

ResearchHub AI

Intelligent Research Paper Management and Analysis System using Agentic AI

Project Description:

With the exponential growth of academic research publications and the increasing complexity of staying current with scientific literature, researchers often struggle to efficiently discover, organize, and analyze relevant research papers. Traditional methods of paper management involve manual searching, downloading, organizing, and reading through extensive literature—a time-consuming process that limits research productivity. To address this challenge, ResearchHub AI was developed as an intelligent, agentic AI-powered research paper management platform. Built using React and TypeScript for the frontend, FastAPI for backend processing, and integrated with Groq's Llama 3.3 70B model for advanced natural language understanding, the platform enables researchers to seamlessly search for papers, import them into personal workspaces, and interact with an AI chatbot that provides contextual insights, summaries, and answers based on the research content.

Scenario 1: Efficient Research Paper Discovery and Management

Researchers query multiple academic databases through an intelligent search interface, receiving curated results with metadata (title, authors, date, abstract). Papers are imported into personal workspaces with a single click. The React frontend provides an intuitive browsing experience while the FastAPI backend handles database APIs, response processing, and concurrent user management seamlessly.

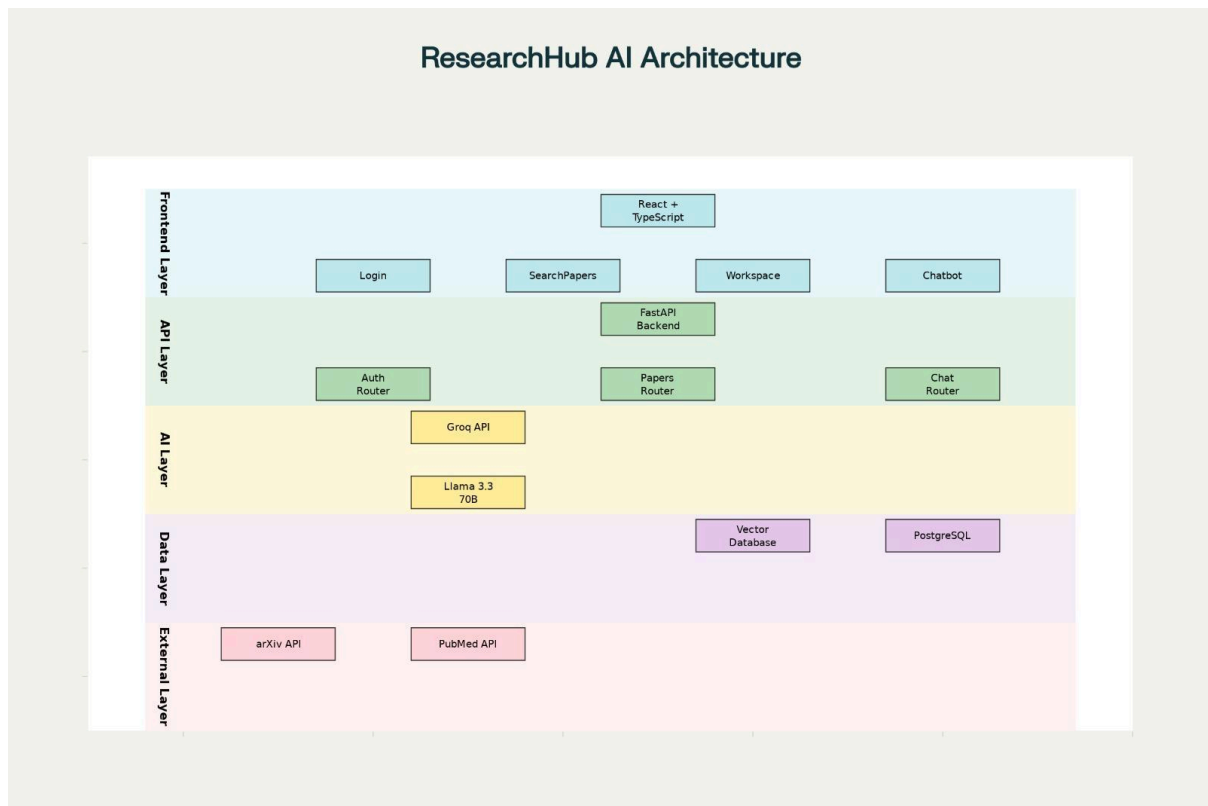
Scenario 2: AI-Powered Contextual Research Analysis

The chatbot powered by Groq's Llama 3.3 70B answers research-specific questions by accessing paper content through vector embeddings. Researchers can ask "What are key differences between transformer and CNN architectures?" or "Summarize main findings across papers?" The AI synthesizes information across multiple documents, reducing manual reading time and enabling rapid comprehension of complex topics.

Scenario 3: Organized Workspace Collaboration and Knowledge Management

Researchers create multiple workspaces for different projects—e.g., "Deep Learning Research" and "Medical Imaging Analysis." The AI chatbot maintains context-specific conversations, remembering queries and building refined insights. The system stores conversation history and paper relationships, enabling exploration tracking. FastAPI ensures efficient data management while JWT-based authentication protects sensitive research data and maintains privacy.

Architecture:



Pre-requisites:

- FastAPI Framework Knowledge: [FastAPI Documentation](#)
- Groq API Familiarity: <https://console.groq.com/>
- HTML, CSS, and JavaScript Skills: [W3Schools HTML/CSS/JavaScript Tutorials](#)
- Python Programming Proficiency: [Python Documentation](#)
- Version Control with Git: [Git Documentation](#)
- Development Environment Setup: [FastAPI Installation Guide](#)

Key Features:

- User Registration & Login
- Research Paper Search & Import
- Workspace Management
- AI Chatbot with Context Awareness
- Vector-based Semantic Search
- Conversation History

Step 1: Users register and authenticate through secure login system powered by JWT tokens for session management and security

Step 2: Researchers search for academic papers using keyword-based or semantic queries sent to the FastAPI backend API endpoints

Step 3: The system queries multiple academic databases and repositories, returning relevant papers with comprehensive metadata including abstracts, citations, and publication details

Step 4: Users can import selected papers into their personal workspaces for organization, categorization, and future reference with one-click functionality

Step 5: The AI chatbot, powered by Groq Llama 3.3 70B model, analyzes imported papers using vector embeddings and maintains contextual understanding across conversations

Step 6: Users interact with the chatbot to get summaries, insights, comparisons, and research-specific answers derived from their curated paper collection

Step 7: Built with React/TypeScript frontend and FastAPI backend, the platform ensures real-time responsiveness, secure data handling, semantic search capabilities, and intelligent conversational AI assistance for enhanced research productivity

PRIOR KNOWLEDGE:

You must have prior knowledge of the following topics to complete this project:

Agentic AI and Groq API:

An agentic AI system represents advanced artificial intelligence that can autonomously plan, reason, and execute complex tasks without constant human supervision. In this project, the Llama 3.3 70B model integrated with the Groq API enables ultra-fast inference for real-time research paper analysis and contextual conversation.

FastAPI:

Knowledge of building high-performance web backends using FastAPI, a modern Python framework used for developing RESTful APIs. FastAPI's automatic API documentation, async support, and type hints make it ideal for AI-powered applications requiring real-time responses.

Frontend Development (React, TypeScript, Tailwind CSS):

Familiarity with creating responsive, interactive web applications using React with TypeScript for type safety and better development experience. Understanding component architecture,

state management, API integration, and modern styling with Tailwind CSS is required. Knowledge of building conversational interfaces, search components, and workspace management UIs is particularly important

Vector Databases and Semantic Search:

Basic understanding of vector embeddings, semantic search, and how vector databases store and retrieve information based on conceptual similarity rather than keyword matching. This knowledge is essential for implementing the paper search and AI chatbot features that understand research context and provide relevant responses.

PROJECT WORKFLOW

Milestone 1: Requirements Specification and Project Setup

Activity 1.1: Create requirements.txt file

Activity 1.2: Install the required libraries

Milestone 2: Groq API Integration and Model Initialization

Activity 2.1: Generate Groq API Key

Activity 2.2: Configure API credentials

Activity 2.3: Initialize the Groq client and model

Milestone 3: Backend Development with FastAPI

Activity 3.1: Create authentication endpoints

Activity 3.2: Implement paper search API

Activity 3.3: Build AI chatbot endpoints

Milestone 4: Frontend Development with React and TypeScript

Activity 4.1: Create authentication components

Milestone 5: AI Agent Implementation and Context Management

Activity 5.1: Develop research paper analysis functions

Milestone 6: Testing and Deployment

Activity 6.1: Running the Backend

Activity 6.2: Running the Frontend

Activity 6.3: Configure CORS and environment variables**PROJECT STRUCTURE**

Create the ResearchHub AI project folder with the following structure:

```
ResearchHub-AI/  
├── backend/  
│   ├── main.py  
│   ├── requirements.txt  
│   ├── models/  
│   ├── routers/  
│   └── utils/  
├── frontend/  
│   ├── src/  
│   │   ├── components/  
│   │   ├── pages/  
│   │   ├── utils/  
│   │   └── App.tsx  
│   ├── package.json  
│   └── tailwind.config.js  
└── README.md
```

Milestone 1: Requirements Specification and Project Setup

This milestone focused on defining the project's objectives, functional requirements, and technical specifications. It involved selecting the necessary tools, frameworks, and libraries while setting up the development environment for smooth implementation. The goal was to establish a clear roadmap and a stable foundation for subsequent design, development, and integration phases.

Activity 1.1: Create requirements.txt file

Specifying the required libraries in the requirements.txt file ensures a smooth setup and reproducibility of the project environment. This makes it easier for others to replicate the development environment.

```
fastapi==0.104.1
uvicorn[standard]==0.24.0
python-dotenv==1.0.0
groq==0.4.1
httpx==0.25.2
python-multipart==0.0.6
python-jose[cryptography]==3.3.0
passlib[bcrypt]==1.7.4
sqlalchemy==2.0.23
databases[postgresql]==0.8.0
numpy==1.24.3
sentence-transformers==2.2.2
```

Activity 1.2: Install the required libraries

This bash script sets up an isolated Python environment using venv, which prevents dependency conflicts with other projects. The virtual environment ensures all project dependencies are contained in one place, making the development environment reproducible and portable.

```
# Create virtual environment
python -m venv venv
source venv/bin/activate

# On Windows: venv\Scripts\activate

# Install dependencies
cd backend
pip install -r requirements.txt
```

Milestone 2: Groq API Integration and Model Initialization

This milestone focused on integrating the **Groq API** to enable high-speed AI inference and response generation within the application. It involved configuring API credentials, setting up secure communication, and initializing the model for efficient query processing. The objective

was to ensure seamless backend connectivity, enabling fast and reliable AI-driven outputs during user interactions.

Activity 2.1: Generate Groq API Key

1. Visit <https://console.groq.com/>
2. Click on API Keys from the navigation menu
3. Click on Create API Key
4. Copy the generated API key
5. Store it securely in your environment variables

Activity 2.2: Configure API credentials

The .env file stores sensitive configuration data like API keys and database credentials that should not be committed to version control. Using environment variables follows security best practices by keeping secrets separate from source code and allowing easy configuration changes between development, testing, and production environments without code modifications.

Create a .env file in the backend directory:

```
GROQ_API_KEY=gsk_your_api_key_here  
SECRET_KEY=your_jwt_secret_key_here  
DATABASE_URL=your_database_url_here
```

Activity 2.3: Initialize the Groq client and model

This module initializes the Groq client with API authentication and defines model configuration parameters. The temperature setting (0.3) ensures more deterministic, focused responses suitable for research analysis rather than creative tasks. The max_tokens parameter limits response length to prevent excessive API costs, and the model is set to llama-3.3-70b-versatile which provides excellent reasoning capabilities for complex research queries. This centralized configuration makes it easy to adjust model parameters globally across the application.

```
# backend/utils/groq_client.py  
import os  
from groq import Groq  
from dotenv import load_dotenv  
  
load_dotenv()  
  
client = Groq(  
    api_key=os.getenv("GROQ_API_KEY")  
)  
  
MODEL_CONFIG = {  
    "model": "llama-3.3-70b-versatile",
```

```
"temperature": 0.3,  
"max_tokens": 2000,  
"top_p": 0.9  
}
```

Milestone 3: Backend Development with FastAPI

This milestone focused on developing the core backend infrastructure using FastAPI to manage requests, responses, and data flow efficiently. It involved creating API endpoints, integrating the AI model with the server, and ensuring smooth communication between the frontend and backend. The goal was to build a fast, scalable, and reliable backend that supports real-time interaction and AI-driven functionality.

Activity 3.1: Create authentication endpoints

This authentication module implements secure user registration and login endpoints. Passwords are hashed using bcrypt before storage in the database, ensuring they cannot be retrieved even if the database is compromised. JWT (JSON Web Token) access tokens are generated upon login, enabling stateless authentication where the frontend can include the token in subsequent requests to verify user identity without server-side session storage.

```
# backend/routers/auth.py  
from fastapi import APIRouter, HTTPException, Depends  
from passlib.context import CryptContext  
from jose import JWTError, jwt  
  
router = APIRouter()  
pwd_context = CryptContext(schemes=["bcrypt"], deprecated="auto")  
  
@router.post("/register")  
async def register(user_data: UserCreate):  
    hashed_password = pwd_context.hash(user_data.password)  
    # Store user in database  
    return {"message": "User registered successfully"}  
  
@router.post("/login")  
async def login(user_credentials: UserLogin):  
    access_token = create_access_token(data={"sub": user.email})  
    return {"access_token": access_token, "token_type": "bearer"}
```

Activity 3.2: Implement paper search API

These endpoints enable authenticated users to search academic databases for research papers and import selected papers into their personal workspace. The search endpoint queries external academic databases and returns results with metadata, while the

import endpoint stores selected papers in the database associated with the user's account. The Depends(get_current_user) parameter ensures only authenticated users can access these endpoints, and the association with current_user.id ensures users can only see and manage their own imported papers.

```
# backend/routers/papers.py
from fastapi import APIRouter, Depends
from utils.groq_client import client, MODEL_CONFIG

router = APIRouter()

@router.get("/search")
async def search_papers(query: str, current_user: User = Depends(get_current_user)):
    search_results = await query_academic_databases(query)
    return {"papers": search_results}

@router.post("/import")
async def import_paper(paper_data: PaperImport, current_user: User =
Depends(get_current_user)):
    imported_paper = await store_paper(paper_data, current_user.id)
    return {"message": "Paper imported successfully", "paper": imported_paper}
```

Activity 3.3: Build AI chatbot endpoints

This endpoint handles user queries within a specific workspace and generates AI-powered responses based on the papers in that workspace. The function retrieves all papers from the user's workspace, constructs a research context containing paper summaries and abstracts, and sends this context along with the user's question to the Groq LLM. The response is then stored in the conversation history for future reference, enabling the system to maintain context awareness across multiple interactions within the same workspace.

```
# backend/routers/chat.py
@router.post("/chat")
async def chat_with_papers(message: ChatMessage, workspace_id: int,
                           current_user: User = Depends(get_current_user)):
    workspace_papers = await get_workspace_papers(workspace_id, current_user.id)
    context = create_research_context(workspace_papers, message.content)

    response = client.chat.completions.create(
        messages=[
            {"role": "system", "content": f"You are a research assistant. Context: {context}"},
            {"role": "user", "content": message.content}
        ]
    )
```

```

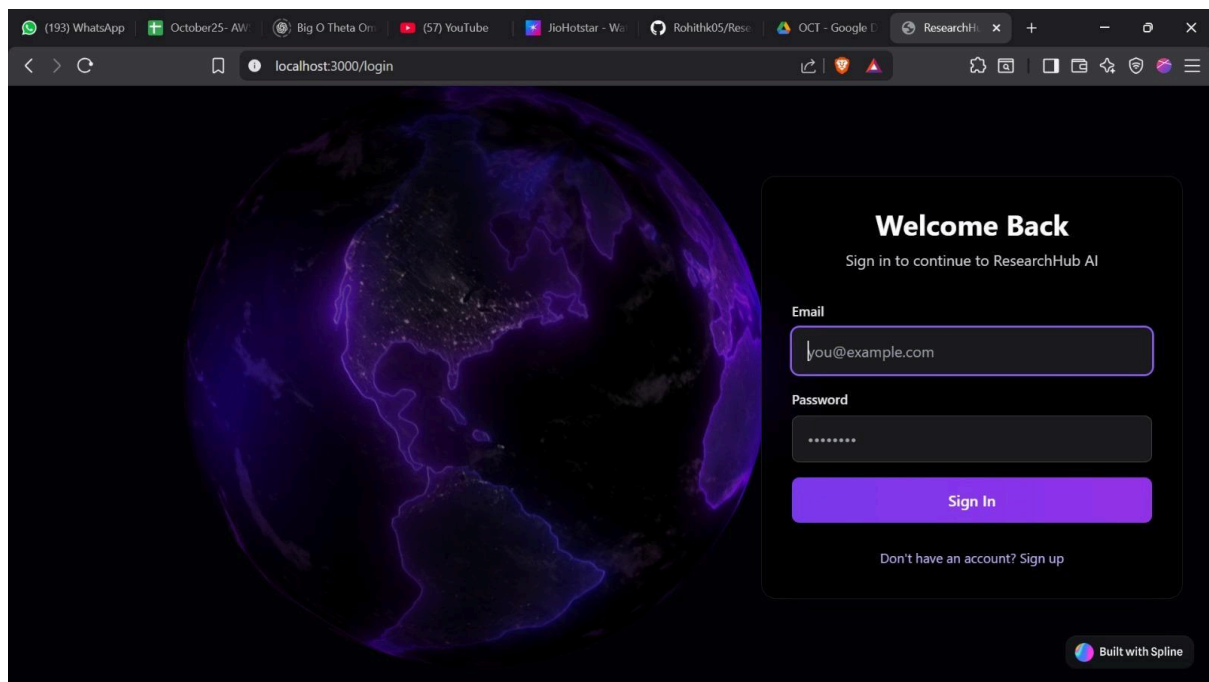
    ],
    **MODEL_CONFIG
)

await store_conversation(workspace_id, message.content,
response.choices[0].message.content)
return {"response": response.choices[0].message.content}

```

Milestone 4: Frontend Development with React and TypeScriptActivity

This milestone focused on building an interactive and responsive user interface using React and TypeScript. It involved developing dynamic components for user interaction, integrating API calls for real-time data exchange, and implementing a clean, modular design for better maintainability. The goal was to deliver a seamless, visually appealing frontend that enhances user experience while ensuring type safety and performance efficiency. Frontend login component with state management and form handling.



Milestone 5: AI Agent Implementation and Context Management

This milestone focused on implementing the **AI agent** to handle intelligent interactions and maintain contextual awareness across user sessions. It involved designing logic for multi-turn conversations, context retention, and dynamic response adaptation. The objective was to enhance the system's conversational depth and consistency, enabling more personalized, coherent, and contextually relevant AI-driven interactions.

Activity 5.1: Develop research paper analysis functions

The `create_research_context` method aggregates paper information (titles, authors, abstracts) into a structured prompt that provides the LLM with relevant background information. The `generate_research_response` method sends this context to the Groq API, enabling the AI agent to provide informed, contextually-accurate answers based on the researcher's imported papers. This architecture enables autonomous reasoning and synthesis across multiple research documents.

```
# backend/utils/research_assistant.py
class ResearchAssistant:
    def __init__(self):
        self.conversation_history = []

    def create_research_context(self, papers, query):
        context_parts = []
        for paper in papers:
            paper_context = f'''
            Title: {paper['title']}
            Authors: {' '.join(paper['authors'])}
            Abstract: {paper['abstract']}
            '''
            context_parts.append(paper_context)

        full_context = "\n---\n".join(context_parts)
        return f"Research Papers Context:\n{full_context}\n\nUser Query: {query}"

    def generate_research_response(self, context, query):
        messages = [
            {"role": "system", "content": "You are an expert research assistant."},
            {"role": "user", "content": f"Context: {context}\n\nQuestion: {query}"}
        ]

        response = client.chat.completions.create(
            messages=messages,
            **MODEL_CONFIG
        )

        return response.choices[0].message.content
```

Milestone 6: Testing and Deployment

This milestone focused on verifying the overall functionality, performance, and reliability of the system before deployment. Comprehensive testing—including unit, integration, and end-to-end validation—was conducted to ensure all components worked cohesively. After successful

testing, the application was deployed in a configured environment, ensuring smooth execution, scalability, and readiness for real-world use.

Activity 6.1: Running the Backend

This script starts with the FastAPI development server with auto-reload enabled, allowing the server to automatically restart whenever code changes are made. The `--host 0.0.0.0` parameter makes the server accessible from any network interface, while `--port 8000` specifies the server runs on port 8000. The reload feature significantly speeds up development iteration by eliminating the need to manually restart the server after code modifications.

```
# Navigate to backend directory
cd backend

# Activate virtual environment
source venv/bin/activate
# On Windows: venv\Scripts\activate

# Start FastAPI server
uvicorn main:app --reload --host 0.0.0.0 --port 8000
```

Activity 6.2: Running the Frontend

These commands set up and launch the React development server. `npm install` installs all JavaScript dependencies specified in `package.json`, while `npm run dev` or `npm start` launches the development server (typically on port 3000), which provides hot module reloading—any changes to React components are automatically reflected in the browser without manual refresh, enabling rapid frontend development and iteration.

```
# Navigate to frontend directory
cd frontend

# Install dependencies
npm install

# Start development server
npm run dev
# or
npm start
```

Activity 6.3: Configure CORS and environment variables

The `allow_origins` parameter specifies which domains can make requests to the API, `allow_credentials=True` enables authentication tokens to be sent with requests, and

`allow_methods=["*"]` permits all HTTP methods (GET, POST, PUT, DELETE, etc.), providing the flexibility needed for a full-featured API.

```
# backend/main.py
from fastapi import FastAPI
from fastapi.middleware.cors import CORSMiddleware

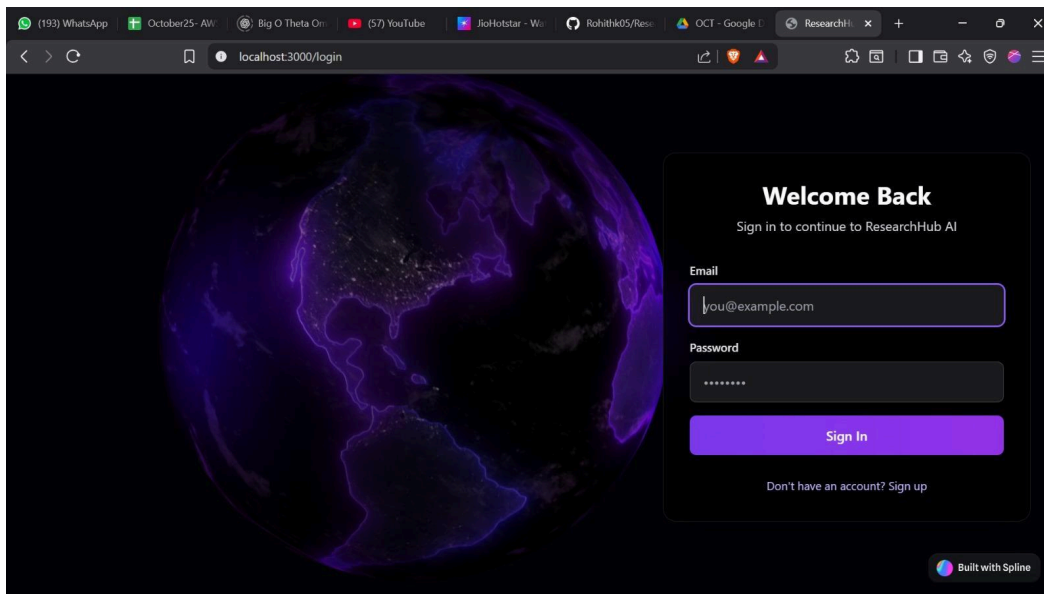
app = FastAPI(title="ResearchHub AI API", version="1.0.0")

app.add_middleware(
    CORSMiddleware,
    allow_origins=["http://localhost:3000", "https://your-frontend-domain.com"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

@app.get("/")
async def root():
    return {"message": "ResearchHub AI API is running"}
```

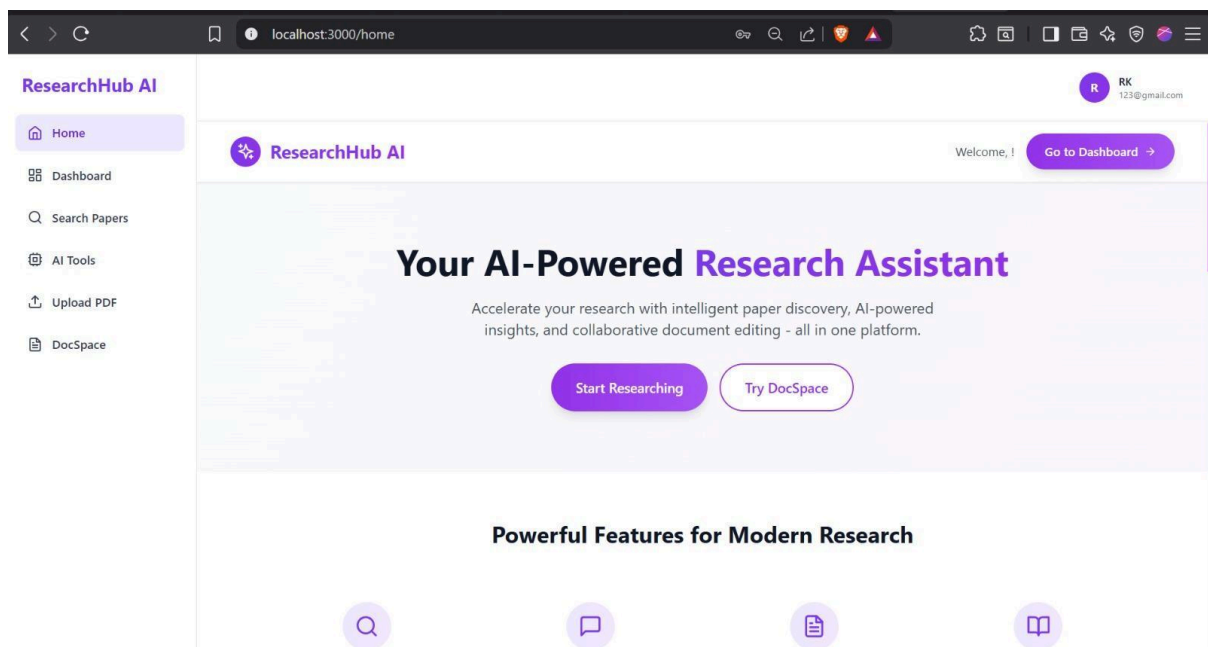
Login page:

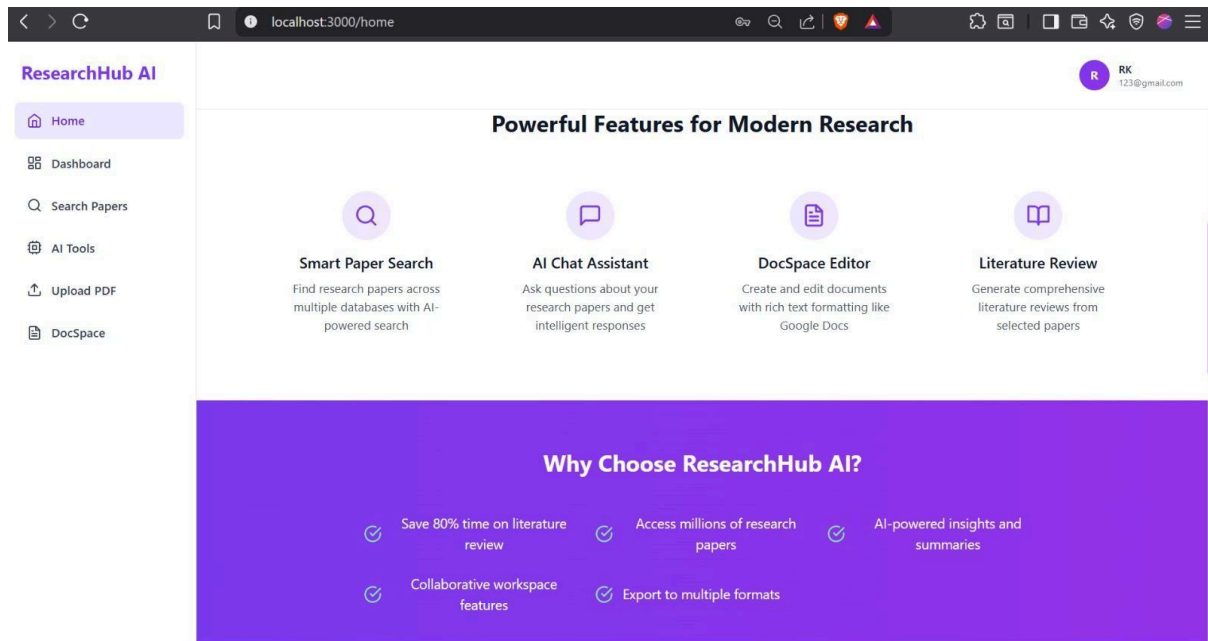
The login page serves as the primary entry point for researchers to access the ResearchHub AI platform. Users can enter their credentials securely, with form validation ensuring proper email and password format. This authentication gateway protects user research data and ensures each researcher's papers and conversations remain private and organized within their personal workspace.



Home page:

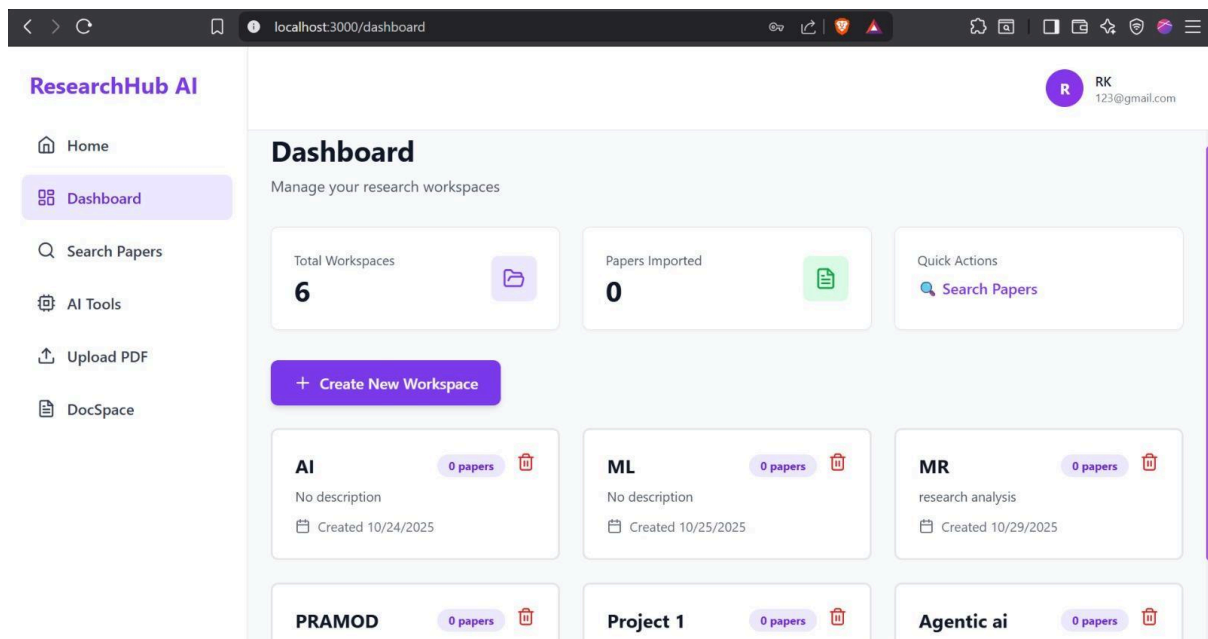
The home page provides an overview of the platform and its key features, welcoming new and returning users. It displays navigation options to access paper search functionality, existing workspaces, and the AI chatbot interface. This landing area helps users understand the platform's capabilities and quickly navigate to the features they need for their research tasks.





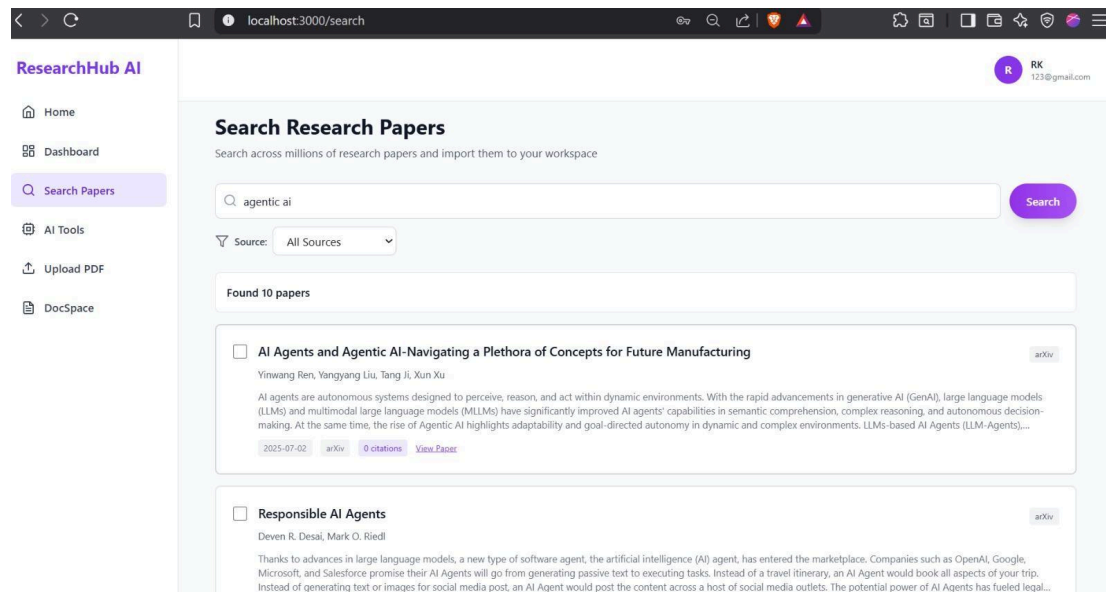
Dashboard:

The dashboard offers a personalized view of the researcher's activity and research progress. Users can see recently accessed databases papers, active workspaces, and quick links to frequently used features. This central hub enables researchers to efficiently manage their research projects and track their analysis progress across multiple ongoing studies.



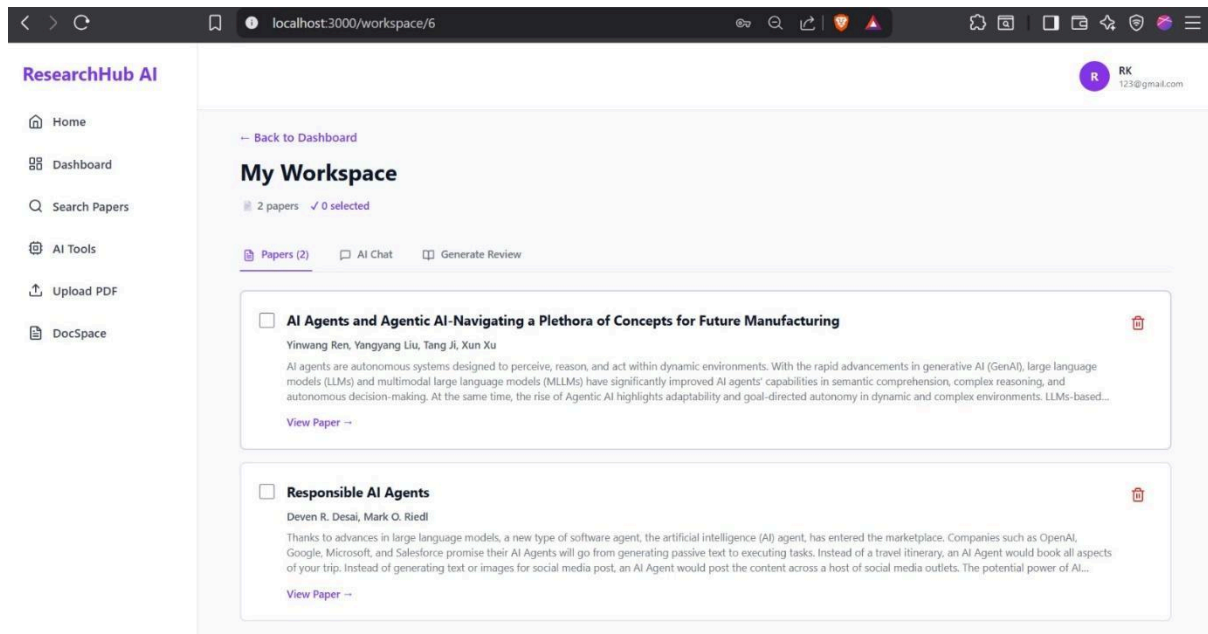
Search papers page:

The search papers page enables researchers to query academic databases for relevant publications. Users can enter keywords or research topics, filter results by date, author, or relevance, and preview paper metadata including abstracts and citations. Once relevant papers are identified, researchers can import them directly into their workspaces for further analysis and organization.

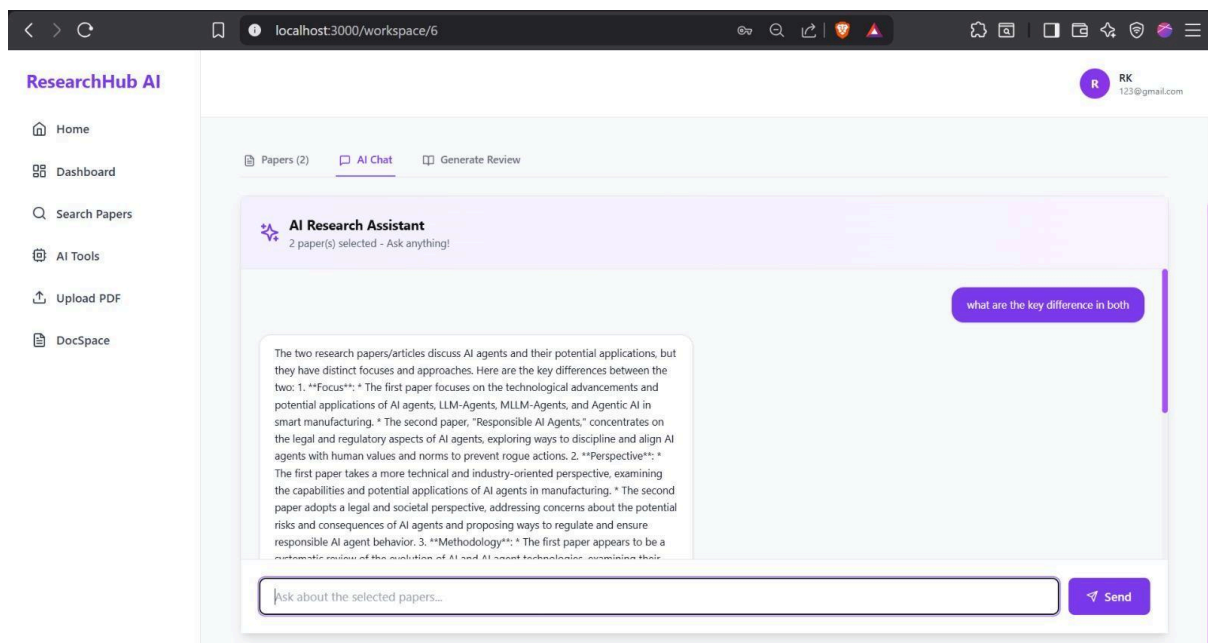


Workspace:

Workspaces function as project-specific containers where researchers organize and analyze their imported papers. Within each workspace, users can view all imported papers, create custom collections, and access the AI chatbot for research-specific queries. Workspaces maintain separate conversation histories and paper collections, allowing researchers to manage multiple independent research projects simultaneously without confusion.



The screenshot shows the 'My Workspace' page in ResearchHub AI. The left sidebar contains navigation links: Home, Dashboard, Search Papers, AI Tools, Upload PDF, and DocSpace. The main content area has a 'Back to Dashboard' link and a 'My Workspace' title. Below the title, it shows '2 papers' and '0 selected'. There are three tabs: 'Papers (2)', 'AI Chat', and 'Generate Review'. The 'Papers (2)' tab is active, displaying two paper cards. The first card is titled 'AI Agents and Agentic AI-Navigating a Plethora of Concepts for Future Manufacturing' by Yinwang Ren, Yangyang Liu, Tang Ji, and Xun Xu. The second card is titled 'Responsible AI Agents' by Deven R. Desai and Mark O. Riedl. Each card has a 'View Paper' link.



The screenshot shows the 'AI Research Assistant' interface in ResearchHub AI. The left sidebar is the same as the previous screenshot. The main content area has tabs for 'Papers (2)', 'AI Chat', and 'Generate Review'. The 'AI Chat' tab is active, displaying a chat window. The chat window has a header 'AI Research Assistant' and a sub-header '2 paper(s) selected - Ask anything!'. The chat area contains a message from the AI assistant summarizing the key differences between the two selected papers. The message is: 'The two research papers/articles discuss AI agents and their potential applications, but they have distinct focuses and approaches. Here are the key differences between the two: 1. **Focus**:: * The first paper focuses on the technological advancements and potential applications of AI agents, LLM-Agents, MLLM-Agents, and Agentic AI in smart manufacturing. * The second paper, "Responsible AI Agents," concentrates on the legal and regulatory aspects of AI agents, exploring ways to discipline and align AI agents with human values and norms to prevent rogue actions. 2. **Perspective**:: * The first paper takes a more technical and industry-oriented perspective, examining the capabilities and potential applications of AI agents in manufacturing. * The second paper adopts a legal and societal perspective, addressing concerns about the potential risks and consequences of AI agents and proposing ways to regulate and ensure responsible AI agent behavior. 3. **Methodology**:: * The first paper appears to be a systematic review of the evolution of AI and AI agent technologies, examining their...'. Below the chat area is a text input field with the placeholder 'Ask about the selected papers...' and a 'Send' button.

AI Tools:

The AI Tools section provides access to advanced research analysis capabilities powered by the Groq Llama 3.3 70B model. Researchers can leverage these tools to generate paper summaries, compare multiple papers, extract key findings, and answer domain-specific research questions. The AI agent intelligently synthesizes information across papers, providing insights that would require hours of manual analysis if done manually.

ResearchHub AI

Home

Dashboard

Search Papers

AI Tools

Upload PDF

DocSpace

RK
123@gmail.com

AI Tools

AI-powered research analysis tools • 2 papers available • 2 selected

Select Papers for Analysis

☒ AI Agents and Agentic AI-Navigating a Plethora of Concepts for Future Manufacturing
Yinwang Ren, Yangyang Liu, Tang Ji, Xun Xu

☒ Responsible AI Agents
Deven R. Desai, Mark O. Riedl

2 paper(s) selected

AI Summaries

Generate concise summaries of selected research papers

Generate Summaries

Key Insights

Extract key insights and trends from research papers

Extract Insights

Literature Review

Generate comprehensive literature reviews automatically

Generate Review

AI Summaries

Generate concise summaries of selected research papers

Generate Summaries

Key Insights

Extract key insights and trends from research papers

Extract Insights

Literature Review

Generate comprehensive literature reviews automatically

Generate Review

AI Summaries Results


Download

AI Agents and Agentic AI-Navigating a Plethora of Concepts for Future Manufacturing

This paper reviews the evolution of AI and AI agent technologies, examining the core concepts and advancements of LLM-Agents, MLLM-Agents, and Agentic AI, and explores their potential applications in smart manufacturing. The study aims to address the gap in understanding the definitions, capabilities, and practical applications of these emerging AI paradigms in manufacturing. By doing so, it opens new avenues for smart manufacturing and highlights potential challenges that may arise.

Responsible AI Agents


This paper addresses concerns around the potential risks and harms of AI Agents, proposing a computer-science approach to value-alignment to improve user control and prevent undesired actions. The authors argue that AI Agents should not be given legal personhood status and that humans are responsible for their actions, providing a guide for building and maintaining responsible AI Agents. The paper aims to mitigate perceived risks and enable desired economic outcomes by leveraging technical solutions to discipline AI Agents and align them with user norms and interactions.



AI Summaries

Generate concise summaries of selected research papers


[▶ Generate Summaries](#)



Key Insights

Extract key insights and trends from research papers

[🔍 Extract Insights](#)



Literature Review

Generate comprehensive literature reviews automatically

[📄 Generate Review](#)

Key Insights Results


[Download](#)

After analyzing the two research papers, I've extracted key insights, trends, and findings. Here are the results:

****Key Insights:****

- **Advancements in AI Agents**:** The first paper highlights the rapid advancements in generative AI, large language models (LLMs), and multimodal large language models (MLLMs), which have improved AI agents' capabilities in semantic comprehension, complex reasoning, and autonomous decision-making.
- **Agentic AI**:** The first paper introduces Agentic AI, which emphasizes adaptability and goal-directed autonomy in dynamic and complex environments, expanding AI's capabilities in information processing, environmental perception, and autonomous decision-making.
- **Responsible AI Agents**:** The second paper discusses the importance of responsible AI Agents, highlighting the need for discipline and value-alignment to prevent rogue, undesired actions and ensure desired economic outcomes.
- **Human Responsibility**:** The second paper emphasizes that humans are responsible for AI Agents' actions and provides guidance on building and maintaining responsible AI Agents.


****Trends:****



AI Summaries

Generate concise summaries of selected research papers


[▶ Generate Summaries](#)



Key Insights

Extract key insights and trends from research papers

[🔍 Extract Insights](#)



Literature Review

Generate comprehensive literature reviews automatically

[📄 Generate Review](#)

Literature Review Results

[Download](#)

****Comprehensive Literature Review: AI Agents and Responsible AI Agents****

1. Overview

The two research papers under review explore the concepts of AI Agents and Agentic AI, and their potential applications in future manufacturing. The first paper examines the evolution of AI and AI agent technologies, with a focus on Large Language Models (LLMs) and Multimodal Large Language Models (MLLMs)-based AI Agents, and Agentic AI. The second paper delves into the concerns surrounding AI Agents, including their potential for rogue commerce, human manipulation, and intellectual property harms, and proposes ways to discipline AI Agents and ensure responsible behavior.

The papers provide a comprehensive overview of the current state of AI Agents and their potential applications, as well as the challenges and risks associated with their development and deployment. The review of these papers aims to synthesize the key findings, identify research gaps, and provide conclusions on the future of AI Agents and Agentic AI.

2. Key Findings

The key findings from the two papers can be summarized as follows:

Upload PDF:

The upload PDF functionality allows researchers to add papers from their personal collection into the system. Users can drag-and-drop or browse for PDF files, which are then processed and stored in their workspace. This feature enables researchers to complement database searches with papers they already possess, creating a comprehensive personal research library within the platform.

ResearchHub AI

- Home
- Dashboard
- Search Papers
- AI Tools
- Upload PDF**
- DocSpace

RK
123@gmail.com

Upload Research Paper

Upload a PDF to extract text and generate AI insights



Upload PDF

Drop your PDF file here or click to browse

Select PDF File

RRpdf ✓

Generate AI Summary

Save to Workspace

Download Text

Extracted Text:

Generate AI Summary

Save to Workspace

Download Text

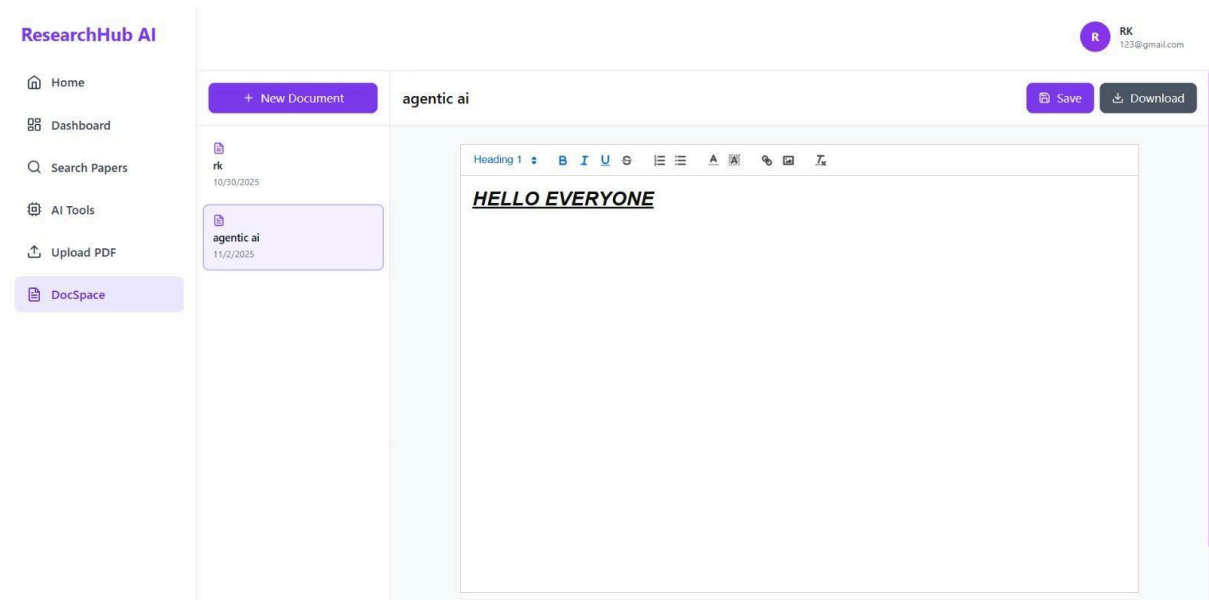
AI Summary

Here is a comprehensive summary of the research paper in 7 bullet points:

- * **Main topic**: The paper explores the use of mobile devices by advanced learners of English as a foreign language (EFL) for language study, with a focus on learner autonomy and the role of mobile technology in language learning.
- * **Research question**: The study investigates how students engage with their mobile devices to develop learning experiences that meet their needs and goals as EFL learners, and why and how they do so.
- * **Methodology**: The study used a semi-structured interview approach, where 20 Polish university students of English philology were interviewed about their use of mobile devices for English language study. The data was analyzed using both qualitative and quantitative methods.
- * **Key findings**: The study found that all participants used their mobile devices (smartphones and/or tablets) to learn English, with the most common activities being practicing vocabulary, checking pronunciation, and accessing online resources. The majority of students used their mobile devices for informal learning, and most reported that using mobile devices helped them study English more effectively and efficiently.
- * **Device usage and preferences**: The study found that smartphones were the most commonly used device, and that students preferred using mobile devices due to their convenience, ease of use, and quick access to online resources. Students also reported using a variety of mobile apps, including dictionaries, language learning apps, and social media platforms.
- * **Conclusions**: The study concludes that mobile devices can play a significant role in supporting learner autonomy and language learning, particularly in terms of providing access to online resources and facilitating

Doc Space:

The Doc Space section provides centralized access to all documents associated with a researcher's projects. This includes uploaded PDFs, generated summaries, extracted notes, and AI-generated analysis reports. Researchers can organize documents by project, search through them semantically, and reference them easily during research and writing activities.



CONCLUSION

ResearchHub AI represents a significant advancement in research paper management through the integration of agentic AI technology. By combining React and TypeScript frontend development, FastAPI backend architecture, and Groq's Llama 3.3 70B model, the platform delivers an intelligent solution that automates paper discovery, enabling researchers to focus on insights rather than information management. The agentic AI architecture enables autonomous reasoning and context-aware analysis, transforming how researchers interact with academic literature and extract meaningful knowledge from complex research domains. The platform's multi-agent capabilities—including semantic search through vector embeddings, conversational AI with persistent context, and autonomous paper analysis—address fundamental challenges in modern research workflows. ResearchHub AI exemplifies how agentic AI systems can revolutionize knowledge work by automating tedious tasks and empowering scholars to achieve deeper research insights at scale. Key achievements include autonomous research assistance that independently analyzes papers and synthesizes insights, semantic search capabilities that understand conceptual similarity beyond keyword matching, multi-workspace organization enabling parallel research projects, and a scalable architecture supporting concurrent users.