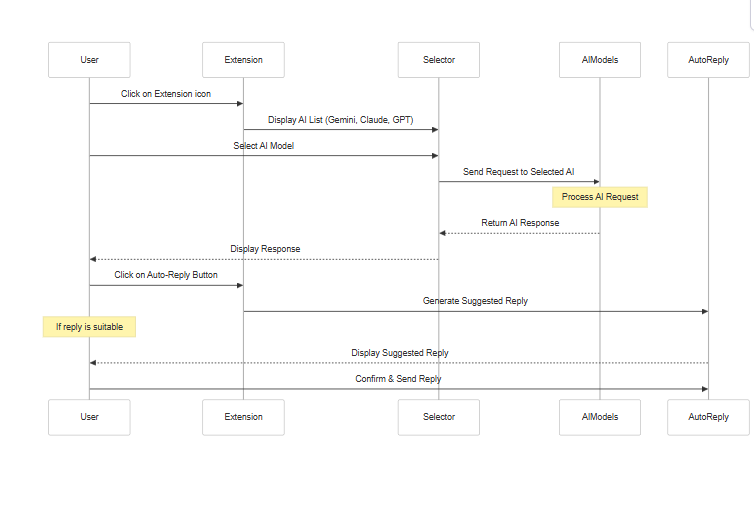


Figure - Sequence Diagram

## **Collaboration Diagram**

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming.

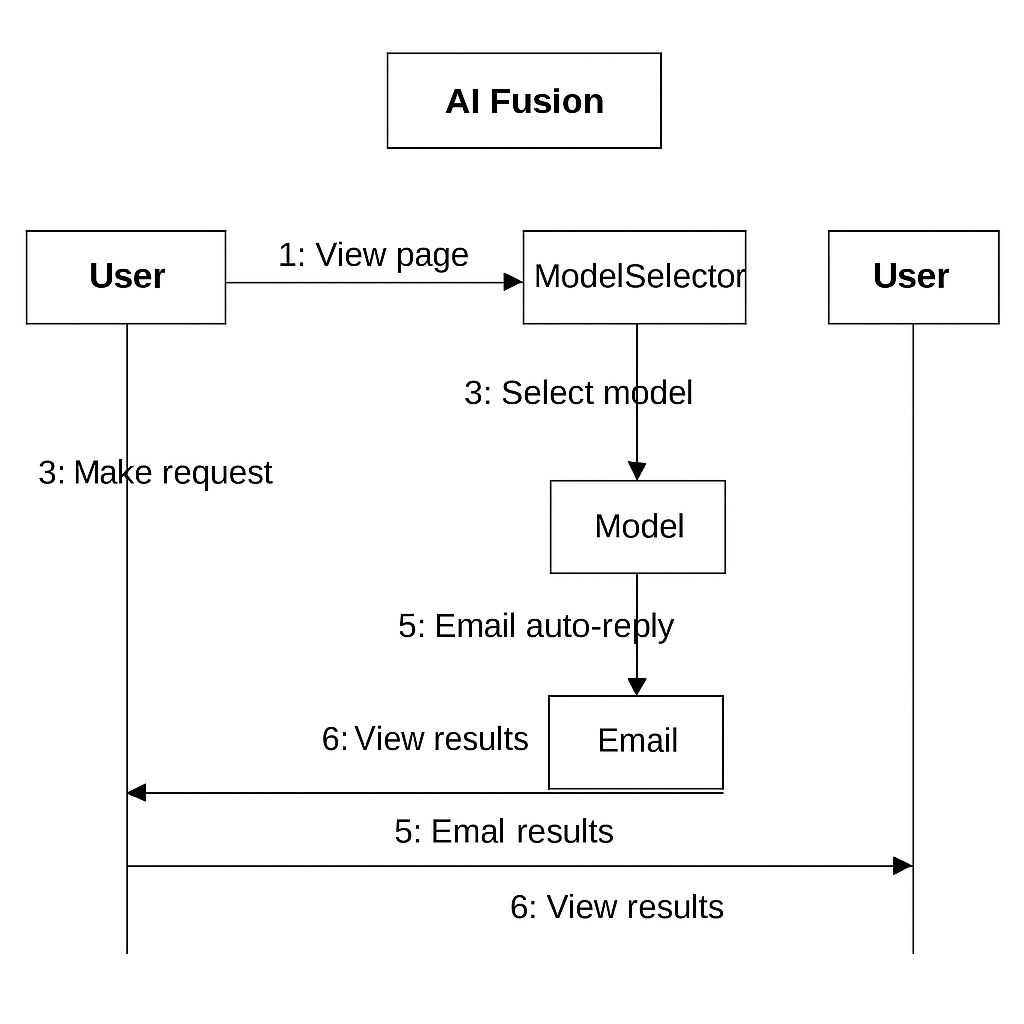


Figure - collaborative diagram

## **State Chart Diagram**

A state diagram, also known as a state machine diagram or state chart diagram, is an illustration of the states an object can attain as well as the transitions between those states in the Unified Modelling Language (UML). The Figure 4.5 shows the state chart diagram representation of system.

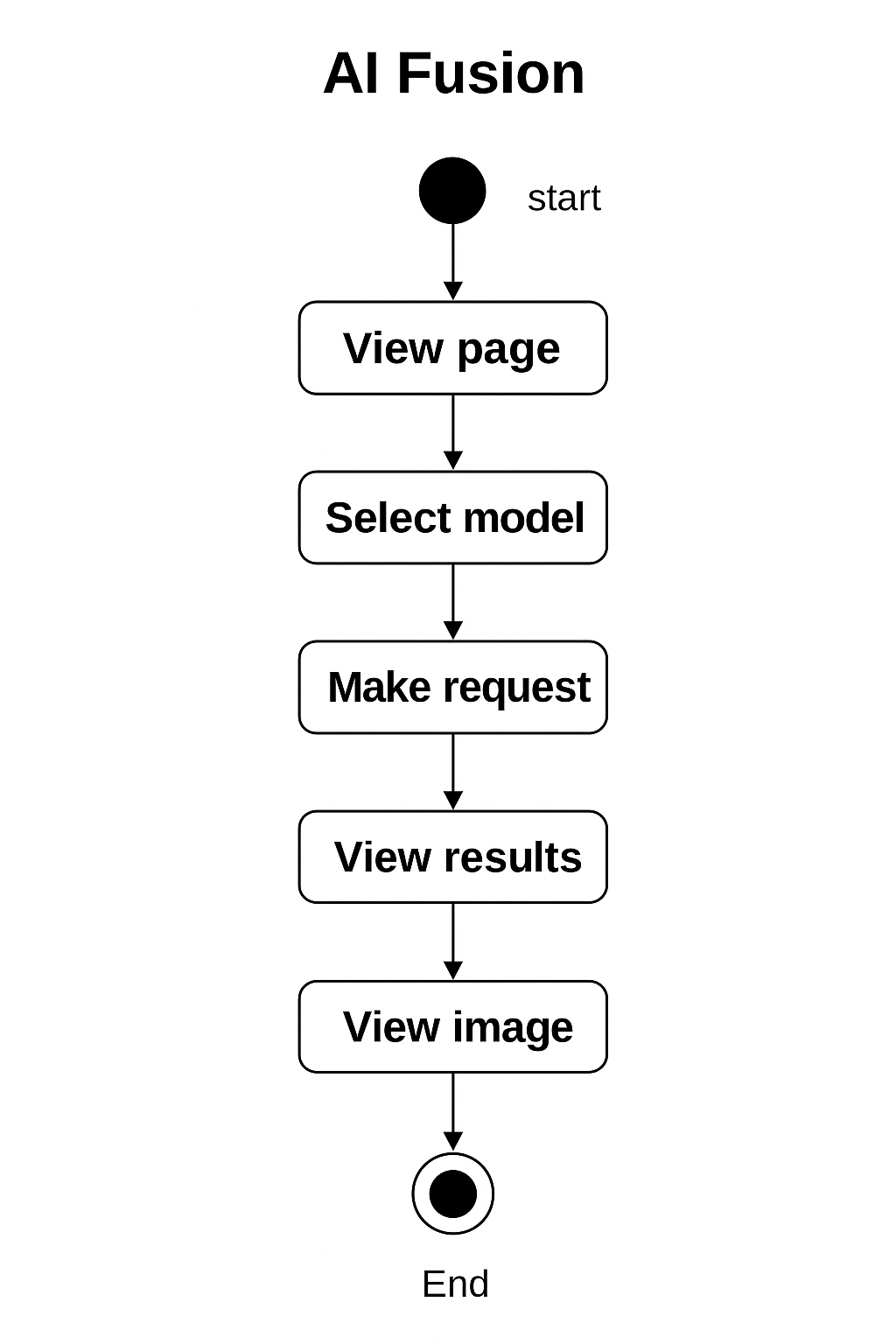


Figure - start chart Diagram

# **Implementation**

We developed a robust browser-based extension named AI Fusion, enabling users to interact with various AI models such as GPT, Gemini, and Claude. The extension offers capabilities such as multi-model querying, email auto-reply generation, and image generation, all from a mini popup interface. By integrating these features into a unified platform, AI Fusion enhances productivity, creativity, and communication workflows for users.

## **Requirements**

Requirements are critical to the success of the AI Fusion project. Meeting them ensures the extension operates smoothly, avoids compatibility issues, and provides a consistent user experience across systems. Failure to adhere to these requirements could lead to extension crashes, inconsistent performance, or limited functionality.Hardware Requirements

Hardware compatibility is key for running the development tools and backend services powering AI Fusion effectively. These specifications ensure smooth performance during development and testing:

* 1. RAM: Minimum 16GB
  2. Storage: Minimum 12GB SSD
  3. CPU: X64-based processor
  4. Operating System: 64-bit OS

**SoftwareRequirements**

AI Fusion’s functionality depends on reliable and compatible software components:

1. **Programming Language**: Python 3.10 or higher
2. **Frontend IDE**: Visual Studio Code
3. **Browser**: Chrome (for extension deployment and testing)
4. **Backend Framework**: Django (for API and logic handling)

### **Libraries**

**Core Framework & Server**

1. **Django==5.1**: High-level Python web framework for building secure, maintainable web apps.
2. **djangorestframework==3.15.2**: Toolkit for building Web APIs with Django.
3. **django-cors-headers==4.4.0**: Handles Cross-Origin Resource Sharing (CORS) headers in Django apps.
4. **asgiref==3.8.1**: ASGI support for Django for async capabilities.

**Search & Data Parsing**

1. **beautifulsoup4==4.12.3**: Extracts data from HTML and XML.
2. **lxml==5.3.0**: Powerful XML and HTML processing library.
3. **soupsieve==2.6**: CSS selector library used by BeautifulSoup.

## **Networking & API Calls**

## **httpx==0.27.0**, **httpcore==1.0.5**, **requests==2.32.3**, **urllib3==2.2.2**: Libraries for handling HTTP requests.

## **google-api-python-client==2.142.0**, **google-cloud-secret-manager==2.20.2**: For integrating with Google Cloud services.

## **grpcio==1.66.0**, **grpcio-status==1.62.3**: gRPC protocol support.

**ML/Data Analysis Libraries**

1. **numpy==2.1.0**: Numerical operations and array handling.
2. **pandas==2.2.2**: Data analysis and manipulation.
3. **scipy==1.14.1**: Scientific computing.
4. **scikit-learn (implied)**: Not listed directly, but likely used with the ML stack.

**Parsing, Serialization & Utilities**

1. **PyYAML==6.0.2**: YAML parser and emitter.
2. **pydantic==2.8.2**, **pydantic\_core==2.20.1**: Data validation and parsing.
3. **python-dotenv==1.0.1**: Manage environment variables.
4. **protobuf==4.25.4**, **proto-plus==1.24.0**: Google's Protocol Buffers for structured data.

## **Code**

Code for implementation of AI Fusion

### **GitHub Link**

<https://github.com/Manjula-Kavya/ai-fusion-backend>

[**https://github.com/Manjula-Kavya/ai-fusion-frontend**](https://github.com/Manjula-Kavya/ai-fusion-frontend)

The above GitHub repository contains code implementation, thesis work and presentation PPT for the project permission based detection of AI Fusion.

### **Importing all necessary packages**

The backend uses a wide range of Python libraries including:

1. **Django and DRF** for REST API construction.
2. **OpenAI, Anthropic, Google Generative AI, Hugging Face** to connect with large language models.
3. **Pandas, NumPy, Scikit-learn** for data manipulation and future ML enhancements.
4. **PyMuPDF, BeautifulSoup, lxml** to parse and extract content from PDFs and HTML.
5. **Google Cloud client libraries** for secret management.
6. **Tokenizers, Pydantic, Protobuf** for serialization and token management.

## **Multi-Model AI Communication**

The code enables users to query different AI models (ChatGPT, Claude, Gemini) via unified API endpoints:

1. Requests are routed to the correct provider using keyword-based or model-based selection.
2. Each integration uses the respective SDK (OpenAI, Anthropic, Google Generative AI).
3. The code handles formatting prompts, sending API requests, and parsing model responses into a common format.
4. Token usage and API key security are maintained via environment variables and Google Cloud Secret Manager.

## **Email Summarization and Auto Reply Logic**

A core feature is assisting users with emails via:

1. **Summarization**: Converts long emails into brief summaries using selected AI models.
2. **Auto Replies**: Generates context-specific replies (Yes, No, Sorry, Follow-up) using prompt templates based on email content.
3. Templates are passed to the AI model as structured prompts, and responses are formatted for direct insertion into emails.

## **PDF and HTML Content Extraction**

The code also supports file parsing:

1. **PDFs** are parsed using PyMuPDF (fitz), extracting clean text content page by page.
2. **Web content** is parsed using BeautifulSoup and lxml, enabling the extraction of body text, links, and other HTML structures.
3. This functionality supports prompt injection, summarization, and document-based question answering.

## **AI Routing and Prompt Engineering**

A key architectural part is prompt routing:

1. Depending on user request type (image, summarization, general Q&A), the system dynamically forms prompts.
2. Prompts are optimized per provider (e.g., Gemini's JSON-safe structure vs OpenAI's flexible prompt).
3. A central model\_response function routes and abstracts responses, allowing easy swapping or fallback between AI providers

## **Dataset Handling and ML Extension (Planned)**

Although current functionality relies on external AI models, foundational ML support is implemented:

1. Libraries like Scikit-learn and Pandas are included for future enhancements such as user behavior modeling, usage pattern learning, or custom NLP models.
2. Dataset parsing and transformation utilities are prepped for future local model integration or analytics.

## **Frontend Architecture**

1. The React frontend is structured into components:
2. **MiniWindow.jsx** – Renders the main popup with tabs for each model.
3. **ModelSelector.jsx** – Lets users choose among GPT, Claude, and Gemini.
4. **EmailQuickReply.jsx** – Generates canned responses like "Yes", "No", "Thanks", etc.
5. **ImageGenerator.jsx** – Sends prompts and displays generated images.
6. **Summarizer.jsx** – Summarizes email content using selected model.

1. Saves the GeneticSelector object to a file named ga.pkl in the specified directory using the pickle module. This allows the object to be loaded and used later without having to re-run the genetic algorithm.

The frontend listens for the summary and updates the UI using a useState hook.

1. Evaluates the performance of the trained neural network on the test data X\_test and labels y\_test using the genetic feature selection results from the sel object.
2. Finally, the trained neural network model AN is being serialized and saved to a file named ANN\_GA.pkl in the specified directory.

## **Model Switching Logic**

The Chrome extension supports multi-model integration through:

* A dropdown allowing selection of GPT, Claude, or Gemini.
* The selected model is stored in the frontend state and passed with each request to the backend.
* Backend reads the model identifier and uses respective API keys and endpoints to fetch results.

## **Image Generation Module**

Users can enter a prompt and generate an image via the image generation tab.

* The prompt is sent to the backend, which uses the OpenAI or Gemini Vision API to create an image.
* The image is returned as a base64-encoded string and displayed in the UI.
* Users can copy or save the image directly from the extension.

## **Email Auto-Reply Feature**

This feature allows users to auto-generate replies to emails by selecting a response type (Yes/No/Thanks/Sorry/Follow-Up). Here's how it works:

* The content of the email is passed to the backend with a specified intent.
* The backend formats the message and queries the selected model (e.g., Gemini or GPT) to return a response.
* The result is then rendered as a suggested reply below the email.

# **Results**

## **Interface**

The interface for AI Fusion is built using **React.js** for the frontend and **Python Flask** as the backend framework. Upon running the application from the development environment, the Flask backend serves the application which is displayed in the browser through a user-friendly React-based UI.

The interface is designed to be clean, interactive, and intuitive, allowing users to:

1. **Access a floating mini-window** integrated into the browser via a Chrome Extension.
2. **Select from multiple AI models** such as GPT, Gemini, and Claude.
3. **Input queries or tasks** to the selected AI model directly from the mini interface.
4. **Generate responses instantly**, with support for both text-based and image-based queries.
   1. Additionally, the interface includes advanced features such as:
5. **Auto-reply options for emails** like "Yes", "No", "Sorry", "Thanks", and "Follow-up".
6. **Summarization functionality** for incoming emails.
7. **Image generation** capabilities using AI models, accessible directly from the interface.

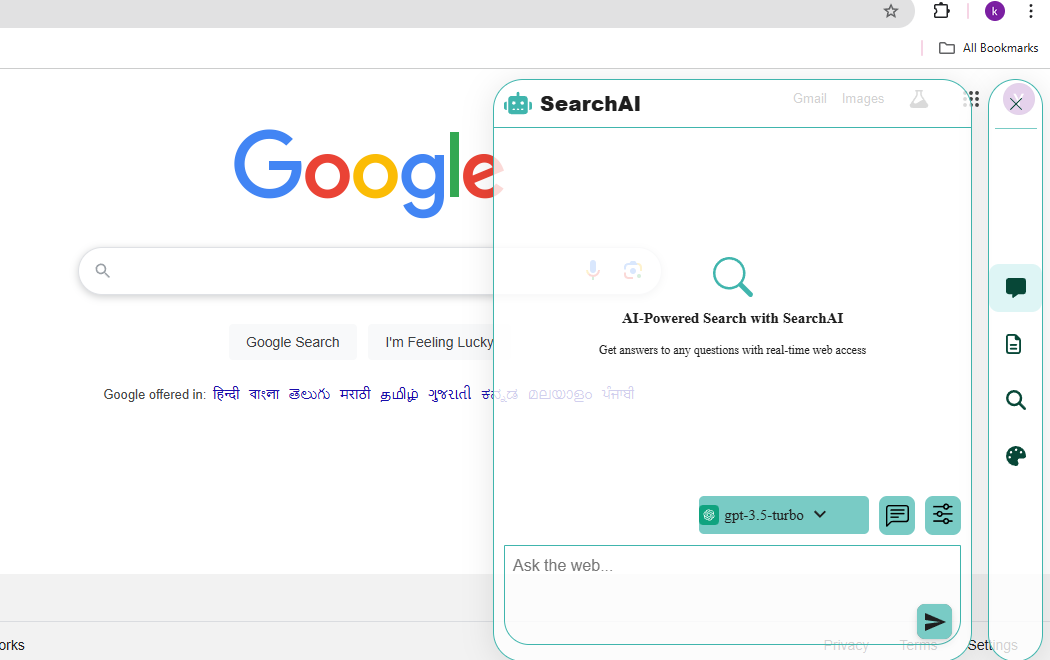


Figure - user interface

## **Output when user make a request**

When a user interacts with the AI Fusion interface—whether by entering a prompt, uploading an image, or selecting an email for summarization—the system processes the input based on the selected AI model and displays the result in real time.

The response section is styled clearly and consistently, allowing the user to copy the output, regenerate a different result, or switch models on the fly for comparison. This immediate and seamless feedback enhances user experience by making the AI's output both accessible and actionable.

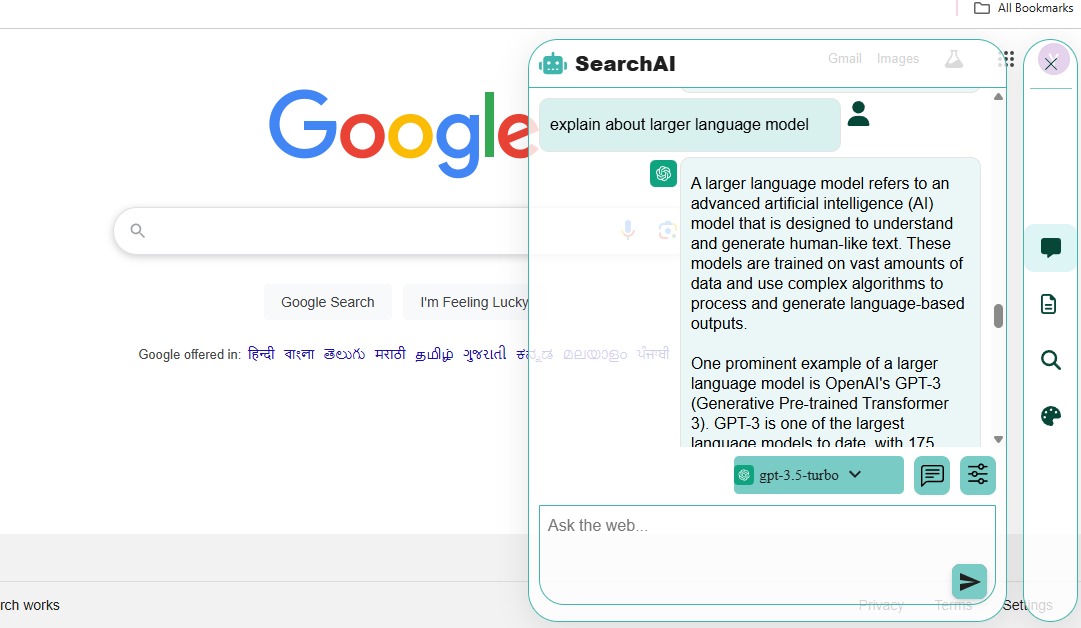


Figure - Result when user make request

## **Output when user make request to generate image**

When the user selects the image generation option, they can enter a prompt describing the desired image. Once submitted, the selected AI model processes the input and renders the image directly within the interface. The generated image is displayed with options to download or regenerate it.

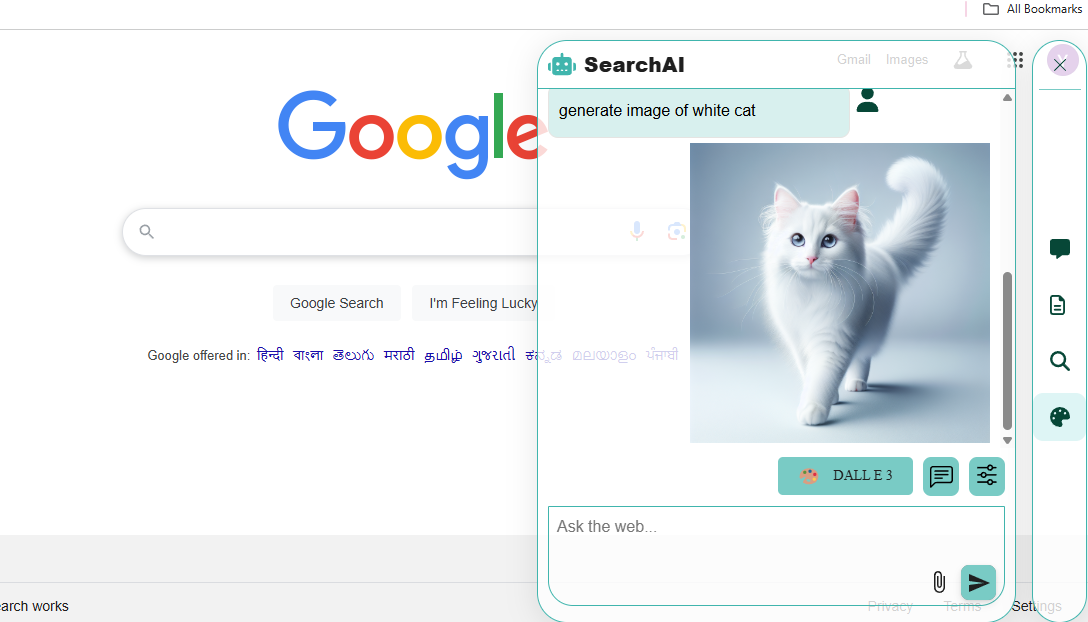


Figure - Result when user make request to generate image

## **Output when user make request to summarize the mail**

## 

Figure - Result when user make request to summarization of mail

## **Output when user make request to compose the mail**

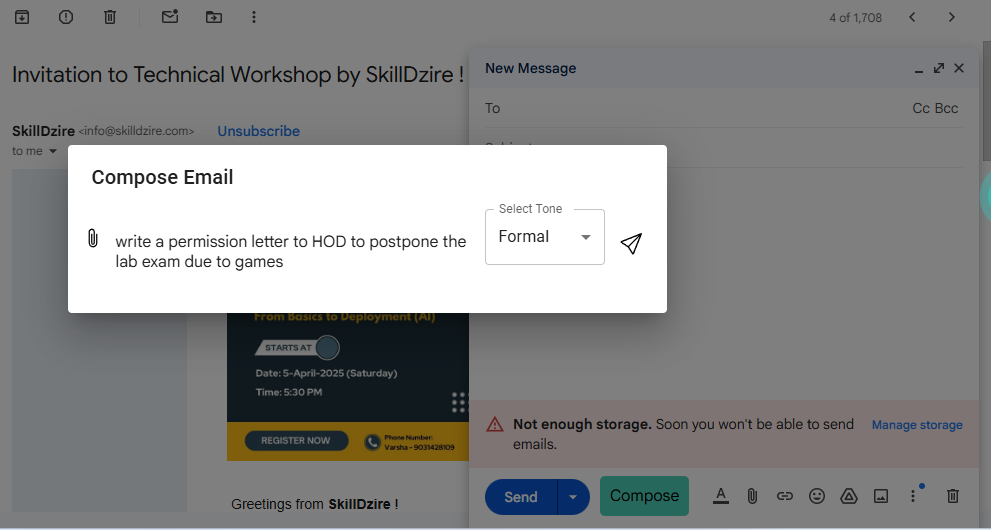


Figure - Result when user make request to summarization of mail

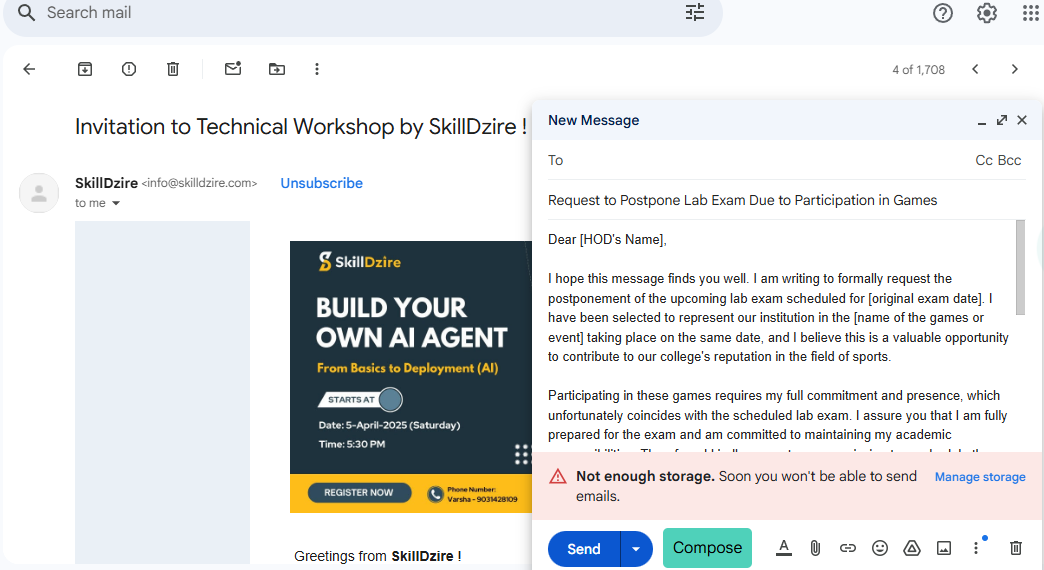


Figure - Result of requested mail

# **Conclusions**

AI Fusion is a powerful and user-friendly Chrome extension that integrates multiple AI models into a single interface. It enables users to chat with different AI models, auto-generate and reply to emails, and create images from text prompts. Built using React.js for the frontend and Django for the backend, AI Fusion enhances productivity by streamlining interactions with AI. Its seamless interface, multi-model support, and smart email tools make it a versatile solution for both personal and professional use.

AI Fusion not only simplifies access to leading AI technologies like ChatGPT, Gemini, and Claude, but also empowers users with smart automation features. With tools like one-click email replies, instant summaries, and context-aware suggestions, users can handle communication tasks faster and more effectively. The image generation feature further enhances creativity by turning ideas into visuals in seconds. Overall, AI Fusion stands as a comprehensive assistant designed to boost efficiency, creativity, and seamless multi-AI collaboration—all from a compact, easy-to-use browser extension..

# **References**

[1] Kannan, A.; Kurach, K.; Ravi, S.; Kaufmann, T.; Tomkins, A.; Miklos, B.; Corrado, G. Smart Reply: Automated Response Suggestion for Email. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), San Francisco, CA, USA, 13–17 August 2016; pp. 955–964.

[2] Shokouhi, M.; Guo, Q. Detecting and Representing Latent Factors of Email Conversations. In Proceedings of the 24th International Conference on World Wide Web (WWW), Florence, Italy, 18–22 May 2015; pp. 1089–1099.

[1] Blasch, E.; Bin, J.; Liu, Z. Certifiable Artificial Intelligence Through Data Fusion. In Proceedings of the 24th International Conference on Information Fusion (FUSION), Sun City, South Africa, 5–8 July 2021; pp. 1–8.

[2] Islam, M.A.; Anderson, D.T.; Pinar, A.J.; Havens, T.C.; Scott, G.; Keller, J.M. Enabling Explainable Fusion in Deep Learning with Fuzzy Integral Neural Networks. In Proceedings of the 22nd International Conference on Information Fusion (FUSION), Ottawa, ON, Canada, 2–5 July 2019; pp. 1–8.

[3] Zhang, L.; Liu, Z.; Zhang, S.; Yang, X.; Qiao, H.; Huang, K.; Hussain, A. Cross-Modality Interactive Attention Network for Multispectral Pedestrian Detection. *Information Fusion* 2019, 50, 20–29.