**CODING NINJAS ASSIGNMENT**

**(AI Multi-Agent Project)**

**MULTI-AGENT AI TUTOR**

**Problem Statement ->**

Online students often struggle with self-paced learning due to lack of structured guidance, difficulty understanding complex topics, unanswered doubts, and low motivation. Existing online resources are scattered, and students cannot get personalized support like a human tutor.

**My solution** is a **Multi-Agent AI Tutor** that simulates a personal mentor. Each AI agent specializes in a specific task such as planning study schedules, simplifying complex concepts, answering doubts, motivating learners, curating high-quality resources, and evaluating understanding. By working together, these agents provide a **complete, interactive, and personalized learning experience**, addressing the common challenges online students face.

**Why AI Agents Are Suitable & Value of Multi-Agent Collaboration ->**

AI agents are a great fit because they can **quickly provide guidance, answer questions, and adapt explanations to each student’s level**, something that’s hard to do manually for every learner. They can work anytime, anywhere, giving students instant support.

Having **multiple specialized agents** makes the system even better:

****Planner Agent**** organizes study schedules so learning is efficient.

****Simplifier Agent**** explains tough topics in an easy way.

****Doubt Solver Agent**** answers questions whenever a student is stuck.

****Motivator Agent**** keeps students encouraged and engaged.

****Resource Curator** Agent** finds useful learning material.

****Evaluator Agent**** checks understanding with quizzes and feedback.

Together, these agents act like a small team of tutors, each focusing on a different task but working together to make learning smoother, more personalized, and less stressful for students.

**Application Overview ->**

I have built **Multi-Agent AI Tutor**, an intelligent online learning assistant for students. The application acts like a personal team of tutors, guiding learners through their study journey. It provides structured study plans, simplifies difficult topics, answers doubts in real-time, curates useful learning resources, motivates students, and tests their understanding.

The goal of the application is to make online learning **easier, more interactive, and personalized**, so students can study efficiently without feeling lost or overwhelmed. It combines multiple AI agents that collaborate to provide a **complete learning experience**, simulating the support a human tutor would offer.

**How different agents react ->**

Each agent in the system has its own role but they work together to provide a smooth learning experience:

The **Planner Agent** organizes what the student should study and in what order.

The **Simplifier Agent** explains the topics chosen by the Planner in simple, easy-to-understand language.

The **Doubt Solver Agent** answers any questions the student has while studying, ensuring no confusion slows them down.

The **Resource Curator Agent** provides helpful materials like articles, videos, or tutorials relevant to the current topic.

The **Motivator Agent** sends encouraging messages to keep the student engaged and confident.

The **Evaluator Agent** tests the student’s understanding with quizzes and provides feedback to adjust the study plan.

These agents **collaborate intelligently**: for example, after the Planner sets a study topic, the Simplifier explains it, the Resource Curator suggests supporting material, and the Doubt Solver answers any immediate questions. Meanwhile, the Motivator keeps the student engaged, and the Evaluator tracks progress to inform future planning.

This collaboration ensures that students have **personalized guidance, real-time support, and a complete learning experience**, just like having a small team of human tutors working together.

**Tools, Libraries, and Frameworks Used ->**

**Python 3.9+** – The main programming language used to build the agents and the overall system.

**Gemini API** – The AI engine powering all agents, generating explanations, answering doubts, and providing content in a human-like manner.

**Streamlit** – Used to create a simple and interactive web interface for students to interact with the AI tutor.

**dotenv** – Manages environment variables like API keys securely without exposing them in the code.

**Standard Python libraries** – Including os, time, and random for basic functionality and scheduling.

**Agent Frameworks and Interaction ->**

Each agent is implemented as an independent module, allowing it to operate on its own tasks.

Agents communicate indirectly via shared input/output data. For example, the Planner determines the topic, which is then passed to the Simplifier and Resource Curator agents.

The system uses a **modular orchestration approach**, where each agent is loosely coupled but collaborates to complete the overall learning process efficiently.

This setup ensures the system is **scalable, maintainable, and flexible**, so new agents or features can be added easily in the future.