SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMAKURU-572103

(An Autonomous Institute under Visvesvaraya Technological University, Belagavi)

Project Report on

"Kuma: AI-Powered Personal Assistant"

submitted in partial fulfillment of the requirement for the completion of V semester of

BACHELOR OF ENGINEERING

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Submitted by

Suraj Kumar	(1SI23AD057)
Himanshu Rai	(1SI23AD016)
Aditya Raj	(1SI23CS008)

under the guidance of

Dr Sheela S

Assistant Professor

Department of Computer Science and Engineering

SIT, Tumakuru-03

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2025-26

SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMAKURU-572103

(An Autonomous Institute under Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the mini project work entitled "KUMA: AI-POWERED PERSONAL ASSISTANT" is a bonafide work carried out by Suraj Kumar (1SI23AD057), Himanshu Rai (1SI23AD016) and Aditya Raj (1SI23CS008) in partial fulfillment for the completion of V Semester of Bachelor of Engineering in Artificial Intelligence and Data Science from Siddaganga Institute of Technology, an autonomous institute under Visvesvaraya Technological University, Belagavi during the academic year 2025-26. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The Mini project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering degree.

Dr Sheela S
Assistant Professor
Dept. of CSE
SIT,Tumakuru-03

Head of the Department
Dept. of CSE
SIT,Tumakuru-03

External viva:

Names of the Examiners **Signature with date**

1.

2.

ACKNOWLEDGEMENT

We offer our humble pranams at the lotus feet of **His** Holiness, **Dr. Sree Sree Sivakumara Swamigalu**, Founder President and **His** Holiness, **Sree Sree Siddalinga Swamigalu**, President, Sree Siddaganga Education Society, Sree Siddaganga Math for bestowing upon their blessings.

We deem it as a privilege to thank **Dr. Shivakumaraiah**, CEO, SIT, Tumakuru, and **Dr. S V Dinesh**, Principal, SIT, Tumakuru for fostering an excellent academic environment in this institution, which made this endeavor fruitful.

We would like to express our sincere gratitude to **Dr Sunitha. N R**, Professor and Head, Department of Computer Science and Engineering, SIT, Tumakuru for her encouragement and valuable suggestions.

We thank our guide **Dr Sheela S**, Assistant Professor, Department of Computer Science and Engineering, SIT, Tumakuru for the valuable guidance, advice and encouragement.

Suraj Kumar	(1SI23AD057)
Himanshu Rai	(1SI23AD016)
Aditya Raj	(1SI23CS008)

Course Outcomes

CO1: To identify a problem through literature survey and knowledge of contemporary engineering technology.

CO2: To consolidate the literature search to identify issues/gaps and formulate the engineering problem

CO3: To prepare project schedule for the identified design methodology and engage in budget analysis, and share responsibility for every member in the team

CO4: To provide sustainable engineering solution considering health, safety, legal, cultural issues and also demonstrate concern for environment

CO5: To identify and apply the mathematical concepts, science concepts, engineering and management concepts necessary to implement the identified engineering problem

CO6: To select the engineering tools/components required to implement the proposed solution for the identified engineering problem

CO7: To analyze, design, and implement optimal design solution, interpret results of experiments and draw valid conclusion

CO8: To demonstrate effective written communication through the project report, the one-page poster presentation, and preparation of the video about the project and the four page IEEE/Springer/ paper format of the work

CO9: To engage in effective oral communication through power point presentation and demonstration of the project work

CO10:To demonstrate compliance to the prescribed standards/ safety norms and abide by the norms of professional ethics

CO11: To perform in the team, contribute to the team and mentor/lead the team

Attainment level: - 1: Slight (low) 2: Moderate (medium) 3: Substantial (high)

POs: PO1: Engineering Knowledge, PO2: Problem analysis, PO3: Design/Development of solutions, PO4: Conduct investigations of complex problems, PO5: Engineering tool usage, PO6: Engineer and the world, PO7: Ethics, PO8: Individual and collaborative team work, PO9: Communication, PO10: Project management and finance, PO11: Lifelong learning

PSO1:Computer based systems development,PSO2:Software development,PSO3:Computer Communications and Internet applications

CO-PO Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSo3
CO-1											3		3	
CO-2		3		3									3	
CO-3										3	3		3	
CO-4						3	3						3	
CO-5	3	3											3	
CO-6					3								3	
CO-7		3	3	3									3	
CO-8									3				3	
CO-9									3				3	
CO-10							3						3	
CO-11								3					3	

Abstract

[Add abstract here: A concise summary of the Kuma AI-powered personal assistant project. The abstract should answer:

- Why this project? (Motivation - first paragraph)
- What is the main objective? (second paragraph)
- How is it implemented? (Implementation details, tools/software used - third paragraph)

Include brief mentions of technologies used: TypeScript, Bun, React, Prisma, LangChain, Google Gemini AI, etc.]

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

[Discuss the need for AI-powered personal assistants in the modern digital age. Talk about the challenges of managing multiple tasks, handling information overload, and the need for intelligent automation. Explain how AI technologies like LangChain and Google Gemini can revolutionize personal productivity.]

1.2 Objective of the project

[State the primary objectives of the Kuma project:

- Development of an intelligent personal assistant using AI technologies
- Integration of multiple services (Gmail, Calendar, Docs, Drive)
- Implementation of specialized agents for different tasks
- Creating a user-friendly interface for seamless interaction
- Providing secure authentication and data management

]

1.3 Organisation of the report

[Describe the structure of the report:

- Chapter 1: Introduction to the project and its objectives
- Chapter 2: Literature survey on AI assistants, LangChain, and related technologies
- Chapter 3: System design, architecture, and methodology
- Chapter 4: Implementation details of backend and frontend
- Chapter 5: Results, screenshots, and performance analysis

- Chapter 6: Conclusion and future enhancements
- Bibliography and Appendices

]

Chapter 2

Literature Survey

[This chapter should include a comprehensive review of:

- Existing AI-powered personal assistants (Google Assistant, Siri, Alexa)
- LangChain framework and its applications
- Google Gemini AI capabilities
- Agent-based systems and multi-agent architectures
- Web technologies: React, TypeScript, Vite
- Backend technologies: Bun runtime, Express, Prisma ORM
- OAuth authentication and security best practices
- Database design for chat applications

Include citations from IEEE, Springer, ACM, and other peer-reviewed journals and conferences. Use proper citation format as shown: [?], [?].]

Chapter 3

System Design & Methodology

3.1 Functional & Non-Functional Requirements

3.1.1 Functional Requirements

[List functional requirements:

1. User authentication and authorization
2. Chat interface for user interactions
3. AI agent processing and response generation
4. Integration with Google services (Gmail, Calendar, Docs, Drive)
5. File upload and document management
6. Memory management for conversation context
7. Web search capabilities
8. Stock market information retrieval
9. Vision capabilities for image processing

]

3.1.2 Non-Functional Requirements

[List non-functional requirements:

1. Performance: Response time ¡ 2 seconds
2. Security: Encrypted data storage and secure authentication
3. Scalability: Support for multiple concurrent users
4. Reliability: 99% uptime

5. Usability: Intuitive user interface
6. Maintainability: Modular code architecture

]

3.2 List of Hardware & Software Requirements

3.2.1 Hardware Requirements

[Specify hardware requirements:

- Processor: Intel Core i5 or equivalent (minimum)
- RAM: 8 GB (minimum), 16 GB (recommended)
- Storage: 20 GB free space
- Network: Stable internet connection

]

3.2.2 Software Requirements

[Specify software requirements:

- Operating System: Windows 10/11, macOS, or Linux
- Runtime: Bun ¿= 1.0.0
- Database: PostgreSQL
- Node.js: Version 18+ (optional)
- Browser: Chrome, Firefox, Safari (latest versions)
- Development Tools: VS Code or similar IDE
- Version Control: Git

Key Technologies:

- Backend: TypeScript, Bun, Express, Prisma ORM
- Frontend: React, Vite, TypeScript, Zustand, shadcn/ui

- AI/ML: LangChain, Google Gemini API
- APIs: Google OAuth, Gmail API, Calendar API, Drive API

]

3.3 System Architecture

[Describe the overall system architecture. Include:

- Three-tier architecture diagram
- Client-Server communication flow
- Database schema overview
- AI agent architecture
- Integration layer for external services

Refer to Figure 3.1 for the system architecture diagram.]

Figure 3.1: System Architecture of Kuma AI Assistant

3.4 Data Flow Diagrams

3.4.1 Level 0 DFD - Context Diagram

[Describe the context-level data flow showing the system as a single process with external entities (User, Google Services, AI APIs)]

Figure 3.2: Context Diagram (Level 0 DFD)

3.4.2 Level 1 DFD

[Describe the Level 1 DFD showing major processes: Authentication, Chat Management, Agent Processing, File Management, Integration Services]

Figure 3.3: Level 1 Data Flow Diagram

3.4.3 Level 2 DFD - Agent Processing

[Describe detailed data flow within the agent processing module]

Figure 3.4: Level 2 DFD - Agent Processing Module

3.5 Algorithms

3.5.1 Chat Processing Algorithm

[Describe the algorithm for processing user queries:

1. Receive user input
2. Authenticate user session
3. Retrieve conversation context from memory
4. Select appropriate agent based on query type
5. Process query using LangChain agent
6. Generate response using Gemini AI
7. Update conversation memory
8. Return response to user

]

3.5.2 Agent Selection Algorithm

[Describe how the system selects the appropriate agent for a given task]

3.5.3 Memory Management Algorithm

[Describe the algorithm for managing conversation context and history]

3.5.4 OAuth Authentication Flow

[Describe the OAuth 2.0 authentication flow for Google services integration]

Chapter 4

Implementation Details

4.1 Backend Implementation

4.1.1 Project Setup

[Describe the backend setup using Bun, TypeScript, and Express. Include package installation, configuration, and project structure.]

4.1.2 Database Design

[Describe the Prisma schema, database tables, relationships, and migrations. Include the schema for:

- User table
- Chat table
- Message table
- Agent table
- Document table
- Memory table

]

4.1.3 API Endpoints

[Document all API endpoints:

- Authentication routes (/auth)
- Chat routes (/chat)
- Agent routes (/agents)
- Document routes (/documents)

- Upload routes (/upload)
- App integration routes (/apps)

]

4.1.4 AI Integration

[Describe the implementation of:

- LangChain agent configuration
- Google Gemini API integration
- Custom tool development
- Prompt engineering
- Memory management using LangChain

]

4.1.5 Google Services Integration

[Describe the implementation of:

- Gmail API integration
- Google Calendar integration
- Google Docs integration
- Google Drive integration
- OAuth 2.0 authentication flow

]

4.2 Frontend Implementation

4.2.1 Project Setup

[Describe the frontend setup using React, Vite, and TypeScript. Include component structure and routing.]

4.2.2 State Management

[Describe Zustand store implementation for managing application state]

4.2.3 UI Components

[Describe the implementation of key UI components:

- Chat interface
- Message bubbles
- File upload component
- Authentication forms
- Navigation menu
- Settings panel

]

4.2.4 API Integration

[Describe how the frontend communicates with the backend API, including error handling and loading states]

4.3 Security Implementation

[Describe security measures:

- JWT-based authentication
- Data encryption
- CORS configuration
- Input validation and sanitization
- Secure API key management

]

4.4 Code Snippets

[Include relevant code snippets for key implementations]

Chapter 5

Results

5.1 Screenshots

[Include screenshots of:

- Login/Registration page
- Main chat interface
- AI agent responses
- File upload functionality
- Google service integrations
- Settings page
- Mobile responsive views

Example reference:]

Figure 5.1: Main Chat Interface

Figure 5.2: Gmail Integration

5.2 Analysis

5.2.1 Performance Metrics

[Present performance analysis with tables and graphs:

- Response time measurements
- API latency

- Database query performance
- Memory usage
- Concurrent user handling

]

Table 5.1: Performance Metrics

Metric	**Minimum**	**Average**	**Maximum**
Response Time (ms)			
API Latency (ms)			
Memory Usage (MB)			
Concurrent Users			

5.2.2 Comparison with Existing Systems

[Compare Kuma with existing AI assistants]

Table 5.2: Comparison with Existing AI Assistants

Feature	**Kuma**	**Google Assistant**	**Siri**	**Alexa**
Custom Agents	Yes	No	No	Limited
Gmail Integration	Yes	Yes	No	No
Calendar Integration	Yes	Yes	Yes	Yes
Document Management	Yes	Limited	Limited	No
Open Source	Yes	No	No	No
Customizable	Yes	No	No	Limited

5.2.3 Testing Results

[Present results from:

- Unit testing
- Integration testing
- User acceptance testing

- Performance testing

]

5.2.4 User Feedback

[If applicable, include user feedback and satisfaction metrics]

Chapter 6

Conclusion & Future Enhancement

6.1 Conclusion

[Summarize the project achievements:

- Successfully developed an AI-powered personal assistant
- Implemented multi-agent architecture using LangChain
- Integrated multiple Google services seamlessly
- Created a responsive and user-friendly interface
- Achieved performance goals and security requirements

Highlight the contributions and learning outcomes from the project.]

6.2 Future Enhancement

[Discuss potential future improvements:

- Support for additional service integrations (Slack, Microsoft Office)
- Voice input and output capabilities
- Mobile application development
- Advanced natural language processing
- Personalized agent training based on user behavior
- Multi-language support
- Collaborative features for team usage
- Enhanced security with biometric authentication
- Integration with IoT devices

- Offline mode capabilities

]

Bibliography

[1] LangChain Documentation, *"LangChain: Building applications with LLMs through composability"*, `https://langchain.com/docs`, 2024.

[2] Google DeepMind, *"Gemini: A Family of Highly Capable Multimodal Models"*, arXiv preprint, 2023.

[3] Meta Open Source, *"React: A JavaScript library for building user interfaces"*, `https://react.dev`, 2024.

[4] Microsoft Corporation, *"TypeScript: JavaScript with syntax for types"*, `https://www.typescriptlang.org`, 2024.

[5] Prisma Data, Inc., *"Prisma: Next-generation Node.js and TypeScript ORM"*, `https://www.prisma.io`, 2024.

[6] IETF OAuth Working Group, *"The OAuth 2.0 Authorization Framework"*, RFC 6749, October 2012.

Appendices

Appendix A

Sustainable Development Goals addressed

#	**SDG**	**Level**
1	No Poverty	
2	Zero Hunger	
3	Good Health and Well-being	
4	Quality education	
5	Gender Quality	
6	Clean water and Sanitation	
7	Affordable and Clean Energy	
8	Decent work and Economic Growth	
9	Industry, Innovation and Infrastructure	
10	Reduced Inequalities	
11	Sustainable cities and Communities	
12	Responsible Consumption and production	
13	Climate action	
14	Life below water	
15	Life on Land	
16	Peace, Justice and Strong Institutions	
17	Partnership's for the Goals	

Levels: Poor:1, Good :2, Excellent:3

Appendix B

Self-Assesment of the Project

#	PO and PSO	Contribution from the Project	Level
1	Engineering Knowledge:		
2	Problem Analysis:		
3	Design/development of solutions		
4	Conduct investigations of complex problems:		
5	Modern tool usage:		
6	The Engineer and the world:		
7	Ethics:		
8	Individual and Team Work:		
9	Communication:		
10	Project Management and Finance:		
11	Life-long Learning:		
1	PSO1		
2	PSO2		
3	PSO3		

PSO1: Computer based systems development: Ability to apply the basic knowledge of database systems, computing, operating system, digital circuits, microcontroller, computer organization and architecture in the design of computer based systems.

PSO2: Software development: Ability to specify, design and develop projects, application softwares and system softwares by using the knowledge of data structures, analysis and design of algorithm, programming languages, software engineering practices and open source tools.

PSO3: Computer communications
and Internet applications: Ability to design and develop network protocols and internet applications by incorporating the knowledge of computer networks, communication protocol engineering, cryptography and network security, distributed and cloud comput-

ing, data mining, big data analytics, ad hoc networks, storage area networks and wireless sensor networks.

Levels: Poor:1, Good :2, Excellent:3

Appendix C

Data Sheet of component 1

Note: Only include relevant details of the components that are referred w.r.t. project.

Appendix D

Data Sheet of component 2