Project Work Task Breakdown

Phase 1: Requirement Analysis & Planning

Task 1.1: Define project scope, objectives, and features.

Task 1.2: Collect references of trusted tutorial sources (YouTube, edX, Coforge, Blogs, etc.).

Task 1.3: Select appropriate APIs and web scraping tools (BeautifulSoup, Requests, Selenium, YouTube API, etc.).

Task 1.4: Identify suitable LLM model for NLP (OpenAI, Hugging Face, or other open-source LLMs).

Phase 2: System Design

Task 2.1: Design the system architecture (frontend, backend, database, AI module).

Task 2.2: Prepare ER Diagram / Database Schema for storing tutorials, tags, user history.

Task 2.3: Create UI/UX wireframes (Search bar, Results page, Save/History section).

Task 2.4: Define block diagram & workflow diagram.

Phase 3: Backend Development (Flask)

Task 3.1: Set up Flask project structure.

Task 3.2: Implement routing for: Homepage, Search Results, User Dashboard.

Task 3.3: Connect Flask with database (MongoDB / PostgreSQL).

Task 3.4: Build REST APIs for tutorial search & retrieval.

Phase 4: Data Collection (Web Scraping & APIs)

Task 4.1: Implement scraping scripts for platforms like Coforge, Blogs.

Task 4.2: Integrate APIs (YouTube Data API, edX API).

Task 4.3: Normalize & clean fetched tutorial data (title, description, duration, source link).

Task 4.4: Store tutorials in database with metadata (platform, difficulty, tags).

Phase 5: LLM & AI Integration

Task 5.1: Train/Use LLM model for: Understanding natural language queries, Generating tutorial tags, Difficulty classification (Beginner/Intermediate/Advanced).

Task 5.2: Implement recommendation system using AI (relevance, popularity, difficulty).

Task 5.3: Test model responses on sample queries.

Phase 6: Frontend Development

Task 6.1: Develop UI in Flask (Jinja templates / Bootstrap / Tailwind CSS).

Task 6.2: Create search interface (search bar, filters).

Task 6.3: Display results with ranking, difficulty, tags.

Task 6.4: Add Save for Later / Watch History feature.

Phase 7: Testing & Validation

Task 7.1: Unit testing of scraping modules.

Task 7.2: Integration testing of Flask backend + database.

Task 7.3: User acceptance testing (search queries & recommendations).

Task 7.4: Debugging & performance optimization.

Phase 8: Deployment & Documentation

Task 8.1: Deploy project on cloud (Heroku / AWS / PythonAnywhere).

Task 8.2: Write detailed project report (Introduction, Literature Review, Methodology, Results, Conclusion).

Task 8.3: Prepare user manual (how to use the website).

Task 8.4: Final presentation & demonstration.