



A Project Synopsis on

**Knowy** – "Know why before you know what"

Submitted by: Ganesha Subramanya 1NT23MC020

Under the Guidance of

Mr. Channa Basava Assistant Professor

Department of MCA Academic Year: 2024-25



SUBJECT: PROJECT WORK CODE: 22MCA403P SEMESTER:IV

|  |  |
| --- | --- |
| Title of the Project | **Knowy** – "Know why before you know what" |
| Organization where the project is conducted | Nitte Meenakshi Institute Of Technology |
| Address | Yelahanka Bangalore |
| External Guide Name with Designation, Email, Mobile No. | Channa Basava Assistant Prorfessor |
| Internal Guide Name with Designation | Channa Basava Assistant Professor |



# Brief Description

This project is an AI-powered academic note-taking web application that not only stores notes categorically by subject but also helps students understand the foundational concepts behind what they paste or type. Unlike typical AI tools that directly answer a question, this system identifies and explains the underlying concepts first (e.g., before explaining a C++ inheritance code, it explains OOP basics), and then proceeds to answer the original query. It supports multi-domain learning (programming, science, math, history), includes personalized understanding scores, and organizes memory by subject for long-term review.

# Problem Statement

Many students struggle to grasp academic material due to gaps in foundational knowledge, making it difficult to build upon complex topics. While existing AI tools offer direct answers, they often fail to teach underlying concepts, leaving learners without a deeper understanding. Additionally, traditional study notes tend to be disorganized and lack contextual memory, making revision inefficient. Compounding these issues, students receive little to no feedback on their overall comprehension of a subject. To address these challenges, there is a pressing need for an intelligent, AI-powered system that prioritizes conceptual explanations, enhances note-taking with structured learning pathways, and provides measurable insights into a student’s understanding, bridging the gap between knowledge acquisition and true mastery.



# Objectives

* Build an AI-integrated website for **contextual academic note-taking**.
* Use LLMs to **explain foundational concepts** before addressing the main query.
* Support **multi-subject categorization** (e.g., Math, Science, History, Programming).
* Provide an **"Understanding Score"** per topic and subject.
* Store notes **persistently**, categorized and searchable.
* Integrate a **memory system** that remembers what the user has learned and from which domain.
* Allow optional **quizzes** to improve and validate learning.



# Scope

The project focuses on developing an AI-powered academic assistant that generates foundational explanations for user-provided study material, spanning multiple subjects such as programming, math, and history. Key features include AI-driven concept breakdowns, real-time note storage with auto-tagging, and personalized understanding scores to track learning progress, supplemented by optional quizzes for self-assessment. The system relies on cloud-based LLMs (e.g., Gemini, OpenAI) for scalable processing. However, voice-based tutoring, collaborative workspaces, offline AI processing, and advanced analytics dashboards are excluded from the current scope and reserved for future iterations.



# Methodology

* **Requirement Gathering & Planning**Identify user needs: concept understanding, note organization, progress tracking  
  Finalize tech stack: MERN + Python + Cloud LLMs
* **Design**UI/UX design for intuitive note-taking and review  
  Database schema for note storage, concept tracking, and scores
* **AI Integration**Use Gemini/OpenAI APIs for concept extraction and explanations  
  Implement concept-first answer generation
* **Understanding Score Engine**Calculate scores based on engagement, concept coverage, and quiz results
* **Note Management**Store notes by subject/topic with timestamps and explanation history
* **Testing & Evaluation**Functional, usability, and AI output validation  
  Evaluate using mock student data
* **Hardware Requirements:**System: Any system capable of running Node.js/Python  
  RAM: Minimum 8 GB  
  Processor: Intel i5 or equivalent  
  Storage: Minimum 256 GB  
  Internet: Required for cloud APIs (LLM access**)**
* **Software Requirements:**Node.js + Express: Backend server  
  React.js: Frontend UI  
  MongoDB: Note and memory storage  
  Python: AI logic and score calculation  
  Google AI Studio/OpenAI API: LLM backend for explanations  
  Git & GitHub: Version control  
  Postman: API testing  
  VS Code: Development IDE





# Development Methodology

## Agile Scrum Framework

The project will follow an Agile methodology using the Scrum framework. Work will be divided into fixed-length sprints (e.g., two-week iterations) to iteratively deliver features. Each sprint will include planning, implementation, testing, and review phases.

## Sprint Planning and Review

* + Sprint Planning: At the start of each sprint, I will define the user stories and tasks to work on, based on the project backlog.
  + Sprint Review: At the end of each sprint, I will conduct a self-review of the work completed and evaluate if the sprint goals were met. Feedback will be collected from peers or mentors (if applicable) to refine the project further.

## Daily Standups

* + Daily Check-ins: Even as a solo developer, I will hold short daily self- check-ins. During these, I will review my progress, identify any blockers, and plan tasks for the next day. This can be done through a brief log or journal.



* + Continuous Integration (CI): I will set up automated testing pipelines to run tests and builds with each commit, ensuring that defects are caught early, and the codebase remains deployable.

## User-Focused Design

* + UI/UX Design: I will continuously refine the design based on feedback from potential users or peers. Prototyping and usability evaluations will be incorporated into early sprints to ensure the journal and conversational interfaces are intuitive and effective.

## Documentation

* + I will maintain clear and up-to-date technical documentation throughout the project, including architecture diagrams, API specifications, and code comments. This documentation will be updated regularly to reflect the evolving state of the project