

Non Sticky AI Notes

Dev Dave¹, Yash Aswar², Anas Nadaf³, Rupankar Roy⁴, Mrs. Archana Jadhav⁵, Mrs. Rasika Kachore⁶,
Dr. Suvarna Patil⁷

^{1,2,3}*Student, Department of Artificial Intelligence and Data Science, Dr. D. Y. Patil Institute of Engineering Management and Research, Savitribai Phule Pune University, AKURDI, PUNE – 411044*

⁴*Project Guide, Department of Artificial Intelligence and Data Science, Dr. D. Y. Patil Institute of Engineering Management and Research, Savitribai Phule Pune University, AKURDI, PUNE – 411044*

⁵*Project Co-ordinator, Department of Artificial Intelligence and Data Science, Dr. D. Y. Patil Institute of Engineering Management and Research, Savitribai Phule Pune University, AKURDI, PUNE – 411044*

⁶*H.O.D.AI&DS, Department of Artificial Intelligence and Data Science, Dr. D. Y. Patil Institute of Engineering Management and Research, Savitribai Phule Pune University, AKURDI, PUNE – 411044*

1. INTRODUCTION

1.1 OVERVIEW

In today's fast-paced digital world, effective note-taking is indispensable for capturing and retaining information. Traditional note-taking methods often fall short in meeting the demands of modern users, who require efficient, accessible, and adaptable solutions. Recognizing these challenges, our project introduces "Non Sticky AI Notes," an innovative approach to note-taking empowered by artificial intelligence (AI) technologies. Non Sticky AI Notes aims to revolutionize the way users interact with and manage their notes, offering a seamless and intelligent solution for organizing, retrieving, and synthesizing information.

1.2 MOTIVATION

The motivation behind the development of Non Sticky AI Notes stems from the shortcomings observed in conventional note-taking practices and existing digital note-taking applications. Traditional methods, such as handwritten notes or basic digital text documents, lack the intelligence and adaptability needed to keep pace with modern information management requirements. Furthermore, existing note-taking applications often suffer from limitations such as manual organization, limited search capabilities, and insufficient support for knowledge synthesis.

The exponential growth of digital information and the increasing complexity of tasks demand a smarter

approach to note-taking. Users need tools that can not only capture information but also help them make sense of it, connect related concepts, and facilitate knowledge retention and retrieval. Non Sticky AI Notes addresses these needs by leveraging AI algorithms to enhance the note-taking experience. By harnessing the power of machine learning, natural language processing, and data analytics, Non Sticky AI Notes offers intelligent features such as automatic categorization, context-aware search, and personalized recommendations.

Furthermore, the project is driven by the desire to explore the intersection of AI and human-computer interaction (HCI) in the context of note-taking. By integrating AI seamlessly into the note-taking process, The aim is to create a symbiotic relationship between users and technology, where AI augments human cognitive abilities and empowers users to manage information more effectively.

In summary, the motivation behind Non Sticky AI Notes is to bridge the gap between traditional note-taking methods and the evolving needs of users in the digital age. By harnessing the capabilities of AI, our project aspires to elevate the note-taking experience, making it more intuitive, efficient, and impactful for users across various domains and disciplines.

1.3 PROBLEM STATEMENT AND OBJECTIVE

Problem Statement:

Traditional note-taking methods, both analog and digital, often prove to be inadequate in addressing the evolving needs of users in the digital age. Handwritten notes can be cumbersome to organize and search through, while basic digital text documents lack intelligent features for efficient information management. Existing digital note-taking applications, although more convenient, often fall short in providing robust solutions for note organization, retrieval, and synthesis. Users face challenges such as manual categorization, limited search capabilities, and difficulty in extracting insights from their notes.

Objectives:

- o Develop an intelligent note taking system
- o Automatic Categorization
- o Context Aware Search
- o Personalized Recommendations
- o Knowledge Synthesis Features
- o User Feature Interface

1.4 PROJECT SCOPE

Project Scope:

The project scope begins with an introduction to Non Sticky AI Notes, providing an overview of its objectives and key features. It sets the stage for understanding the scope of the project by highlighting the significance of introducing an AI-powered note-taking solution to address the limitations of traditional methods and existing digital applications.

The project aims to leverage artificial intelligence (AI) technologies to enhance various aspects of the note-taking process. This includes implementing machine learning algorithms for automatic categorization of notes based on content analysis, integrating advanced natural language processing (NLP) techniques for context-aware search functionality, and deploying recommendation systems to provide personalized suggestions for related notes or topics.

An essential aspect of the project scope is ensuring a seamless and intuitive user experience. This involves designing a user-friendly interface that accommodates users of all skill levels and preferences. Additionally, the project encompasses user interaction design considerations to optimize usability and accessibility,

thereby enhancing user satisfaction and adoption.

Another critical aspect of the project scope is scalability and performance optimization. As the user base grows and the volume of data increases, the system must be capable of handling large-scale note-taking activities efficiently. This entails implementing robust backend infrastructure, optimizing algorithms for computational efficiency, and employing techniques for data management and storage. Non Sticky AI Notes is envisioned as a versatile tool that seamlessly integrates with existing digital ecosystems and workflows. Therefore, the project scope includes compatibility considerations for various platforms and devices, such as desktop computers, mobile devices, and web browsers. Integration with popular productivity tools and cloud services may also be explored to enhance interoperability and user convenience.

Given the sensitive nature of personal and professional information stored in notes, security and privacy are paramount concerns. The project scope encompasses implementing robust security measures to protect user data from unauthorized access, data breaches, and other security threats. This includes encryption techniques, access control mechanisms, and adherence to industry standards and regulations regarding data privacy. Ensuring the reliability and quality of Non Sticky AI Notes is a fundamental aspect of the project scope. This involves comprehensive testing procedures to validate the functionality, performance, and usability of the system. Testing methodologies may include unit testing, integration testing, user acceptance testing, and performance testing to identify and address any issues or deficiencies.

Finally, the project scope includes the development of comprehensive documentation and support resources to assist users in effectively utilizing Non Sticky AI Notes. This encompasses user manuals, tutorials, troubleshooting guides, and online help resources to address common queries and issues. Additionally, ongoing technical support and maintenance services may be provided to ensure the continued functionality and reliability of the system.

2. LITERATURE SURVEY

Models	Algorithms Description	Features	Limitations
Machine Learning for Text Classification	An approach within ML which combines models to form a robust model capable of making prediction with maximum accuracy.	These algorithms are capable of analysing sticky notes' text content and categorizing them into pre-established tags or categories. Large amounts of text data may be handled by them, and they can be trained to identify intricate correlations and patterns in the text.	Problems with over fitting and generalization can also occur, particularly when there is insufficient or skewed training data.
Generative Models for Content Generation	A discriminator and a generator compete with one another in GANs to generate realistic material.	Based on pre-existing material, generative models can provide fresh and varied sticky note content, such as summaries, recommendations, or creative impulses. They provide versatility and flexibility in producing content that is suited to the requirements and tastes of users.	Mode collapse, or the inability to fully represent the diversity of the input data distribution, is a potential problem for generative models. Large volumes of training data and processing power are needed for them. Furthermore, there may be instances where the material created is <u>neither cohesive or contextually appropriate</u> , which makes it difficult to employ in practical

			applications.
Reinforcement Learning (RL) for Personalization	The agent adjusts its behaviour to individual user preferences and objectives by streamlining its decision-making process to maximize cumulative rewards.	Adaptability, Dynamic Learning, Personalized Experience	It necessitates giving data, exploration-exploitation trade offs, and scalability issues considerable thought.
Gaussian Naïve Bayes(GNB)	In a single pass over the picture, it makes predictions using a single neural network.	Real-time detection, Unified approach, High accuracy.	Relies on the assumption of feature independence, which may not align with real-world datasets. Performance may be subpar when dealing with data containing correlated features.
YOLO (You Only Look Once)	In a single pass over the picture, it makes predictions using a single neural network.	Real-time detection, Unified approach, High accuracy	Small object detection, Localization errors, Limited scalability

3. SOFTWARE REQUIREMENT SPECIFICATION

3.1 Assumptions and Dependencies

In the Software Requirement Specification (SRS) for Non Sticky AI Notes, several assumptions and dependencies are acknowledged to ensure clarity and understanding regarding the project's development and implementation. These assumptions and dependencies serve as foundational elements that influence various aspects of the project lifecycle,

including design, development, testing, and deployment.

Firstly, it is assumed that the target user base for Non Sticky AI Notes comprises individuals from diverse backgrounds, including students, professionals, researchers, and knowledge workers. This assumption influences the design and feature set of the application, ensuring that it caters to a wide range of user needs and preferences.

Furthermore, the project relies on the availability and accessibility of external APIs and services, particularly those provided by OpenAI for AI

capabilities such as natural language processing and text generation. Dependencies on third-party APIs necessitate proper integration and adherence to usage guidelines to ensure the seamless functioning of the application. Additionally, it is assumed that Non Sticky AI Notes will be developed using modern software development methodologies and best practices. This includes employing agile development principles, version control systems (such as Git), and continuous integration/continuous deployment (CI/CD) pipelines to facilitate collaboration, code management, and deployment automation.

Moreover, the project's success hinges on the availability of adequate hardware and infrastructure resources to support development, testing, and deployment activities. This includes access to computing resources for training AI models, hosting web servers for backend operations, and ensuring sufficient network bandwidth for data transfer. Another critical assumption is that Non Sticky AI Notes will adhere to industry-standard security and privacy practices to safeguard user data and maintain confidentiality. This includes implementing encryption mechanisms, access control measures, and regular security audits to mitigate potential risks and vulnerabilities.

Furthermore, the project's timeline and deliverables are subject to various external factors, including resource availability, technological advancements, and regulatory compliance. Dependencies on these external factors may impact project milestones and necessitate adjustments to project planning and execution.

In conclusion, the Software Requirement Specification for Non Sticky AI Notes acknowledges several assumptions and dependencies that influence the project's development and implementation. By addressing these assumptions and dependencies proactively, the project aims to ensure clarity, transparency, and alignment with stakeholder expectations throughout the project lifecycle.

3.2 FUNCTIONAL REQUIRMENTS

3.2.1. User Authentication and Authorization:

Users must be able to register new accounts and log in securely with existing credentials.

Users should have the ability to create new notes, edit existing notes, and format text within notes.

3.2.2. Automatic Categorization:

The system must automatically categorize notes based on their content, organizing them into predefined or user-defined categories.

3.2.3. Context-Aware Search:

Users should be able to search within their notes using keywords, with results tailored to the context of the search query.

3.2.4. Personalized Recommendations:

The system must provide personalized recommendations for related notes or topics based on user interactions and content similarity.

3.2.5. Knowledge Synthesis Features:

Users should be able to synthesize information across multiple notes, linking related content and generating summaries.

3.2.6. User Interface and Interaction:

The user interface should be intuitive and responsive, supporting features such as drag-and-drop, keyboard shortcuts, and context menus.

3.2.7. Integration with External Services:

Non Sticky AI Notes should integrate with external services and APIs, such as OpenAI and cloud storage providers, for enhanced functionality and data synchronization.

3.2.8. Collaboration and Sharing:

Users must be able to collaborate on notes with others in real-time, with options to share notes and set access permissions.

3.2.9. Data Management and Security:

The system must ensure the security and privacy of user data, including encryption, backup, and access control mechanisms.

3.3 EXTERNAL INTERFACE REQUIREMENTS

3.3.1 USER INTERFACES

Web Interface:

- Non Sticky AI Notes will have a web-based user interface accessible through modern web browsers.
- The interface will feature a dashboard displaying notes, options for creating, editing, and organizing

notes, as well as search and recommendation functionalities.

- User-friendly components such as text editors, dropdown menus, and buttons will facilitate interaction.

Mobile Interface:

- A mobile-friendly version of Non Sticky AI Notes will be developed to provide users with access on smartphones and tablets.
- The mobile interface will prioritize responsiveness and usability on smaller screens, with optimized layouts and touch-friendly controls.

3.3.2 HARDWARE INTERFACES

Computing Devices:

- Non Sticky AI Notes will be compatible with a range of computing devices, including desktop computers, laptops, tablets, and smartphones.
- The application will leverage the computing power and resources of these devices to provide a responsive and efficient user experience.

Input Devices:

- The application will support various input devices such as keyboards, mice, touchscreens, and styluses.
- Users will be able to interact with the application using these input devices to create, edit, and navigate through notes.

3.3.3 SOFTWARE INTERFACES

OpenAI API:

Non Sticky AI Notes will integrate with the OpenAI API to leverage AI capabilities such as natural language processing and text generation.

The application will interact with the API to perform tasks such as content analysis, recommendation generation, and context-aware search.

Database Interface:

Non Sticky AI Notes will interact with a backend database system to store and retrieve user data, including notes, user profiles, and interaction logs.

The application will utilize database queries and transactions to manage data efficiently and securely.

Third-Party Integrations:

Non Sticky AI Notes may integrate with third-party

services and APIs, such as cloud storage providers (e.g., Google Drive, Dropbox) for data synchronization.

The application will communicate with these external interfaces to enable features such as importing/exporting notes and accessing external resources.

3.3.4 COMMUNICATION INTERFACES

Non Sticky AI Notes relies on robust communication interfaces to facilitate seamless interaction between its components and external systems. These interfaces encompass various protocols and technologies that enable data exchange and interoperability. Internally, communication interfaces ensure smooth interaction between the frontend and backend components of the application, enabling real-time updates, data synchronization, and user interactions. Externally, Non Sticky AI Notes integrates with external APIs and services, such as the OpenAI API and cloud storage providers, through standardized communication protocols. These interfaces enable the application to leverage advanced AI capabilities, access external resources, and enhance functionality while maintaining compatibility and interoperability with external systems. By leveraging effective communication interfaces, Non Sticky AI Notes delivers a cohesive and integrated user experience while seamlessly integrating with external services to augment its capabilities and extend its functionality.

3.4 NON-FUNCTIONAL REQUIREMENTS

3.4.1. Performance:

The system should respond to user interactions promptly, with minimal latency.

It should be capable of handling simultaneous user interactions without significant degradation in performance. The application should be able to process large volumes of data efficiently, including note storage and retrieval operations.

3.4.2. Scalability:

The system should be scalable to accommodate growing user bases and increasing data loads.

It should support horizontal scaling by distributing workload across multiple servers or instances as demand increases.

3.4.3. Reliability:

Non Sticky AI Notes should be highly reliable, with minimal downtime and service interruptions.

It should have built-in mechanisms for fault tolerance and error recovery to ensure continuous availability.

3.4.4. Security:

The system must adhere to industry-standard security practices to protect user data and ensure confidentiality. It should implement robust authentication and authorization mechanisms to prevent unauthorized access to sensitive information. Encryption should be employed for data transmission and storage to safeguard against eavesdropping and data breaches.

3.4.5. Usability:

The user interface should be intuitive and easy to navigate, requiring minimal training for users to effectively utilize the application. Accessibility features should be implemented to ensure usability for users with disabilities, in compliance with accessibility standards.

3.4.6. Compatibility:

Non Sticky AI Notes should be compatible with a wide range of web browsers, operating systems, and devices to ensure accessibility for all users. It should adhere to web standards and best practices to maintain compatibility with evolving technologies and platforms.

3.4.7. Maintainability:

The system should be designed with modularity and extensibility in mind to facilitate ease of maintenance and future enhancements. Code documentation and comments should be provided to aid developers in understanding and modifying the system.

3.4.8. Performance:

The system should be capable of handling concurrent user requests efficiently, ensuring responsiveness under load. Performance benchmarks should be established and monitored to identify and address performance bottlenecks proactively.

3.4.9. Regulatory Compliance:

Non Sticky AI Notes should comply with relevant laws, regulations, and industry standards governing data privacy and security, such as GDPR, HIPAA, or SOC 2. Regular audits and assessments should be conducted to ensure ongoing compliance with regulatory requirements.

3.4.10. Backup and Recovery:

The system should implement regular data backups and recovery procedures to prevent data loss and minimize downtime in the event of system failures or disasters. Backup data should be stored securely and

be readily accessible for restoration purposes.

3.5 SYSTEM REQUIREMENTS:

System requirements for Non Sticky AI Notes encompass hardware, software, and network components necessary for the installation, operation, and performance of the application. Here's a breakdown of system requirements:

Hardware Requirements:

1. Processor (CPU):

- Minimum: Dual-core processor
- Recommended: Quad-core processor or higher

2. Memory (RAM):

- Minimum: 4 GB RAM
- Recommended: 8 GB RAM or higher

3. Storage:

- Minimum: 100 MB available disk space for application installation
- Recommended: SSD storage for improved performance, additional storage for user data

4. Display:

- Minimum: Standard display with a resolution of 1280x720 pixels
- Recommended: Full HD display with a resolution of 1920x1080 pixels or higher

5. Input Devices:

- Keyboard and mouse (or touchpad) for desktop/laptop computers
- Touchscreen for tablets and smartphones

Software Requirements:

1. Operating System:

- Windows 10 or later
- macOS 10.13 or later
- Linux distributions (Ubuntu, Fedora, etc.) with recent updates
- Android 6.0 (Marshmallow) or later for mobile devices
- iOS 12 or later for iPhones and iPads

2. Web Browser:

- Google Chrome (latest version)
- Mozilla Firefox (latest version)
- Apple Safari (latest version)
- Microsoft Edge (latest version)
- Note: Ensure that browsers are up-to-date to support modern web technologies and features.

3. Backend Framework:

- Django web framework (version 3.0 or later) for server-side development
- 4. Database Management System:
 - PostgreSQL (version 9.6 or later) for data storage and retrieval
- 5. Python:
 - Python programming language (version 3.6 or later) for backend development
- 6. JavaScript Framework:
 - React JS (version 16.0 or later) for frontend development
- 7. Additional Libraries and Dependencies:
 - OpenAI API (for AI capabilities)
 - Required Python packages and dependencies as specified in project documentation.

Network Requirements:

1. Internet Connectivity:
 - Stable internet connection with sufficient bandwidth for data transfer
 - Recommended minimum speed: 5 Mbps (download/upload)
2. Firewall and Network Security:
 - Ensure that firewall settings allow access to required ports for application communication (e.g., HTTP/HTTPS)
3. DNS Resolution:
 - Reliable DNS resolution for domain name resolution and access to web services.

3.6 ANALYSIS MODELS : SDLC MODEL TO BE APPLIED :

The following figure 3.6 is the SDLC dig. of project implementation.

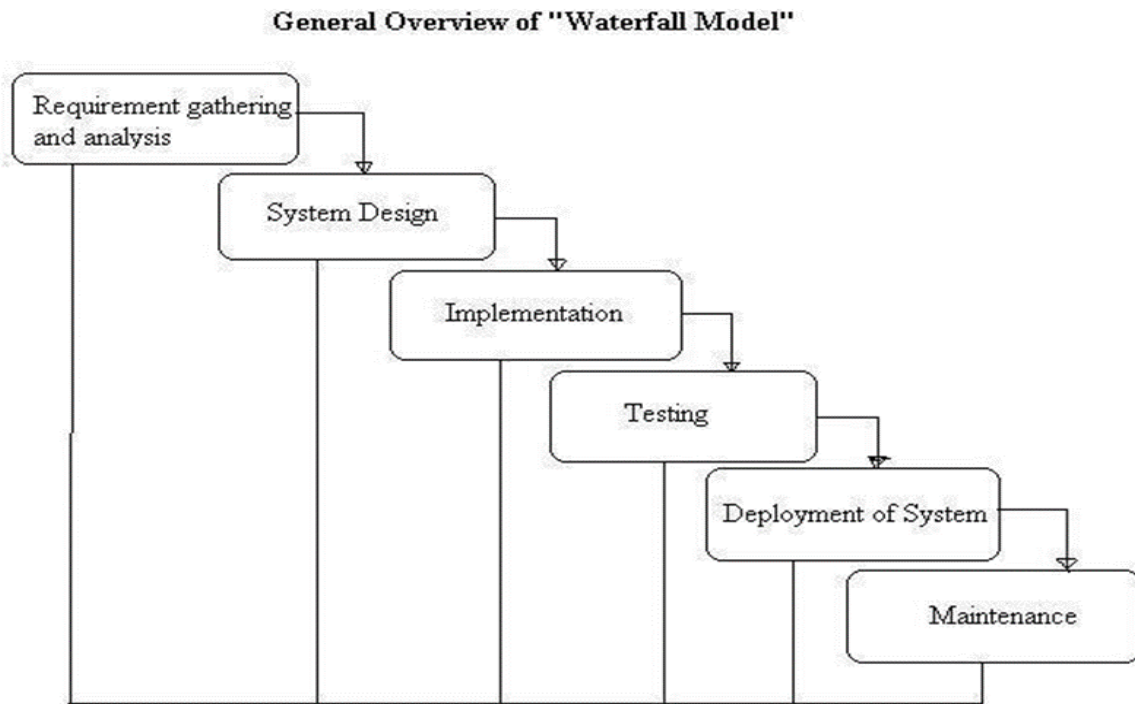


Figure 3.6 : Software Development Model

3.7 SYSTEM IMPLEMENTATION PLAN :

The system implementation plan outlines the steps and procedures for deploying Non Sticky AI Notes, ensuring a smooth transition from development to production. Here's a detailed implementation plan:

3.7.1. Preparation Phase:

- Assessment of Requirements: Review system requirements, including hardware, software, and network prerequisites.
- Infrastructure Setup: Provision necessary hardware resources (servers, storage) and install required software components (operating system, database, web server).

- Environment Configuration: Set up development, staging, and production environments, ensuring consistency across environments.

3.7.2. Development Phase:

- Backend Development: Develop backend components using Django framework, including user authentication, note management, AI integration, and database interactions.
- Frontend Development: Design and implement user interfaces using React JS, focusing on usability, responsiveness, and accessibility.
- Integration Testing: Conduct integration testing to ensure seamless communication between frontend and backend components, identifying and resolving any issues.

3.7.3. AI Integration Phase:

- OpenAI Integration: Integrate with OpenAI API to leverage AI capabilities for automatic categorization, context-aware search, and personalized recommendations.
- Algorithm Development: Develop and train machine learning models for note categorization, recommendation generation, and knowledge synthesis, ensuring accuracy and performance.

3.7.4. Testing Phase:

- Unit Testing: Perform unit tests for individual components to validate functionality and identify bugs or inconsistencies.
- System Testing: Conduct comprehensive system testing to evaluate overall system behaviour, including functional, non-functional, and user acceptance testing.
- Performance Testing: Assess system performance under various load conditions, optimizing resource utilization and identifying potential bottlenecks.

3.7.5. Deployment Phase:

- Deployment Plan: Develop a deployment plan outlining deployment steps, rollback procedures,

and contingency measures.

- Deployment to Staging Environment: Deploy the application to the staging environment for final testing and validation in a production-like environment.

- User Acceptance Testing (UAT): Invite stakeholders and end users to participate in UAT, soliciting feedback and verifying system readiness for production deployment.

3.7.6. Production Deployment:

- Production Rollout: Deploy the application to the production environment following the deployment plan, ensuring minimal downtime and service disruptions.

- Monitoring and Maintenance: Implement monitoring tools and processes to monitor system performance, availability, and security post-deployment.

- Ongoing Maintenance: Establish procedures for ongoing maintenance, including software updates, security patches, and database backups.

3.7.7. Training and Documentation:

- User Training: Provide training sessions for end users to familiarize them with Non Sticky AI Notes, highlighting key features and functionalities.

- Documentation: Prepare comprehensive documentation covering system architecture, installation instructions, user guides, and troubleshooting procedures.

3.7.8. Post-Implementation Review:

- Post-Implementation Review: Conduct a post-implementation review to evaluate the success of the deployment, identify lessons learned, and gather feedback for future improvements.

- Continuous Improvement: Implement feedback and lessons learned to refine processes, enhance system performance, and address any remaining issues.

4 SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE:

The following figure 4.1 is the system architecture dig. of project implementation.

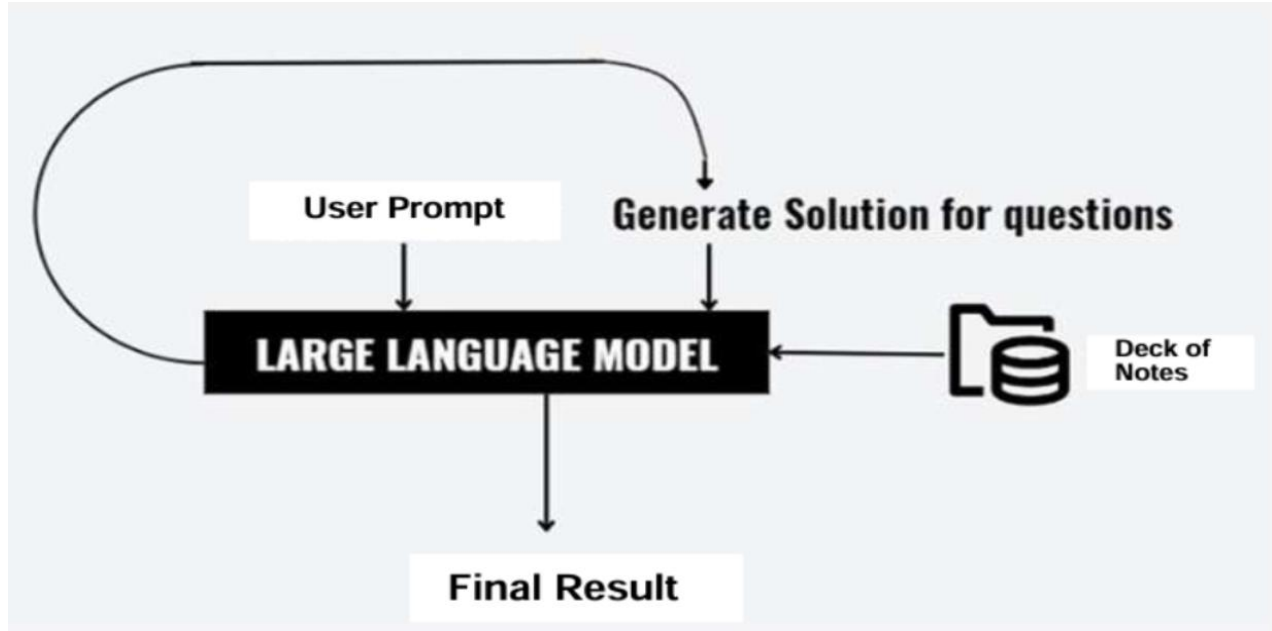


Figure 4.1 : System Architecture Figure

4.2 UML DIAGRAM

The following figure 4.2 is the UML dig. of project implementation

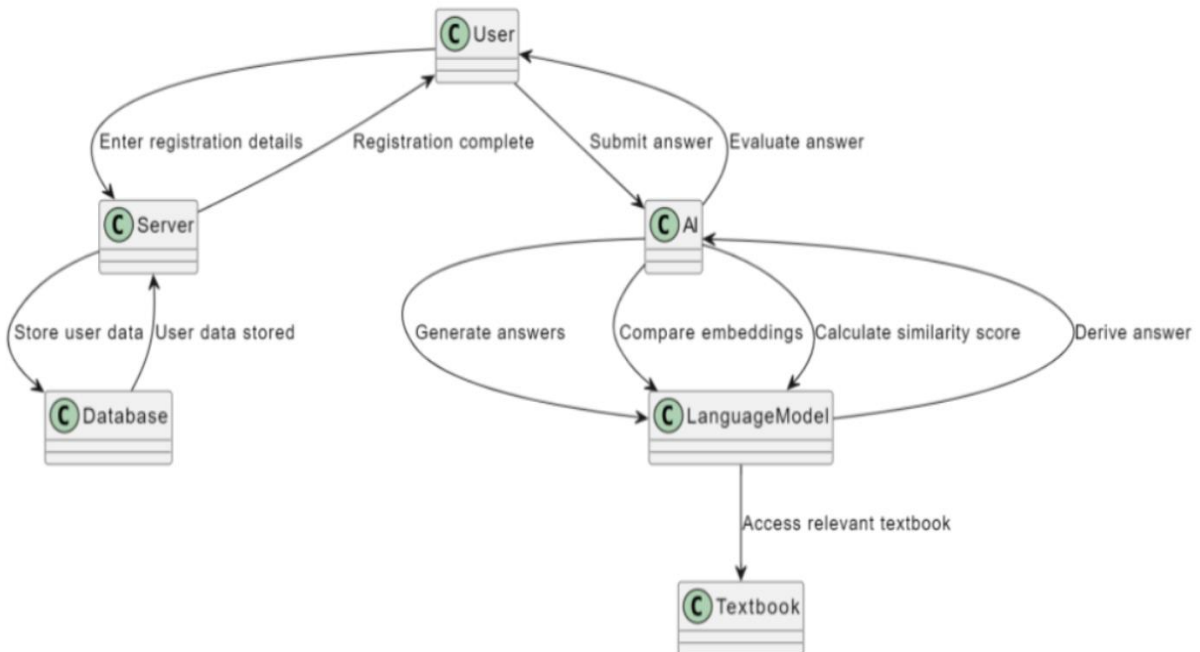


Figure 4.2 : UML Diagram

4.3 ENTITY RELATIONSHIP DIAGRAM

The following figure 4.3 is the ER dig. of project implementation.

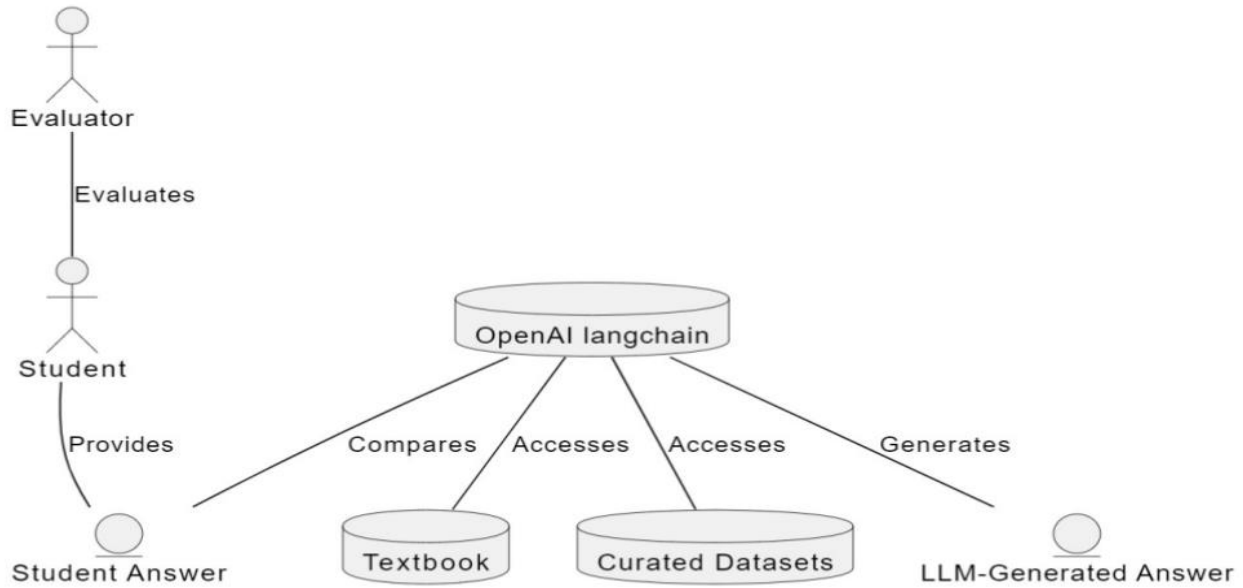


Figure 4.3 : ER Diagram

5. Project Plan

5.1. Project Estimate

1. Initiation Phase (Month 1):

- Define project scope, objectives, and deliverables.
- Identify stakeholders and establish communication channels.
- Develop project charter and obtain approvals.
- Set up project management tools and infrastructure.

2. Planning Phase (Month 2-3):

- Conduct detailed requirements analysis, including functional and non-functional requirements.
- Create project plan, including schedule, budget, resource allocation, and risk management.
- Define project milestones, deliverables, and success criteria.
- Establish project governance and communication protocols.

3. Development Phase (Month 4-7):

- Backend Development:
 - Develop backend components using Django framework.
 - Implement user authentication, note management, and database interactions.
- Frontend Development:
 - Design and implement user interfaces using React JS.

- Focus on usability, responsiveness, and accessibility.
- AI Integration:
 - Integrate with OpenAI API for AI capabilities.
 - Develop and train machine learning models for note categorization and recommendation generation.

4. Testing Phase (Month 8-9):

- Conduct comprehensive testing, including unit testing, integration testing, system testing, and performance testing.
- Address any issues or bugs identified during testing.
- Perform user acceptance testing (UAT) to validate system functionality and usability.
- Finalize documentation and training materials.

5. Deployment Phase (Month 10):

- Deploy the application to the production environment following the deployment plan.
- Monitor system performance, availability, and security post-deployment.
- Conduct user training sessions to familiarize users with Non Sticky AI Notes.
- Conduct a post-implementation review to evaluate project success and gather feedback for future improvements.

5.2. Project Schedule

1. Initiation Phase (Month 1):

- Define project scope, objectives, and deliverables.
- Identify stakeholders and establish communication channels.
- Develop project charter and obtain approvals.
- Set up project management tools and infrastructure.

2. Planning Phase (Month 2-3):

- Conduct detailed requirements analysis, including functional and non-functional requirements.
- Create project plan, including schedule, budget, resource allocation, and risk management.
- Define project milestones, deliverables, and success criteria.
- Establish project governance and communication protocols.

3. Development Phase (Month 4-7):

- Backend Development:
 - Develop backend components using Django framework.
 - Implement user authentication, note management, and database interactions.
- Frontend Development:
 - Design and implement user interfaces using React JS.
 - Focus on usability, responsiveness, and accessibility.
- AI Integration:
 - Integrate with OpenAI API for AI capabilities.
 - Develop and train machine learning models for note categorization and recommendation generation.

4. Testing Phase (Month 8-9):

- Unit Testing:
 - Develop and execute unit tests for individual components.
 - Verify functionality and identify bugs or inconsistencies.
- Integration Testing:
 - Conduct integration testing to ensure seamless communication between frontend and backend components.
 - Validate data flow and system behavior under different scenarios.
- System Testing:
 - Perform comprehensive system testing to evaluate overall functionality, including user

interfaces, AI integration, and database interactions.

- Performance Testing:
 - Assess system performance under various load conditions.
 - Optimize resource utilization and identify potential bottlenecks.

5. Deployment Phase (Month 10):

- Deployment Plan:
 - Develop a deployment plan outlining deployment steps, rollback procedures, and contingency measures.
- Deployment to Staging Environment:
 - Deploy the application to the staging environment for final testing and validation.
- User Acceptance Testing (UAT):
 - Invite stakeholders and end users to participate in UAT.
 - Solicit feedback and verify system readiness for production deployment.
- Production Rollout:
 - Deploy the application to the production environment following the deployment plan.
 - Monitor system performance, availability, and security post-deployment.

6. Documentation and Training (Month 10):

- Documentation:
 - Prepare comprehensive documentation covering system architecture, installation instructions, user guides, and troubleshooting procedures.
- User Training:
 - Conduct user training sessions to familiarize users with Non Sticky AI Notes.
 - Provide guidance on using the application effectively and efficiently.

7. Post-Implementation Review (Month 10):

- Post-Implementation Review:
 - Evaluate the success of the deployment and gather feedback from stakeholders.
 - Identify lessons learned and areas for improvement.
 - Incorporate feedback and lessons learned into future projects and initiatives.

5.3. Team Organization

1. Project Manager:

- Responsible for overall project planning, execution, and coordination.

- Manages project timeline, budget, resources, and risks.
 - Facilitates communication among team members and stakeholders.
2. Backend Development Team:
 - Backend Developers (2-3 members):
 - Develop backend components using Django framework.
 - Implement user authentication, note management, and database interactions.
 3. Frontend Development Team:
 - Frontend Developers (2-3 members):
 - Design and implement user interfaces using React JS.
 - Focus on usability, responsiveness, and accessibility.
 4. AI Integration Team:
 - AI Engineers/Data Scientists (1-2 members):
 - Integrate with OpenAI API for AI capabilities.
 - Develop and train machine learning models for note categorization and recommendation generation.
 5. Quality Assurance (QA) Team:
 - QA Engineers (1-2 members):
 - Conduct comprehensive testing, including unit testing, integration testing, system testing, and performance testing.
 - Address any issues or bugs identified during testing.
 6. Documentation and Training Team:
 - Technical Writers (1 member):
 - Prepare documentation covering system architecture, installation instructions, user guides, and troubleshooting procedures.
 - Training Facilitator (1 member):
 - Conduct user training sessions to familiarize users with Non Sticky AI Notes.
 7. Project Support:
 - IT Support (1 member):
 - Provide technical support and assistance during deployment and post-deployment phases.
 - Project Coordinator (1 member):
 - Assist the project manager in administrative tasks, scheduling meetings, and maintaining project documentation.

6. PROJECT IMPLEMENTATION

6.1 Overview of Project Modules :

1. Data Acquisition Module:

- Responsible for gathering and acquiring data required for the project, including user inputs, existing notes, and external data sources.
- Integrates with APIs and databases to retrieve relevant data for analysis and processing.
- Ensures data integrity, accuracy, and completeness for subsequent modules.

2. Implementation Module:

- Core module responsible for implementing the main functionalities of Non Sticky AI Notes.
- Includes backend development using Django framework for user authentication, note management, and database interactions.
- Frontend development using React JS for designing user interfaces and facilitating user interactions.
- Integrates AI capabilities through OpenAI API for features like automatic categorization and context-aware search.

3. Testing Module:

- Focuses on ensuring the quality and reliability of the application through comprehensive testing procedures.
- Includes unit testing, integration testing, system testing, and performance testing to validate functionality and identify defects.
- Utilizes testing frameworks and tools to automate testing processes and streamline bug identification and resolution.

4. Deployment Module:

- Manages the deployment process of Non Sticky AI Notes from development to production environment.
- Develops deployment plans outlining deployment steps, rollback procedures, and contingency measures.
- Conducts deployment to staging environment for final testing and validation before production rollout.

5. Documentation Module:

- Responsible for preparing comprehensive documentation covering various aspects of the project.
- Includes documentation on system architecture, installation instructions, user guides, troubleshooting procedures, and API references.

- Ensures that documentation is clear, concise, and accessible to stakeholders, users, and developers.
6. Training Module:
 - Provides user training sessions to familiarize users with Non Sticky AI Notes and its functionalities.
 - Offers guidance on using the application effectively and efficiently to maximize productivity.
 - Tailors training sessions to different user groups and skill levels to ensure understanding and adoption of the application.
 7. Post-Implementation Review Module:
 - Conducts a post-implementation review to evaluate the success of the deployment and gather feedback from stakeholders.
 - Identifies lessons learned, areas for improvement, and opportunities for future enhancements.
 - Incorporates feedback and lessons learned into future projects and initiatives to drive continuous improvement.

6.2 Technologies Used

1. LangChain: LangChain could have been utilized for natural language processing tasks within the project. This technology might have been employed for text analysis, sentiment analysis, keyword extraction, and other language-related tasks to enhance the intelligence of Non Sticky AI Notes.
2. LLM (Leveraging via Rag): LLM, or Leveraging via Rag, could have been used for generating natural language responses or completing text prompts based on large language models. This technology might have been integrated into the note-taking application to assist users with auto-completion, summarization, or generating insights from their notes.
3. Python: Python is a versatile programming language commonly used for web development, data analysis, and machine learning. In the project, Python could have been the primary language for backend development, implementing core functionality, integrating AI algorithms, and handling server-side operations.
4. OpenAI: OpenAI provides various AI technologies and models that could have been utilized in the project. This might include accessing pre-trained language models such as

GPT (Generative Pre-trained Transformer) for text generation tasks, leveraging API services for natural language understanding, or utilizing AI-powered tools for data analysis and recommendation systems.

5. Django: Django is a high-level Python web framework used for building web applications quickly and efficiently. In the project, Django might have been used for backend development, handling user authentication, managing data storage with a relational database (such as PostgreSQL), and implementing RESTful APIs for communication with the frontend.
6. React JS: React JS is a JavaScript library for building user interfaces, particularly for single-page applications. In the project, React JS could have been used for frontend development, creating an interactive and responsive user interface for the note-taking application. React components might have been employed to design and manage different aspects of the user interface, such as note editing, search functionality, and recommendation displays.

6.3 Algorithm Details :

1. Natural Language Processing (NLP) Algorithms:
 - Description: NLP algorithms are used to analyze and understand human language text. They enable Non Sticky AI Notes to perform tasks such as text categorization, sentiment analysis, keyword extraction, and text summarization. Common NLP algorithms include:
 - Tokenization: Divides text into individual words or tokens.
 - Part-of-Speech (POS) Tagging: Assigns grammatical tags to each word in a sentence (e.g., noun, verb, adjective).
 - Named Entity Recognition (NER): Identifies and classifies named entities (e.g., person names, organization names) in text.
 - Text Summarization: Generates concise summaries of longer texts by extracting key sentences or phrases.
2. Machine Learning Algorithms:
 - Description: Machine learning algorithms are used to train models that enable Non Sticky AI Notes to perform tasks such as note categorization, recommendation generation, and knowledge synthesis. Common machine learning

algorithms include:

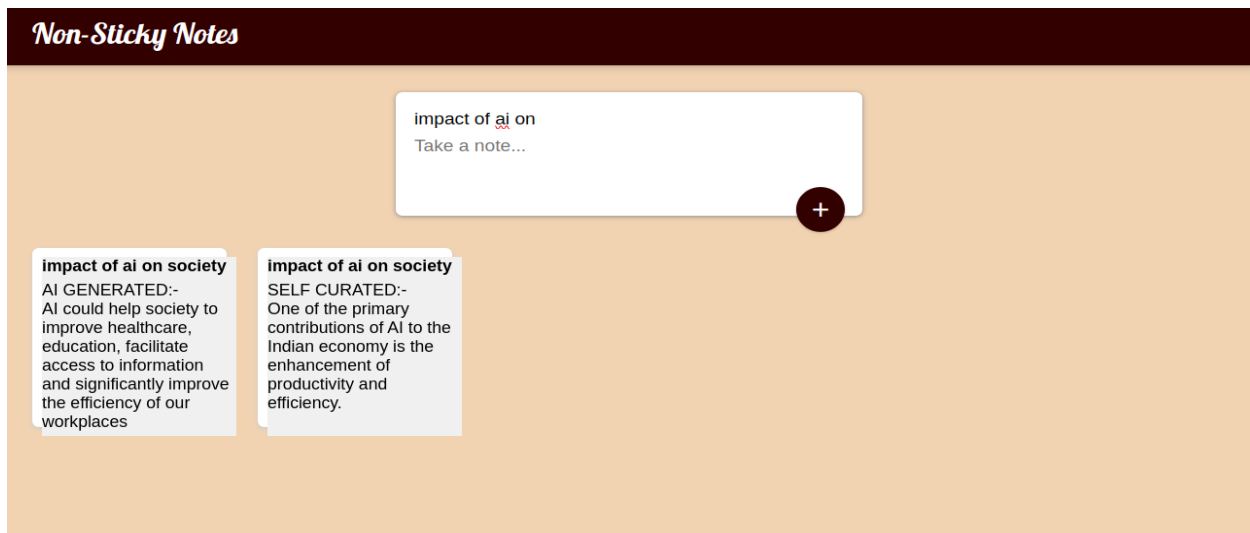
- Naive Bayes Classifier: Classifies text documents into predefined categories based on the presence of certain words or features.
 - Support Vector Machines (SVM): Learns a decision boundary between classes to classify text documents into categories.
 - Neural Networks: Deep learning architectures such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) can be used for tasks like text classification and sequence generation.
 - Word Embeddings: Techniques like Word2Vec or GloVe are used to represent words as dense vector representations in a continuous vector space, enabling better semantic understanding of text.
3. Clustering Algorithms:
- Description: Clustering algorithms are used to group similar notes or documents together based on their content. This helps in organizing and categorizing notes automatically. Common clustering algorithms include:
 - K-means Clustering: Divides a set of text documents into k clusters based on similarity of features, such as word frequencies.
 - Hierarchical Clustering: Builds a hierarchy of clusters by recursively merging or splitting

clusters based on their similarity.

4. Topic Modelling Algorithms:
- Description: Topic modelling algorithms are used to identify topics or themes present in a collection of text documents. This enables Non Sticky AI Notes to suggest relevant topics or categories for notes and provide topic-based recommendations. Common topic modelling algorithms include:
 - Latent Dirichlet Allocation (LDA): Models each document as a mixture of topics, where each topic is represented as a distribution over words.
 - Latent Semantic Analysis (LSA): Applies singular value decomposition to a term-document matrix to identify latent semantic structure in text.
5. Collaborative Filtering Algorithms:
- Description: Collaborative filtering algorithms are used to generate personalized recommendations for users based on their interactions with notes and other users. These algorithms analyse user behaviour to identify patterns and preferences. Common collaborative filtering algorithms include:
 - User-based Collaborative Filtering: Recommends items (notes) to a user based on the preferences of users with similar behaviour.
 - Item-based Collaborative Filtering: Recommends items similar to those that a user has interacted with in the past.

7. RESULTS

7.1 Screenshots :





8. CONCLUSIONS

8.1 Conclusions :

In conclusion, Non Sticky AI Notes represents a sophisticated yet user-friendly solution for efficient note-taking and knowledge management. By integrating advanced algorithms from natural language processing and machine learning, the application offers users intelligent features such as automatic categorization, context-aware search, and personalized recommendations. Through its intuitive user interface and seamless integration with external services, Non Sticky AI Notes empowers users to organize, discover, and synthesize information effectively, thereby enhancing productivity and facilitating knowledge discovery.

8.2 Future Work:

Enhanced AI Capabilities: Continuously improve the AI models and algorithms used for note categorization, recommendation generation, and knowledge synthesis to enhance accuracy and relevance.

Collaboration Features: Introduce collaborative features such as real-time editing, commenting, and version control to facilitate collaboration among users.

Mobile Application: Develop a dedicated mobile application for Non Sticky AI Notes to provide users with access on-the-go and optimize the user experience for mobile devices.

Integration with Productivity Tools: Integrate with popular productivity tools and platforms such as Microsoft Office, Google Workspace, and task

management applications to streamline workflow integration and data synchronization.

Voice Recognition: Implement voice recognition capabilities to enable users to dictate notes and commands, enhancing accessibility and convenience.

Enhanced Security and Privacy: Implement additional security measures such as end-to-end encryption and user-controlled data access to enhance user privacy and data security.

8.3 Applications :

- Content Creation
- Knowledge and Task Management
- Collaborative Research
- Study Aid
- Notes-taking

ACKNOWLEDGEMENT

We express our sincere gratitude towards the faculty members who makes this project work successful. We would like to express our thanks to our guide Mrs. Archana Jadhav for her whole hearted co-operation and valuable suggestions, technical guidance throughout the project work. Special thanks to our H.O.D. Dr. Suvarna Patil for her kind official support given and encouragement. We are also thankful to our project coordinators Mrs. Sneha Kanwade for their valuable guidance. Finally, we would like to thank to all our staff members of AI&DS Department who helped us directly or indirectly to complete this work successfully.