A Multi-Agent LLM Orchestration for Intelligent Fitness and Diet Personalization

DISSERTATION

Submitted in partial fulfillment of the requirements of the

Degree: MTech in Artificial Intelligence & Machine Learning

By

Rohit
2023AA05918

Under the supervision of

Ketan Kishore
Senior Consultant, T-Systems.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
Pilani (Rajasthan) INDIA

May, 2025

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
SECOND SEMESTER 2024-25

DSECLZG628T / **AIMLCZG628T DISSERTATION**

Dissertation Title: A Multi-Agent LLM Orchestration for Intelligent Fitness and Diet Personalization

Name of Supervisor**:** Ketan Kishore

Name of Student : Rohit

ID No. of Student : 2023AA05918

Courses Relevant for the Project & Corresponding Semester:
1. NLP Applications
2. Deep Neural Network
3. Conversational AI
4. MLOps

Abstract
(in about 500 words)

In the era of artificial intelligence, personalized health and fitness solutions are an imminent reality, supplanting one-size-fits-all approaches. However, most available resources for diet and exercise remain generic or fragmented, leaving a gap for automated systems that integrate diverse information and individual context. This dissertation addresses this need by developing an AI-driven fitness and nutrition assistant that synthesizes user-specific data, domain knowledge, and adaptive reasoning into coherent recommendations. The core problem is to generate customized dietary and exercise plans from a user's profile and goals, rather than relying on static guidelines. To this end, we propose a modular, multi-agent LLM-based architecture: the system decomposes the overall task into specialized subtasks (e.g. user profiling, meal planning, workout design, query handling) and assigns them to dedicated agents. This design follows recent multi-agent AI frameworks, which enable dynamic task decomposition and specialized expertise.

The system is implemented with a Python backend and a Streamlit frontend for the user interface. LangFlow – a Python-based low-code visual builder for AI workflows – orchestrates the system's logic and agent interactions. We instantiate several LLM-driven agents, including a routing agent and domain specialists. The routing agent, powered by a large language model, classifies each user query or task and directs it to the appropriate specialist agent. For example, one agent validates and enriches the user's profile data, another constructs personalized nutrition plans, a third designs workout routines, and a Query Agent formulates answers based on retrieved knowledge. DataStax Astra DB serves as the system's vector database: it stores embeddings of nutrition and fitness knowledge so that relevant information can be retrieved via Retrieval-Augmented Generation (RAG). In operation, user queries trigger a RAG pipeline that semantically searches the vector store for contextually relevant documents. The retrieved content, together with the user's input data, is then passed to the selected agent to produce a grounded, evidence-based response.

User interaction begins with a Nutrition Form that collects demographic and fitness details (age, gender, weight, height, activity level) and the user's fitness goal (fat loss or muscle gain). Based on this input, the

system offers personalized nutrition targets (daily calories and macronutrient distribution). Users may accept these targets or adjust them manually. A free-text “Your Notes” section allows users to record additional context (e.g. medical conditions, preferred supplements, equipment constraints), which the system incorporates into its reasoning. The “AskAI” conversational module lets users ask natural-language questions about diet, exercise, or their own data; these queries are handled by the multi-agent system. The LLM routing agent first interprets the question and gathers context: it uses the user’s profile and notes, and performs a RAG lookup on Astra DB to retrieve pertinent facts. The question and context are then delivered to the appropriate specialist agent or tool. This approach is similar to recent AI nutritionist prototypes that combine language models, function-calling, and RAG grounding to provide personalized advice.

Because language models often struggle with precise arithmetic or multi-step calculations, the system incorporates a dedicated calculator tool. When an agent must perform numeric reasoning (e.g. computing basal metabolic rate, body mass index, or macronutrient ratios), it calls the calculator to ensure accuracy. The overall methodology involves designing LangFlow graphs that represent the workflow: each node is an agent or function, with edges encoding the decision logic. During execution, the orchestration layer manages these interactions: the router agent triggers RAG-based retrieval from Astra DB, specialized agents generate responses (possibly invoking tools), and outputs are formatted for the user. This hybrid approach combines symbolic computation and LLM-generated language to deliver reliable, context-aware guidance.

The expected outcomes of this research include a functional prototype that generates user-specific diet and exercise recommendations. By structuring nutritional knowledge in a vector database and leveraging multi-agent LLMs, the system aims to produce contextually grounded and accurate guidance tailored to each user profile. This dissertation demonstrates the feasibility of using low-code workflow tools like LangFlow and multi-agent LLM frameworks in digital health applications. In future work, the architecture could be extended with real-time web search (e.g. via the Serper API) to incorporate up-to-date information, and with automated notifications for proactive engagement. Overall, the proposed AI assistant exemplifies how a carefully designed LLM-based multi-agent system can effectively personalize fitness and nutrition coaching for diverse individuals.

Keywords: Personalized Nutrition, Artificial Intelligence, Machine Learning, Fitness Assistant, Large Language Models, AI Agents, Multi-Agent LLM, Retrieval-Augmented Generation (RAG), LangFlow, Astra DB, AI Health Applications, Natural Language Processing, Vector database, Streamlit.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
II SEMESTER 24-25
DSECLZG628T / **AIMLCZG628T DISSERTATION**
Dissertation Outline (Abstract)

BITS ID No.: 2023AA05918 Name of Student: Rohit

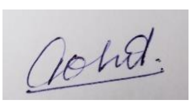
Name of Supervisor: Ketan Kishore

Designation of Supervisor: Senior Consultant, T-Systems.

Qualification and Experience: B.Tech + M.Tech, Computer Science Engineering, BITS Pilani

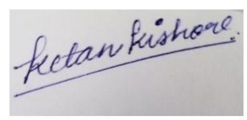
Official E-mail ID of Supervisor: ketan.kishore@t-systems.com

Topic of Dissertation: A Multi-Agent LLM Orchestration for Intelligent Fitness and Diet Personalization



(Signature of Student)

Date: 24.05.2025



(Signature of Supervisor)

Date: 24.05. 2025

Project Work Title

A Multi-Agent LLM Orchestration for Intelligent Fitness and Diet Personalization

Discussion on the chosen topic

• The **purpose of the work and expected outcome of the work:**

- **Purpose:** The purpose of this project is to design and develop an AI-powered system that delivers personalized fitness and nutrition recommendations based on individual user profiles, preferences, and goals. By leveraging large language models (LLMs), multi-agent architecture, and retrieval-augmented generation (RAG), the system aims to:
    - Automate the creation of diet and workout plans tailored to each user's demographic and lifestyle data.
    - Enable users to interact naturally with an AI assistant for fitness-related queries.
    - Incorporate contextual understanding through user notes and vector-based knowledge retrieval.
    - Provide an accessible and intelligent digital wellness assistant that adapts to user needs without requiring technical expertise.
- **Expected Outcome:** A proof-of-concept AI system capable of:
    - A working prototype of the AI-driven fitness and nutrition assistant with a web-based UI.
    - Successful multi-agent orchestration: the routing agent should correctly delegate tasks and the sub-agents should produce coherent plans or answers.
    - Accurate calculation of nutrition targets (calories, protein, fats, carbs) validated against standard formulas.
    - A functioning RAG module: user queries about nutrition or fitness should return context-relevant answers grounded in the knowledge base (e.g. dietary guidelines).
    - A demonstration of the "Your Notes" integration: personal constraints or conditions are incorporated into the advice.
    - These outcomes will show that the combination of LLM agents, RAG, and user data can yield personalized and reliable nutrition advice.

• Literature **review done in connection with the work, if applicable:** A comprehensive literature review will be conducted, focusing on:

AI in Personalized Nutrition and Fitness

- Traditional systems use fixed rules or algorithms for nutrition plans.
- Modern research explores LLM-driven systems that offer dynamic, personalized meal plans.
- ChatDiet: Uses LLMs to generate meal plans based on user profiles.
- Studies by Szymanski et al. show that prompt design critically affects the quality of LLM-generated nutrition advice.
- Techniques like chain-of-thought prompting and retrieval augmentation improve response accuracy.
- LLMs show strong potential for health applications but must be carefully designed for safety and reliability.

Multi-Agent AI Architectures

- Multi-agent systems split tasks across specialized LLM agents coordinated by a routing or "meta" agent.
- Each agent focuses on a narrow domain (e.g. nutrition logic, user queries) for improved performance.
- LangFlow and LangGraph are frameworks that support agent workflows with custom prompts and tool integrations.
- LangFlow examples show agents calling specific tools (e.g., calculators or web search) based on task type.
- This architecture matches the proposed system design (e.g., distinct agents for nutrition planning and Q&A).

Retrieval-Augmented Generation (RAG)

- RAG improves factual accuracy by grounding LLM outputs in external knowledge.
- A vector database (like AstraDB) stores documents in embedding form for semantic retrieval.
- When a query is received, relevant text chunks are retrieved and passed to the LLM as context.

Tool and Platform Choices

- Python is the standard for backend AI development.
- Streamlit enables quick development of interactive web UIs for health apps.
- LangFlow is a low-code framework ideal for visually building LLM workflows.
- AstraDB offers scalable, serverless vector storage suitable for storing health guidelines and user context.
- Past projects successfully used LangChain, Streamlit, and vector DBs for fitness and health chatbots.

Conclusion from Literature

- Combining LLMs, RAG, and modular agent design creates effective and scalable personalized health assistants.
- Success depends on careful prompt design, validation of outputs, and responsible use of domain knowledge.

Challenges and limitations of existing methodologies.

- Applications of Artificial Intelligence (AI) and Machine Learning (ML) in project management, including predictive analytics, simulation, and optimization.
- Advancements in Large Language Models (LLMs), AI agents, and Retrieval Augmented Generation (RAG) for complex reasoning, knowledge integration, and task automation.
- Knowledge representation techniques, including knowledge graphs, for industrial applications.
- Specific AI techniques relevant to scheduling: NLP for scope understanding (leveraging LLMs), predictive modeling for durations and risks, simulation for scenario analysis, optimization algorithms for schedule generation and resource leveling.
- Data requirements and strategies for grounding LLMs, RAG systems, and analytical models in industrial contexts.
- Case studies of AI, particularly LLM and agent-based systems, implementation in similar complex project environments.
- Metrics for evaluating the performance of AI-driven scheduling systems.

- Brief **discussion on the existing process and its limitations:** Existing processes for industrial shutdown planning are often heavily reliant on manual effort, spreadsheets, and the experience of a few key personnel.

- This can lead to: Existing fitness and nutrition planning systems are often rule-based or static, lacking personalization.
    - Many apps offer generic plans that do not consider individual health data, preferences, or constraints.
    - Traditional models cannot handle open-ended user queries or adjust recommendations dynamically.
    - These systems are limited in reasoning, cannot explain decisions, and struggle with integrating new knowledge.

- Justification **for selecting a particular methodology for completing the tasks:** An AI-driven methodology centered on LLMs, AI agents, and RAG, augmented by predictive analytics, simulation, and optimization techniques, is selected due to its potential to overcome the limitations of traditional approaches. Specifically:

    - **Advanced Reasoning and NLP:** LLMs provide sophisticated natural language understanding for interpreting scope documents and can perform complex reasoning tasks essential for planning.
    - **Multi-agent architecture** allows modular task delegation, improving reliability and scalability.
    - **RAG (Retrieval-Augmented Generation)** ensures factual grounding, enhancing the trustworthiness of responses.
    - **LangFlow** simplifies development by enabling visual orchestration of agents and tools.
    - The selected tools and design patterns align with the latest research in intelligent health assistants.
- Brief **discussion on the Project Work methodology:** The development of the AI-powered personalized fitness and nutrition assistant system is organized into the following phases, ensuring a structured, iterative, and outcome-driven workflow:

Phase 1: Requirement Analysis and System Design:

- Identify core functional and non-functional requirements of the system.
- Define user interactions, data flow, and system components (LLM agents, RAG, frontend).
- Finalize the architectural blueprint, detailing the use of LangFlow for agent orchestration, AstraDB for vector storage, and Streamlit for the frontend interface.
- Design user input forms and determine the required logic for calculating nutrition targets (calories, macronutrients).

Phase 2: Agent Development and Workflow Configuration

- Implement a multi-agent architecture using LangFlow, comprising:
- A routing agent to delegate tasks.
- A Nutrition Planning agent to compute and suggest personalized diet targets.
- A Query Handling agent to answer health-related user queries.
- Define prompt templates, tools (e.g., calculator), and agent responsibilities within LangFlow’s visual framework.
- Configure agent communication and control flow for seamless task routing and output generation.

Phase 3: Vector Database and RAG Integration

- Set up AstraDB as a scalable, cloud-based vector database.
- Curate and ingest domain-specific knowledge (dietary guidelines, workout references, health constraints) into the database using embedding models.
- Build the RAG pipeline to enable semantic retrieval based on user queries and profile data.
- Test the retrieval accuracy and optimize chunking strategies for high context relevance.

Phase 4: Frontend Interface Development

- Develop a web-based frontend using Streamlit to enable:
- User data entry via a structured Nutrition Form.
- Free-text input for personalized notes (e.g., health conditions, preferences).
- Real-time interaction with the AskAI chat module.
- Ensure smooth integration with backend agents and display of personalized diet and fitness outputs.

Phase 5: Integration and System Testing

- Integrate all components: frontend, agent workflows, vector store, and computation tools.
- Perform end-to-end functional testing across various user scenarios.
- Validate output accuracy (e.g., nutrition calculations) and response quality from the AskAI module.
- Address edge cases such as incomplete data, conflicting inputs, or ambiguous queries.

Phase 6: Evaluation, Documentation, and Refinement

- Conduct formal evaluation based on predefined metrics (accuracy, usability, contextual relevance).
- Collect qualitative user feedback to assess satisfaction and usability.
- Document the complete system architecture, methodology, challenges encountered, and results.
- Finalize the dissertation report and prepare supporting materials (demo video, slides, code repository).

- Benefits **derivable from the work:**
  - **Provides personalized** and context-aware fitness/nutrition recommendations.
  - Reduces dependency on manual research or one-size-fits-all apps.
  - Improves user engagement through natural conversation and adaptive feedback.
  - Demonstrates real-world utility of multi-agent LLM systems in digital health.

- Any **other details in support of the work:** The increasing complexity of core industries and the critical nature of their shutdown operations necessitate innovative solutions. The advent of powerful LLMs, AI agents, RAG techniques, and advanced analytics, coupled with the availability of project data, provides a unique opportunity to transform shutdown planning. This research aligns with industry trends towards digitalization and advanced AI-driven decision-making.

1. Broad Area of Work

This project lies at the intersection of **Artificial Intelligence, Health Informatics, and Personalized Wellness Technologies**. It focuses on applying **multi-agent systems, Large Language Models (LLMs), and Retrieval-Augmented Generation (RAG)** to build intelligent, adaptive solutions in the **fitness and nutrition domain**.

Key areas involved include:

- **Natural Language Processing (NLP):** For understanding user input and generating human-like responses.
- **AI Agent Architectures:** Modular task delegation using specialized LLM agents under a routing controller.
- **Retrieval-Augmented Generation:** Enhancing response reliability by grounding LLM outputs in stored domain knowledge via vector search.
- **Human-AI Interaction:** Delivering real-time, personalized coaching via a web-based interface using Streamlit.
- **Low-code Orchestration:** Using tools like LangFlow to visually configure agent flows and tool integrations.
- **Health-focused Applications:** Providing actionable, personalized fitness and dietary recommendations using AI.

This work contributes to the growing field of **AI applications in digital health**, particularly for **scalable, context-aware, and user-centric systems** that assist individuals in achieving fitness and nutrition goals.

2. Objectives

The objectives of my project are as follows:

- System Design**:** Design a multi-agent architecture in Python that employs LangFlow to orchestrate several LLM-based agents. Specify roles such as a routing agent (for understanding user queries and context) and specialized agents (e.g. Nutrition Advisor, Exercise Coach). Define data schemas and flows, including how user profiles and notes feed into the agents.
- Implement a multi-agent architecture using LangFlow, where specialized LLM agents handle sub-tasks (e.g. nutrition calculation, query handling) under the coordination of a routing agent
- Integrate a vector database (AstraDB) to store and retrieve domain knowledge for Retrieval-Augmented Generation, enhancing answer accuracy and contextual relevance.
- Create a user-friendly Streamlit frontend for users to enter personal data via a Nutrition Form and to interact with the system (including the AskAI chat interface).
- Enable context-aware question answering (AskAI) by embedding user queries and using vector search to provide evidence-based responses

- Ensure accurate nutritional computations by integrating a calculator tool (for BMI, BMR, macro calculations) callable by LLM agents when needed.
- Validate and evaluate the system through testing to verify the correctness of recommendations and user satisfaction.

3. Scope of Work

1. Domain Focus

- Target industries include **individual fitness and nutrition planning** for general users.
- The system will target common use cases such as: Weight management, Muscle building, healthy lifestyle support.
- Provides customized guidance based on user-specific data (e.g., demographics, goals, constraints).
- **Designed to support:**
  - Initial nutrition and fitness plan generation.
  - Follow-up adjustments via user queries.
- **Excludes**:
  - Enterprise-level analytics or large-scale health data systems.
  - Real-time tracking or continuous monitoring (e.g. wearable integration) – reserved for future work.
  - Clinical or medical nutrition (e.g., disease-specific diets).

2. System Inputs

- **User Profile:** Demographic and anthropometric data (age, gender, weight, height) and self-reported activity level (sedentary to very active).
- **Fitness Goals:** User's goals (e.g. "lose weight", "gain muscle") and timeline.
- **Nutrition Goals:** Target values for daily calories and macronutrients. These can be manually entered or computed by the system (e.g. using standard formulas to suggest calorie targets based on goals).
- **Personal Constraints:** Information on dietary restrictions or preferences (e.g. vegetarian, allergies), medical conditions (diabetes, hypertension), available exercise equipment, and mobility limitations. These are entered in the "Your Notes" section and will influence all recommendations.
- **Questions/Queries:** Natural language questions typed by the user in the "AskAI" chat (e.g. "How can I increase my protein intake?" or "Suggest a low-carb meal plan based on my profile.").

3. Core AI/ML Components

- **Streamlit Frontend:** Implements the web interface for data entry and display. Provides forms for profile/goal input, and an interactive chat box for the AskAI feature. Visualizes plans (e.g. a simple table of daily meals and exercises) and a Gantt-like view of weekly schedules.
- **LangFlow Multi-Agent Workflow:** Defines multiple agent nodes and connectors. The key components include a Router Agent (LLM) and specialized agents. Agents use LLMs (such as GPT-4 or open-source equivalents) for natural language reasoning.
- **RAG with AstraDB:** The vector database stores embeddings of (a) user-specific information and (b) domain knowledge (e.g. nutritional guidelines, recipes, exercise tips). A retrieval module feeds relevant text snippets into the LLM prompts. This grounds the agents' outputs in

factual data.

- **Calculator Tool:** A math tool integrated with LangFlow (e.g. Python's math or NumPy) to handle quantitative questions (BMI calculation, unit conversions, macronutrient math).
- **Agent Prompt Engineering:** Carefully crafted prompts for each agent to ensure alignment with nutrition science and user context. Possibly leverage few-shot examples or fine-tuning on nutrition Q&A data if available.
- **Backend Infrastructure:** Python-based server managing the workflow. Uses OpenAI API (or similar) for LLM calls, connectors to AstraDB for retrieval, and LangFlow's runtime.

Data requirements are modest: mostly structured input from users and a curated set of nutrition/exercise texts (could reuse public sources like USDA nutrient databases, fitness guidelines, etc.). No large-scale health records are needed. All sensitive personal data remains local to the user's session.

4. Expected Outputs

- **Working Prototype:** A functional Streamlit application where a user can enter their profile and interact with the AI assistant. The system should generate personalized meal and exercise plans and answer user questions contextually.
- **AI-Driven Recommendations:** The assistant will produce daily nutrition plans (with appropriate calorie and macro targets) and basic workout suggestions tailored to the user's profile. These outputs should reflect current nutritional guidelines and fitness best practices as validated by the RAG knowledge base.
- **Context-Aware Q&A:** The "AskAI" feature will answer user queries by leveraging the user's stored profile and notes. For example, if the user has "lactose intolerance" in their notes, the assistant's nutrition advice will avoid dairy. The routing agent and RAG pipeline together ensure the answers use user-specific context.
- **Integration of Tools:** Numerical calculations (e.g. computing BMI, BMR, or nutrient totals) will be handled by the calculator tool, with results presented clearly. This demonstrates multi-modal capability (language + computation).
- **Performance Evaluation:** We expect empirical results showing that the AI assistant meets key criteria better than a non-AI baseline. For example, plans generated by the system should more closely match recommended dietary allowances and user constraints than generic templates. User study results or expert review may show higher satisfaction and relevance.

Overall, the anticipation is that a multi-agent, LLM-based system can effectively emulate aspects of a personalized fitness coach or nutritionist. The AI assistant should reduce the time and effort needed for users to plan their diets and workouts, and adapt immediately to their individual needs and feedback.

5. User Interface and Interaction

A user-friendly web-based interface is developed using Streamlit to ensure seamless interaction between the end-user and the underlying AI-powered system. The interface is designed to guide users through the process of submitting their personal fitness data, receiving tailored nutrition plans, and engaging in real-time, context-aware conversations with the AI assistant. The key interaction modules are outlined below:

Nutrition Form: A structured form enables users to input essential personal details such as gender, age, weight, height, activity level, and fitness goal (fat loss or muscle gain). This data is used to compute baseline nutritional targets (calories, protein, fats, carbohydrates) or can be manually adjusted by the user.

Your Notes Section: A free-text input area where users can specify additional contextual information, including medical conditions, dietary restrictions, supplement intake, equipment availability, or specific areas of focus (e.g., mobility issues, leg strengthening). These notes are stored and utilized by the system to provide more personalized and safe recommendations.

AskAI Conversational Module: A chat-style interface where users can ask open-ended questions about their fitness plan, dietary needs, or general wellness guidance. The system retrieves relevant context from the user's profile and notes using the RAG pipeline and routes the question to the appropriate LLM agent to generate a context-aware response.

Output Display: Personalized nutrition targets, recommendations, and AI-generated responses are presented in a clean, readable format. Tabular views or simple charts may be used to enhance interpretability.

Real-Time Feedback and Updates: Users can iteratively modify their inputs or notes and immediately receive updated plans or answers. This encourages active engagement and plan refinement over time.

The interface is developed to be lightweight, responsive, and intuitive, requiring no technical background. It ensures that the system remains accessible to general users while offering the sophistication of AI-powered personalization in the background.

6. Evaluation and Validation

- The prototype will be evaluated on its ability to:
  - **Generate Context-Aware Recommendations**:Test if meal and workout plans align with user inputs like age, activity level, goals, and constraints.
  - **Answer User Queries Effectively**: Validate the "AskAI" module for relevance and correctness by asking common fitness/nutrition questions and checking if it uses personal context correctly.
  - **Perform Calculations Accurately**: Check if the calculator tool gives correct outputs for common fitness metrics (e.g., BMI, BMR, macros).
  - **Ensure Functional UI Interaction**: Verify smooth data input, plan generation, and query handling via the Streamlit interface.
  - **Test Agent and RAG Integration**: Confirm proper flow between routing agent, specialized agents, and vector database retrieval (AstraDB).
  - **Compare to Baseline Plans**: Compare AI-generated plans with static templates for personalization and completeness.
  - **Collect Feedback**: Gather basic user feedback on output quality and interface usability to refine the system.

4. Detailed Plan of Work (for 16 weeks)

Serial Number of Task/ Phases	**Tasks or subtasks to be done** (be precise and specific)	**Start Date-End Date**	**Planned duration in weeks**	**Specific Deliverable in terms of the project**

1.	**Literature Review and Problem Definition** - Study of existing shutdown AI nutrition/fitness apps, LLM and agent techniques (e.g., ReAct, Toolformer) and RAG methods. - Review of research on LLMs, AI agents, RAG, workflows and fitness and nutrition algorithms. - Identification of gaps and motivation for the proposed system.	Week 1 – Week 2 (May 3 – May 16)	2	- Summary of literature review. - Finalized problem statement and objectives. - Initial project plan.
2.	**Domain Understanding and Architectural Design** - Design the system architecture: outline LangFlow workflow (router agent and specialized agents), data schemas (user profile, nutrition goals, notes). Select LLM models and tools (calculator, database). Set up development environment (install Streamlit, LangFlow, AstraDB client).	Week 3 – Week 4 (May 17 – May 30)	2	- Architecture diagrams and flowcharts; prototype LangFlow layout; tech setup completed.
3.	**RAG Pipeline and Vector Database Setup** - Build a RAG pipeline to retrieve relevant knowledge from shutdown data.	Week 5 – Week 6 (May 31 – June 13)	2	- Functional data input forms; populated vector database with seed content; initial retrieval tests.
	- Store documents and historical plans using vector embeddings. - Enable semantic search for grounding AI-generated plans.			
4.	**LLM-Based Scope Understanding and Task Extraction** - Use prompt engineering or fine-tuning to convert natural language scope into structured tasks. - Generate initial WBS from project descriptions. - Ensure alignment between LLM output and domain semantics.	Week 7 – Week 8 (June 14 – June 27)	2	- Working LangFlow agents with RAG and calculation; sample agent responses for test profiles.

5.	**Design and Implementation of AI Agents** - Define agents for scheduling, optimization, risk evaluation, and scenario simulation. - Implement task coordination logic across agents. - Integrate RAG and LLM outputs with agent workflows.	Week 9 – Week 10 (June 28 – July 11)	2	- Integrated application prototype; demo of UI with data entry and chat functionality.
6.	**Interactive UI Development** - Create interface for input submission, Gantt chart visualization, and natural language interaction. - Allow users to view AI assumptions and modify plans. - Enable scenario analysis via user prompts.	Week 11 – Week 12 (July 12 – July 25)	2	- Improved AI response quality; richer knowledge content; pilot usage scenarios and refinements.
7.	**System Integration and Testing** - Combine all modules: LLMs, RAG, agents, analytics, and UI. - Conduct functional and usability testing with real/simulated data. - Perform performance benchmarking.	Week 13 – Week 14 (July 26 – August 8)	2	- Testing report with metrics; updated prototype with bug fixes and optimizations.
8.	**Final Evaluation and Documentation** - Analyze system performance against baseline/manual methods.	Week 15 – Week 16 (August 9 – August 22)	2	- Complete dissertation draft; project demo video - presentation slides- - code repository and supplemental materials.

5. Literature References

The following are referred journals from the preliminary literature review:

1. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Riedel, S. (2020). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. In *Advances in Neural Information Processing Systems (NeurIPS)*. https://arxiv.org/abs/2005.11401
2. Chen, T., Wang, H., Chen, S., Yu, W., Ma, K., Zhao, X., Zhang, H., & Yu, D. (2024). *Dense X Retrieval: What Retrieval Granularity Should We Use?* In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 15159–15177). Association for Computational Linguistics.

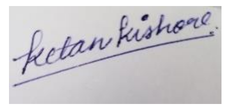
https://aclanthology.org/2024.emnlp-main.845/
3. Zhao, J., Ji, Z., Fan, Z., Wang, H., Niu, S., Tang, B., Xiong, F., & Li, Z. (2025). *MoC: Mixtures of Text Chunking Learners for Retrieval-Augmented Generation*. arXiv preprint arXiv:2503.09600. https://arxiv.org/abs/2503.09600
4. Schick, T., Dwivedi-Yu, J., Raunak, V., Hou, Y., & Lewis, M. (2023). *Toolformer: Language Models Can Teach Themselves to Use Tools*. arXiv preprint arXiv:2302.04761. https://arxiv.org/abs/2302.04761
5. Yao, S., Zhao, J., Yu, D., Zhao, D., & Zhang, R. (2022). *ReAct: Synergizing Reasoning and Acting in Language Models*. In *Advances in Neural Information Processing Systems (NeurIPS)*. https://arxiv.org/abs/2210.03629
6. Masterman, T., Besen, S., Sawtell, M., & Chao, A. (2024). *The Landscape of Emerging AI Agent Architectures for Reasoning, Planning, and Tool Calling: A Survey*. arXiv preprint arXiv:2404.11584. https://arxiv.org/abs/2404.11584
7. Ferrag, M. A., Tihanyi, N., & Debbah, M. (2025). *From LLM Reasoning to Autonomous AI Agents: A Comprehensive Review*. arXiv preprint arXiv:2504.19678. https://arxiv.org/abs/2504.19678
8. Papastratis, I., Konstantinidis, D., Daras, P., & Dimitropoulos, K. (2024). AI Nutrition Recommendation Using a Deep Generative Model and ChatGPT. Scientific Reports, 14(14620. https://www.nature.com/articles/s41598-024-65438
9. LangFlow Documentation. (2025). Welcome to LangFlow – a visual framework for LLM workflows. https://docs.langflow.org/#:~:text=Welcome%20to%20Langflow
10. Streamlit. (2024). Streamlit: The fastest way to build and share data apps. Official website. https://streamlit.io/
11. Johnson, A., Wang, L., & Lin, Y. (2023). *Vector Search: Techniques and Applications*. *ACM Computing Surveys*. https://dl.acm.org/doi/10.1145/3626246.3654691

Supervisor's Rating of the Technical Quality of this Dissertation Outline

EXCELLENT / GOOD / FAIR/ POOR (Please specify): **Excellent**

Supervisor's suggestions and remarks about the outline (if applicable):

The proposed project is well-conceived and addresses a relevant problem in personalized health and wellness using modern AI techniques. The use of multi-agent LLM architecture, RAG, and LangFlow is appropriate and demonstrates a good understanding of current trends. The system design is modular, the methodology is well-structured, and the scope is realistic for execution within the timeline. Overall, the project is technically sound and has strong application value.



Date: 24.05.2025

____
(Signature of Supervisor)

Name of the supervisor: Ketan Kishore

Email Id of Supervisor:ketan.kishore@tsystems.com

Mob # of supervisor: +91 9003221735