Edu-Tutor AI: Project Report

---

1. INTRODUCTION

1.1 Project Overview:

Edu-Tutor AI is an intelligent, web-based learning assistant that combines generative AI, adaptive learning, and LMS integration to offer a personalized educational experience. It aims to support both students and tutors by automating content recommendations, tutor matching, and performance tracking.

1.2 Purpose:

To enhance learning outcomes and tutor efficiency by integrating AI into the educational workflow, enabling real-time personalization, improved accessibility, and scalable deployment.

---

2. IDEATION PHASE

2.1 Problem Statement:

Current Learning Management Systems (LMS) deliver generic content, leading to poor engagement and retention. Tutors are unable to provide individualized support at scale, and students lack adaptive learning paths.

2.2 Empathy Map Canvas:

- **Think & Feel**: Frustrated with irrelevant content
- **See**: Repetitive, unengaging materials
- **Say & Do**: Search for YouTube/Google resources
- **Pain**: No real-time tutor support or progress tracking
- **Gain**: Wants customized learning and faster results

2.3 Brainstorming:

Ideas generated included AI-based quiz generation, subject-based tutor recommendations, LMS analytics, gamified learning, and real-time feedback tools.

---

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map:

1. Visit site → 2. Register/Login → 3. Take placement quiz → 4. Get content/tutor → 5. Receive feedback & analytics

3.2 Solution Requirements:

- **Functional Requirements**:
  - User registration (Gmail, OTP)
  - AI quiz generation
  - Tutor matching
  - Progress dashboard
- **Non-Functional Requirements**:
  - Usability, performance, scalability, security

3.3 Data Flow Diagram:

User → UI → Flask App → AI Module + Database → LMS API

3.4 Technology Stack:

- **Frontend**: HTML/CSS/JS
- **Backend**: Python (Flask)
- **AI**: OpenAI GPT / Custom ML
- **Database**: MongoDB
- **Deployment**: Render/Heroku/AWS

---

4. PROJECT DESIGN

4.1 Problem – Solution Fit:

Our AI solution matches the pain points of learners and tutors with intelligent personalization, real-time analytics, and automation. (Ref: ideahackers.network)

4.2 Proposed Solution:

Edu-Tutor AI delivers: - Personalized AI content - Tutor matching - Quiz generation - Progress tracking - Seamless deployment and integration

4.3 Solution Architecture:

Includes Frontend UI, Flask server, AI modules, MongoDB database, and LMS API integration. Data flows through secure channels with dynamic content delivery. (See AWS voice app architecture adaptation)

---

5. PROJECT PLANNING & SCHEDULING

5.1 Sprint Planning:

- **Sprint 1**: Registration, login, basic dashboard (5 Story Points)
- **Sprint 2**: AI modules, tutor matching, deployment (11 Story Points)
- **Velocity**: 8 Story Points/sprint

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing:

- Load tested for 50 concurrent users
- Average response time < 1.5s
- No data loss under simulated stress

---

7. RESULTS

7.1 Output Screenshots:

- Registration form
- AI-generated quiz
- Matched tutor dashboard
- Analytics dashboard

---

8. ADVANTAGES & DISADVANTAGES

Advantages: - Personalized learning - Scalable tutor management - Improved engagement

Disadvantages: - Initial development cost - AI dependency for quiz quality

---

9. CONCLUSION:

Edu-Tutor AI successfully bridges the gap between traditional LMS and AI-powered personalization, offering measurable value to both learners and educators.

---

10. FUTURE SCOPE:

- Multilingual content support
- Mobile app version
- Integration with external content providers
- Gamified modules and leaderboards

---

11. APPENDIX:

- **Source Code**: [To be provided via GitHub]
- **Dataset Link**: [Used for quiz training]
- **Demo Link**: [Link to deployed project]