**A MINI-PROJECT REPORT**

**ON**

**“ARA”**

***Submitted to***

**Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur**

***in partial fulfillment of the requirement for the degree of***

***BACHELOR OF TECHNOLOGY***

***in***

***Computer Science & Engineering***

**Submitted By**

|  |  |
| --- | --- |
| **Kaustubh Warade** | **Aditya Deshmukh** |
| **Devansh Parapalli** | **Yashasvi Thool** |

**under the Guidance of**

***Dr. D. J. Chaudhary***

****

**Department of Computer Science & Engineering**

**Government College of Engineering**

**Nagpur – 441108 (M. S.)**

**2023-2024**

***Certificate***

***This is to certify that the mini-project entitled***

**“ARA” *Is a bonafide work and it is submitted to the***

***Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur.***

***By***

|  |  |
| --- | --- |
| **Kaustubh Warade** | **Aditya Deshmukh** |
| **Devansh Parapalli** | **Yashasvi Thool** |

***For the partial fulfillment of the requirement for the degree of***

***Bachelor of Technology in Computer Science & Engineering, during***

***the academic year 2023-2024.***

******

***Dr. Devchand Chaudhari Dr. Latest Bhagat***

***Project Guide Head of CSE Dept.***

**Department of Computer Science & Engineering.**

**Government College of Engineering,**

**Nagpur – 441108 (M. S.)**

**2023-2024**

**ACKNOWLEDGMENT**

It gives us immense pleasure in submitting the mini-project report on

**“ARA”**

to our guide **Dr. Devchand Jijiram Chaudhari**, and Head of Department **Dr. Latesh Gagan Malik,** Computer Science & Engineering Department, who were a constant source of guidance and inspiration through the seminar work.

We are also very thankful to all staff members of the Computer Science & Engineering department; whose encouragement and suggestions assisted us to complete the Mini-Project work.

We also express our sincere gratitude to our respected Principal **Dr. Rewatkumar Pithuji Borkar** for providing us necessary facilities.

At last, we are thankful to our friends whose encouragement and constant inspiration helped us to complete this seminar work verbally and theoretically.

Aditya Deshmukh  
Devansh Parapalli

Kaustubh Warade

Yashasvi Thool

(Third Year B.E.)

Computer Science & Engineering

Government College of Engg., Nagpur

**ABSTRACT**

*The exponential growth of research data and information sources has posed significant challenges for researchers, leading to information overload, disconnected insights, missed opportunities, and inefficiencies in organizing and synthesizing knowledge. To address these issues, we have developed ARA (Artificial Intelligence-powered Research Assistant), an innovative application that leverages advanced AI technologies to revolutionize the research process by enhancing information retrieval, analysis, and synthesis.*

*ARA represents a cutting-edge solution that transcends traditional note-taking approaches. It leverages large language models, semantic web technologies, and knowledge graphs to create a dynamic, interconnected web of research information that can be understood and processed by AI models.*

*ARA has delivered an advanced and efficient tool that enhances researchers' capabilities, streamlines their workflows, and contributes to the overall productivity of research activities. By fostering effective information retrieval and adaptability, ARA has the potential to drive groundbreaking discoveries and accelerate the pace of scientific progress across various disciplines.*

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  |  | **Page No.** |

**LIST OF FIGURES**

**LIST OF TABLE**

# INTRODUCTION

The modern research landscape is characterized by an ever-increasing deluge of information, posing significant challenges for researchers attempting to navigate this complex terrain. The exponential growth of research data, publications, and online resources has led to a phenomenon known as "information overload," where researchers struggle to manage and process the vast amounts of information available. Consequently, valuable insights and connections often remain hidden, opportunities for collaboration are overlooked, and the overall efficiency of the research process is hindered.

In response to these challenges, ARA (Artificial Intelligence-powered Research Assistant) has been developed, a revolutionary application that harnesses the power of advanced AI technologies to transform the way researchers approach information gathering, organization, and synthesis. ARA represents a paradigm shift from traditional note-taking applications, leveraging cutting-edge techniques such as large language models, knowledge graphs, and semantic web technologies to create a dynamic, interconnected web of research information that can be understood and processed by AI models.

The core objective of ARA is to empower researchers by providing a comprehensive and intelligent research assistant that streamlines their workflow, unveils hidden insights, expands research horizons, and sharpens critical thinking and communication skills. By automating routine tasks, accelerating data analysis, and providing contextual access to relevant information, ARA aims to significantly boost research productivity and efficiency, ultimately contributing to the acceleration of scientific progress across various disciplines.

# LITERATURE SURVEY

In the era of rapidly expanding scientific literature, automated methods have emerged as an asset in managing and streamlining the process. AI-based tools offer a myriad of features, including natural language processing, citation analysis, and automated summarization, enabling researchers to discover, evaluate, and organize relevant research papers efficiently. Given below are tools and platforms which were analyzed as part of the requirements of ARA.

## Lex[]

Lex learns from the text the user adds and uses it to generate new content. Academic and technical writing can be mentally draining, and Lex can take some of the grunt work. Lex helps the user focus on sharing their research findings rather than stressing about technical and scientific writing.

Lex only summarizes the content for the purposes of sharing and does not aid in the process of researching.

## Notion[]

Notion is a collaboration platform with Markdown[] and inducing kanban boards[], tasks, wikis[] and databases. It is a workspace for notetaking, knowledge and data management and project and task management.

Notion only provides rudimentary AI features which enable text prediction, generation and solving of simple arithmetic expressions. Notion cannot search for new sources.

## Elicit[]

Elicit is an AI assistant that uses language models to answer research questions. It can find relevant papers without needing perfect keyword matches, summarize key takeaways, and extract important information.

However, Elicit cannot search the web for generalized information, is not a good fit for identifying facts and theoretical or non-empirical domains.

## Research Rabbit[]

Research Rabbit is an AI-powered platform that assists researchers in discovering, visualizing, and analyzing relevant literature.

Research Rabbit cannot search for general information and does not transform text. The authors of Research Rabbit put it well, “Research Rabbit is like Spotify[] for Research”

## ChatPDF[]

ChatPDF is an AI-powered tool that functions as an interactive chatbot for PDF documents. It can answer queries, rewrite sections, and provide insights.

ChatPDF cannot utilize information beyond the provided PDF and hallucinates frequently.

## Consensus[]

Consensus is an AI-powered search engine that answers questions based on peer-reviewed literature, providing evidence-based answers and a “consensus meter” reflecting the state of current research.

Consensus does not possess the ability to generate summarizations or complete any custom task given by the user.

## IBM Watson[]

IBM Watson is a platform that offers various AI-powered tools for academic research, including data extraction, sentiment analysis, and language processing features to smoothen the research process and discover insights from unstructured data.

However, IBM Watson being a platform provides APIs and SDKs to integrate into your own software, and as such cannot be used by the general populace.

# ANALYSIS

In the due course of researching for our project, we have found the existence of various tools, platforms and components that have a common goal as ours.

## Existing Systems and Drawbacks.

**Lex** can generate new content by learning from provided text, but it does not aid in the research process itself by finding or summarizing sources.

**Notion** provides collaboration and knowledge management features with basic AI capabilities like text prediction but cannot search for or analyze research literature.

**Elicit** uses language models to find relevant papers and summarize key points but cannot search the general web or handle non-empirical domains.

**Research Rabbit** visualizes and analyzes literature but does not transform text or search beyond research papers.

**ChatPDF** allows querying and rewriting PDF documents through an AI chatbot but cannot utilize information beyond the given PDF.

**Consensus** provides evidence-based answers from peer-reviewed literature but lacks the ability to generate summarizations or handle custom tasks.

**IBM Watson** offers powerful AI capabilities for academic research like data extraction and language processing, it requires technical integration as a platform rather than an out-of-the-box solution.

## Proposed Application

The proposed ARA (Artificial Intelligence-powered Research Assistant) is an innovative application that aims to revolutionize the academic research process through the integration of cutting-edge artificial intelligence technologies. Departing from conventional note-taking applications, ARA will facilitate the creation of a dynamic, interconnected web of research data that can be comprehended and processed by AI models.

Key features to be incorporated in ARA include intelligent information retrieval mechanisms to gather data from diverse sources, contextual synthesis algorithms to amalgamate gathered information, effective organization frameworks to structure insights, and cross-disciplinary connection identification capabilities.

ARA is designed to address prevalent challenges faced by researchers, such as information overload, fragmented comprehension, overlooked collaboration opportunities, and inefficiencies in academic writing. By automating routine tasks, accelerating data analysis, and providing expedient access to pertinent information, ARA endeavors to substantially enhance research productivity.

Furthermore, it will leverage AI analysis to unveil latent insights, expand research horizons by recommending relevant advancements and potential collaborations, and refine critical thinking and communication proficiencies – ultimately driving innovation and facilitating groundbreaking discoveries within the academic research domain.

## Feasibility Study

A comprehensive feasibility study was conducted to assess the viability and potential impact of the proposed ARA application. This study examined the current challenges faced by researchers, evaluated existing solutions, and identified the critical requirements for an AI-powered research assistant. The study confirmed the pressing need for a tool that can effectively manage information overload, establish connections between disparate data sources, uncover new opportunities for collaboration, and enhance the communication of research findings.

An in-depth analysis of the latest advancements in artificial intelligence, particularly in areas such as natural language processing, knowledge representation, and machine learning, indicated that the necessary technological foundations are available to develop the envisioned ARA system. Furthermore, the study explored potential data sources, including open-access repositories, academic databases, and online knowledge bases, confirming the availability of sufficient data to train and power the AI models underpinning ARA.

The feasibility study also evaluated the project's resource requirements, including computational power, data storage and management infrastructure, and the integration of relevant AI frameworks and libraries. Based on the findings, the development of ARA was deemed feasible, with a high potential for transforming the research landscape and driving significant productivity gains for academic researchers across various disciplines.

## Technical Design Theory

The technical design of ARA will follow a modular and layered architecture, allowing for scalability, flexibility, and easy integration of various components. The core of the system will be built around state-of-the-art natural language processing (NLP) models and knowledge representation techniques. NLP models will be employed for intelligent information retrieval, content summarization, and text generation tasks. Semantic web technologies, such as Resource Description Framework (RDF) and Web Ontology Language (OWL), will be leveraged to represent and reason over the gathered knowledge, facilitating cross-disciplinary connections and insights.

The system will also incorporate machine learning algorithms for continuous learning and adaptation, ensuring that ARA's capabilities evolve with changing research needs and user feedback. Emphasis will be placed on developing robust security measures to protect sensitive research data and ensure compliance with privacy regulations.

# DESIGN

## Activity Diagram

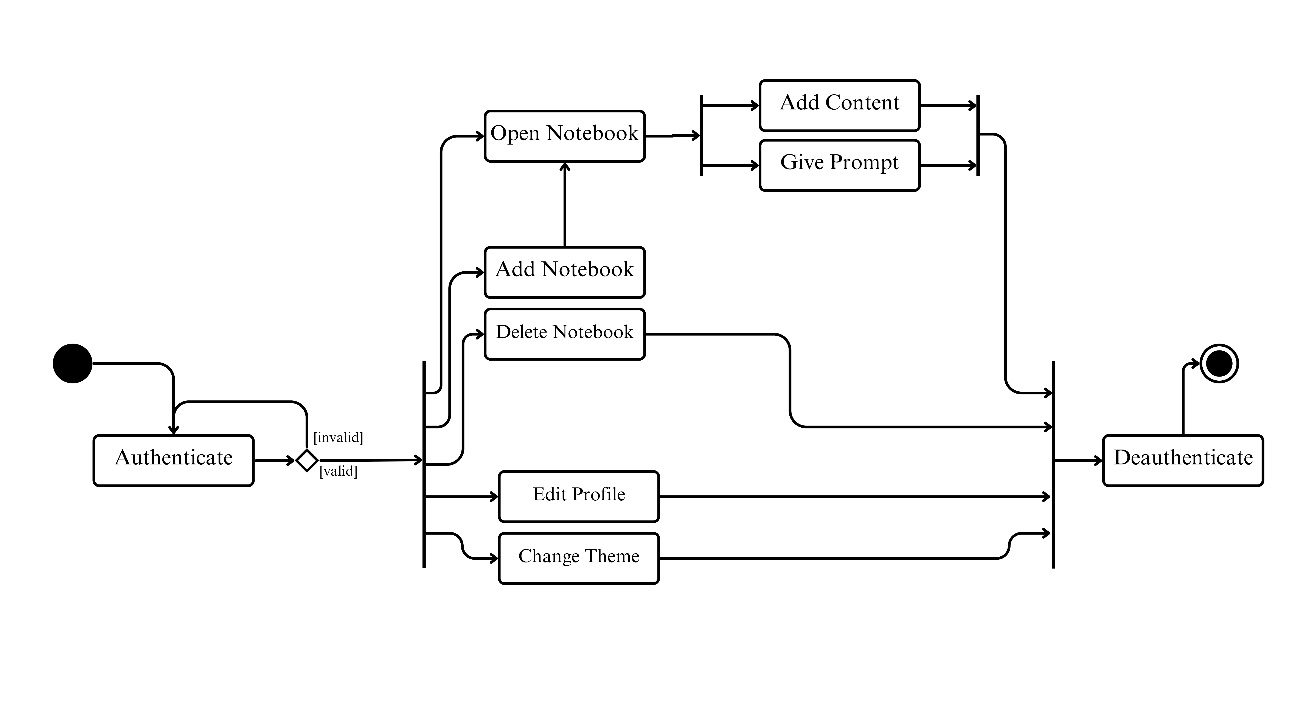


Figure 4.1: Activity Diagram

## Class Diagram

A diagram of a computer

Description automatically generated

Figure 4.2: Class Diagram

## Collaboration Diagram

A diagram of a process

Description automatically generated

Figure 4.3: Collaboration Diagram

## State Diagram

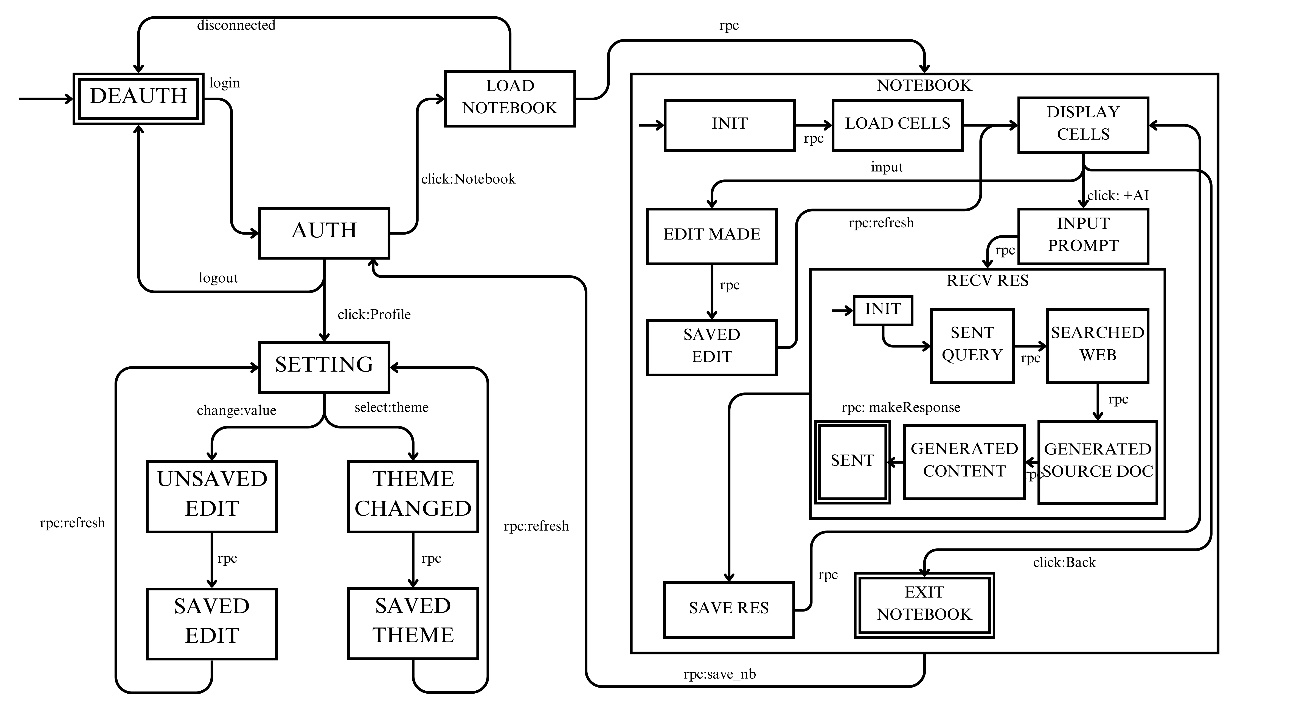


Figure 4.4: State Diagram

## Use Case Diagram

A diagram of a company

Description automatically generated

Figure 4.5: Use Case Diagram

# SYSTEM REQUIREMENTS

ARA is split into 4 components, each with their own set of requirements.

## Deployment

A centralized server is responsible for serving the assets to the user and hosting the Large Language Models along with handling communication between them.

### Hardware Requirements

ARA requires a moderately powerful computer system, capable of handling compilation and CI/CD tasks in parallel to running multiple threads for servicing requests.

Table 5.1: Hardware Requirements – Web Server

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Item** | **Specifications** | **Quantity** |
| 1. | Computer System | Intel(R) Core[TM] i7,  AMD Ryzen 5000 or better.  Nvidia(R)RTX[TM] 4080 or better.  64 GB DDR4 RAM 4 TB SSD Storage 2.5GBPS NIC | 1 |

### Software Requirements

The web server requires an advanced tech-stack for deployment due to the inclusion of Large Language Models.

Table 5.2: Software Requirements – Web Server

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Item** | **Version** |
| 1. | Operating System - Debian | 12.5 |
| 2. | Docker Engine | 26.0.0 |
| 3. | Docker Desktop | 4.28.0 |
| 4. | Docker Compose | 2.26.0 |
| 5. | Kubernetes | 1.29 |

## Development

ARA is built using state-of-the-art libraries and packages, including software still in beta-testing. ARA can be developed sufficiently well on a modern computer, built after 2016. A list of software used during development is given below.

Table 5.3: List of software used during development.

|  |  |
| --- | --- |
| **Sr. No.** | **Item** |
| 1. | Node.js |
| 2. | pnpm |
| 3. | Typescript |
| 4. | Svelte+SvelteKit |
| 5. | Supabase |
| 6. | PostCSS |
| 7. | TailwindCSS |
| 8. | Autoprefixer |
| 9. | Playwright |
| 10. | Vite |
| 11 | Vitest |
| 12 | Langchain.js |
| 13 | Katex |
| 14 | Markdown |
| 15 | Python |
| 16 | Ruff |
| 17 | Langchain.py |
| 18 | FastAPI |
| 19 | Numpy, Pandas, Scikit-learn |
| 20 | Uvicorn, |
| 21 | Mistletoe, |
| 22 | Magnum |
| 23 | Ujson |
| 24 | Gunicorn |
| 25 | Nginx |
| 26 | DaisyUI |

## User Requirements

Since ARA is built using a Client-Server model, the end user does not require very powerful computers. An end user must simply have a computer built after 2010, containing up to date browser software.

ARA is built using transpilation to ensure that it is functional on older systems. Since ARA is built using SvelteKit, the application can be server-rendered in a technique called Server Side Rendering[], eliminating the requirement for JavaScript to access the basic functions of the application.

# IMPLEMENTATION

TODO: Complete This

# SOFTWARE TESTING

Rigorous software testing is crucial for ensuring the reliability, functionality, and performance of the ARA system. A comprehensive testing strategy will be implemented throughout the development lifecycle, employing a range of testing techniques and methodologies.

## Components of Testing.

The testing phase will involve the following key components:

### Unit Testing

Individual units or components of the ARA system have undergone thorough unit testing to validate their functionality and behavior in isolation. These tests were automated using the Vitest package.

### Integration Testing

As individual units are integrated into larger subsystems, integration testing allows for identification and resolution of any interface issues, data inconsistencies, or compatibility problems that arise during the integration phase.

### System Testing

The entire system of ARA was subjected to comprehensive system testing to validate its end-to-end functionality and performance under realistic conditions. The test cases simulated a user browsing through the entire system, picking and prodding at it at every step. Such testing was conducted using a framework called Playwright.

### Acceptance Testing

Upon completion of the above testing paradigms, ARA was subjected to Acceptance Testing to validate the system’s rediness for deployment and conformance to the project’s acceptance criteria.

## Automated Test Cases

ARA incorporated a small set of automated, comprehensive test cases to validate the core functionality of the system.

Table 7.1: Automated End-To-End Test Cases

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Test Case** | **Result** |
| 1. | NotebookFunctionality | PASS |
| 2. | AICellPrompt | PASS |
| 3. | Login\_Register | PASS |
| 4. | Settings\_TestPage | PASS |

Unit testing rigorously validated core components like data transformations, state management, AI models, and UI elements to ensure ARA's reliability and maintainability.

Table 7.2: Automated Unit Test Cases

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Test Case** | **Result** |
| 1. | Model\_response | PASS |
| 2. | Sources\_order | PASS |
| 3. | No broken links | PASS |
| 4. | Citations received | PASS |
| 5. | Citation order correct | PASS |
| 6. | Local Models | PASS |
| 7. | Remote Models | PASS |
| 8. | Supabase (connection to supabase) | PASS |
| 9. | Func: convert\_cells\_to\_context | PASS |
| 10. | Func: getRelativeTime | PASS |

A screenshot of a computer

Description automatically generated

Figure 7.1: Automated End-to-End Test Case Result

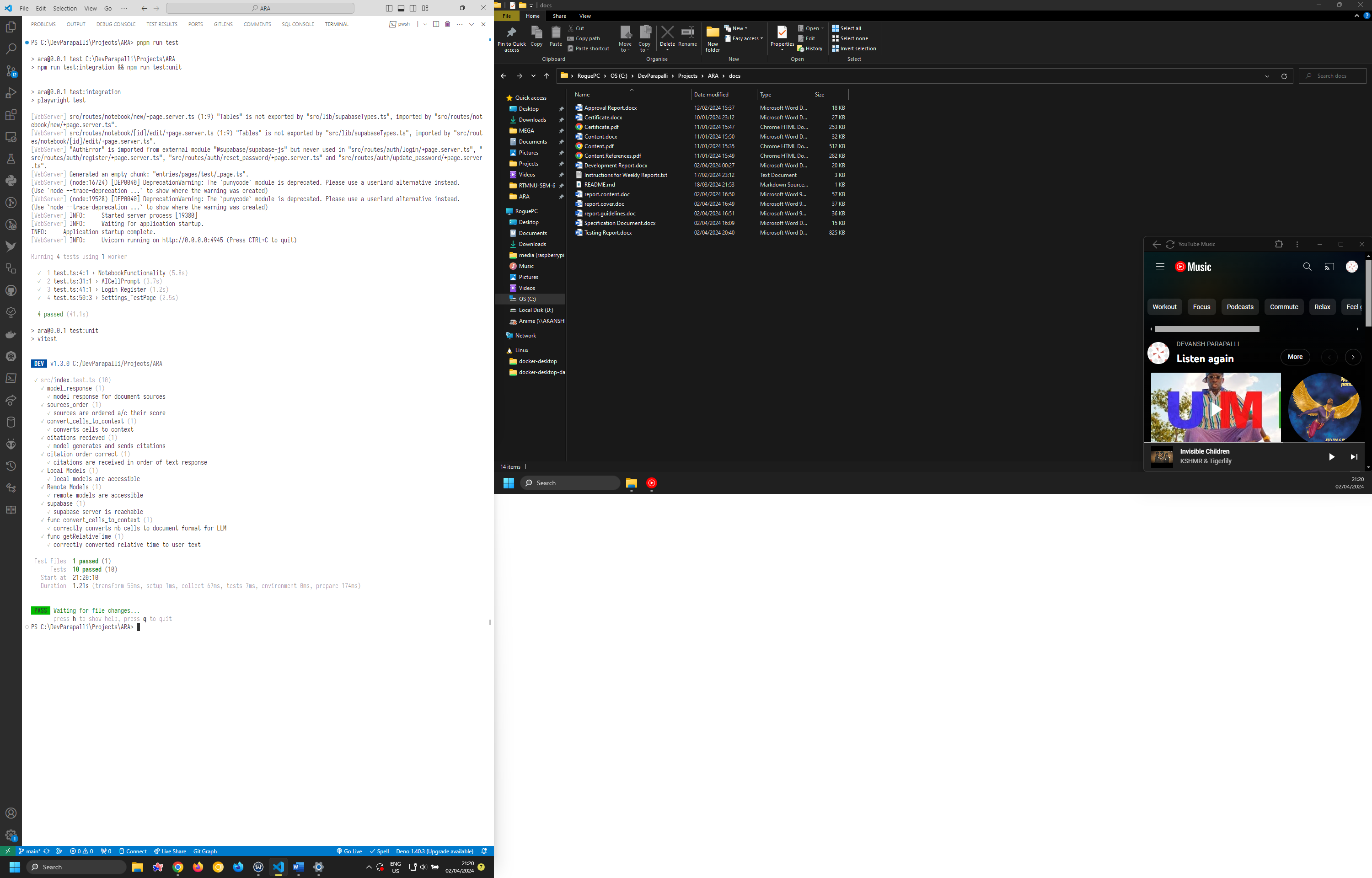
****

Figure 7.2: Automated Unit Test Case Result

## Manual Testing

While automated testing played a crucial role in validating the functionality and performance of ARA, extensive manual testing was also conducted to ensure a comprehensive evaluation of the system's capabilities and user experience.

Manual testing efforts focused on verifying the accuracy and relevance of the information retrieval, synthesis, and organization features. Testers explored a wide range of research topics, assessing the system's ability to gather data from diverse sources, establish meaningful connections, and present insights in a clear and comprehensible manner.

In addition to functional testing, manual testing encompassed an in-depth evaluation of the system's performance under various conditions, including large data sets, complex queries, and resource-intensive operations. Simulated scenarios pushed the system's limits, ensuring that ARA remained responsive, reliable, and scalable even under significant workloads.

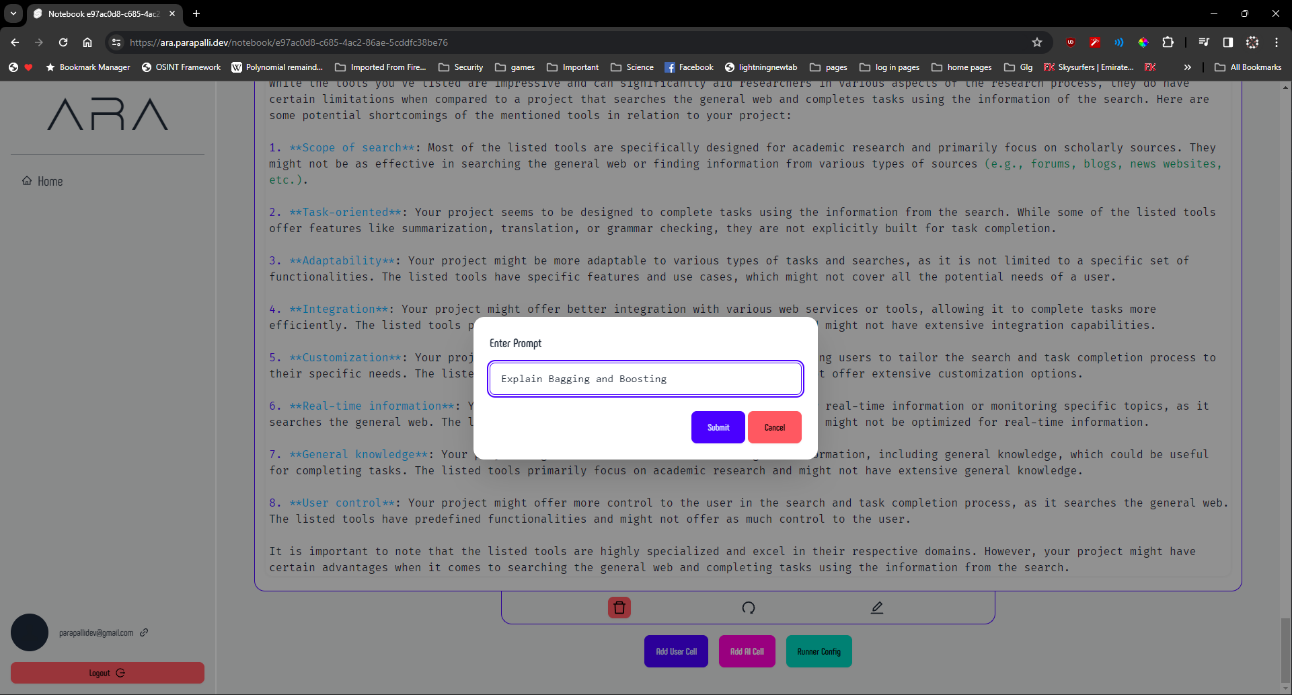


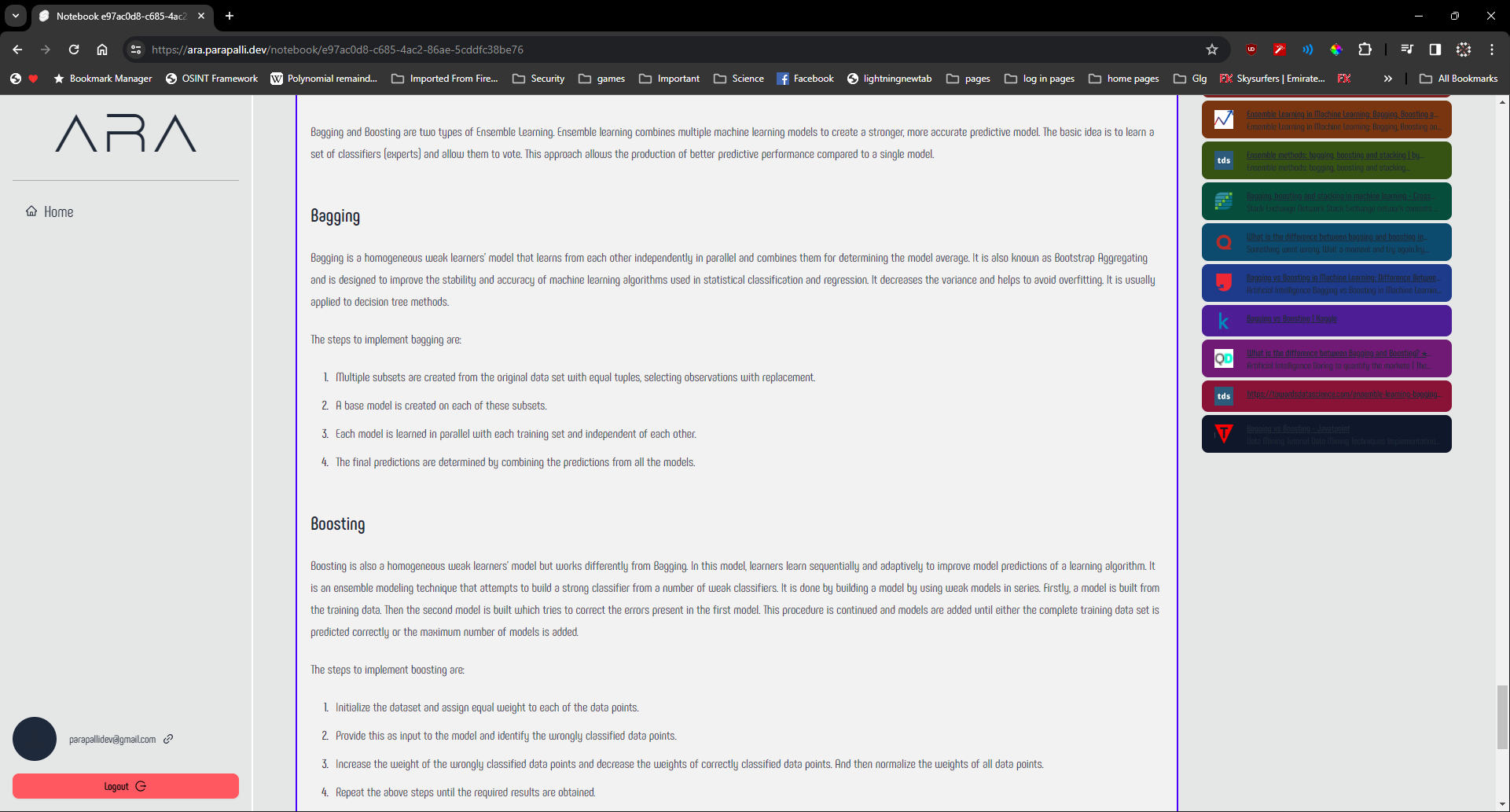
Figure 7.3: Manual Testing Prompt

Figure 7.4 Manual Testing Response

# RESULT DISCUSSION

Result Discussion

# APPLICATION

Application

# CONCLUSION

Conclusion

# REFERENCES

[Citation Number] …

# APPENDIX – A

Content for appendix-a