**EduTutor AI - Detailed Project Documentation**

**1. INTRODUCTION**

**1.1 Project Overview**

EduTutor AI is a personalized learning platform designed to enhance student engagement and educator efficiency through AI-powered quizzes, progress analysis, and adaptive feedback. Built with Streamlit, the platform supports real-time evaluation, role-based dashboards, and interactive educational tools. It utilizes rule-based logic and curated question banks to deliver context-aware assessments and personalized learning journeys.

**1.2 Purpose**

The purpose of EduTutor AI is to bridge learning gaps by delivering adaptive assessments and personalized feedback to students based on their past performance. It aims to empower educators with class-wide analytics while enabling students to monitor their own progress and learning paths. With an intuitive interface and role-based access, EduTutor AI promotes transparency, continuous learning, and skill development.

**2. IDEATION PHASE**

**2.1 Problem Statement**

**Student:**

* Wants personalized help in weak areas and real-time feedback.
* Feels lost in generic, one-size-fits-all assessments.
* Lacks motivation due to non-interactive learning methods.

**Educator:**

* Wants to monitor student progress efficiently.
* Struggles with creating personalized quizzes.
* Needs fast insight into student learning trends.

**2.2 Empathy Map**

**Thinks/Feels:**

* "How can I improve my performance on topics I’m weak in?"
* "How do I keep my class engaged and progressing together?"

**Says/Does:**

* Students: "These quizzes feel repetitive."
* Educators: "I need quick insight into class performance."

**Pains:**

* Students: Unengaging learning experiences.
* Educators: Time-consuming quiz creation and tracking.

**Gains:**

* AI-generated quizzes, progress charts, feedback-based recommendations.

**2.3 Brainstorming**

* Adaptive quiz generation
* Student and educator dashboards
* AI-powered performance evaluation
* Diagnostic assessments
* Feedback-based progress recommendations

**Final Modules:**

* Login & Registration
* Quiz Generation
* Evaluation and Feedback
* Progress Tracking
* Analytics Dashboard

**3. REQUIREMENT ANALYSIS**

**3.1 Customer Journey**

* User visits login page → registers/logs in
* Chooses a subject and topic → receives a quiz
* Submits quiz → sees feedback and performance level
* Can review progress charts and improvement tips

**3.2 Functional Requirements**

* FR-1: User authentication (login, registration)
* FR-2: Quiz generation (topic + difficulty)
* FR-3: Performance evaluation with feedback
* FR-4: Student progress dashboard
* FR-5: Educator analytics overview
* FR-6: Adaptive quiz generation based on history

**3.3 Non-Functional Requirements**

* NFR-1: Usability: Simple and intuitive Streamlit interface
* NFR-2: Security: Hashed passwords using Werkzeug
* NFR-3: Performance: Quiz generation under 2 seconds
* NFR-4: Reliability: Handles DB reconnects and reruns
* NFR-5: Availability: Deployed via Streamlit cloud with minimal downtime

**3.4 Data Flow Diagram**

* User → UI → Backend (quiz, DB, feedback engine) → Response (feedback, dashboard)
* Admin → UI → Dashboard analytics → View student KPIs

**Entities:** Student, Educator, QuizEngine, DB

**Level 0:** Generate quiz → Submit answers → Evaluate → Save → Display progress

**3.5 User Stories**

| **User** | **Requirement** | **User Story** | **Acceptance Criteria** | **Priority** |
| --- | --- | --- | --- | --- |
| Student | Quiz | As a student, I want to take quizzes so I can test my knowledge | Quiz is generated and submitted with feedback | High |
| Student | Progress | As a student, I want to see my progress to track improvements | Score chart and tips shown | High |
| Educator | Analytics | As an educator, I want to monitor class performance | Dashboard with scores, attempts | High |

**3.6 Technology Stack**

| **Component** | **Technology** |
| --- | --- |
| Frontend UI | Streamlit, Plotly |
| Backend | Python, SQLite |
| Authentication | Werkzeug (password hashing) |
| Database | SQLite (edututor.db) |
| AI Logic | Rule-based, static question banks |
| Hosting | Localhost / Streamlit Cloud |

**4. SOLUTION DESIGN**

**4.1 Emotions: Before / After**

| **Phase** | **Emotion** |
| --- | --- |
| Before | Confused, overwhelmed, uninformed |
| After | Informed, empowered, connected |

**Offline User Research:** Conducted via community meetings, poster campaigns, and surveys.

**4.2 Proposed Solution**

| **Parameter** | **Description** |
| --- | --- |
| Problem | Students and educators lack personalized assessments and real-time feedback tools. |
| Solution | AI-powered EduTutor platform using rule-based quiz engine, Streamlit UI, and adaptive performance evaluation. |
| Novelty | Combines diagnostic, adaptive testing, and feedback within one app. No internet/LLM dependency required. |
| Social Impact | Improves academic engagement and personalizes student learning experience. Saves educators time and enhances outcomes. |
| Business Model | Free open-source project, future potential as SaaS tool for institutions. |
| Scalability | Easily scales via cloud hosting and modular DB structure. |

**4.3 Solution Architecture**

* Frontend: Streamlit (UI, Tabs, Widgets, Graphs)
* Backend: Python logic + SQLite database
* AI Logic: Rule-based adaptive quiz generator
* State: Managed via st.session\_state
* Storage: Local database (edututor.db), secrets via .env

**5. PROJECT PLANNING & SCHEDULING**

**5.1 Sprint Planning**

| **Sprint** | **Task** | **Story Points** | **Assigned To** | **Priority** |
| --- | --- | --- | --- | --- |
| Sprint-1 | Setup DB & user auth | 3 | Durga Lakshmi | High |
| Sprint-1 | Create static quiz generator | 4 | Durga Lakshmi | High |
| Sprint-1 | Build basic UI for login/register | 3 | Durga Lakshmi | Medium |
| Sprint-2 | Implement adaptive quiz & evaluation | 5 | Durga Lakshmi | High |
| Sprint-2 | Design educator dashboard | 3 | Durga Lakshmi | High |
| Sprint-2 | Add visualizations & feedback system | 4 | Durga Lakshmi | Medium |

**Sprint Velocity:**

* Total Story Points = 22
* Sprint Velocity = 11 points/sprint (avg)

**6. FUNCTIONAL AND PERFORMANCE TESTING**

**6.1 Performance Testing**

| **Test ID** | **Scenario** | **Expected** | **Actual** | **Result** |
| --- | --- | --- | --- | --- |
| FT-01 | Quiz load time | Under 3s | ~2s | Pass |
| FT-02 | Login/Register process | Form validation and session start | Working as expected | Pass |
| FT-03 | Adaptive quiz accuracy | Difficulty adjusts by user score | Correct level selected | Pass |
| PT-01 | Feedback delivery speed | <1s | <1s | Pass |
| PT-02 | Concurrent usage | No crashes | Smooth operation | Pass |
| PT-03 | Graph rendering | Must render charts without lag | Plots load instantly | Pass |

**7. RESULTS**

**7.1 Output Screenshots**

(Add annotated screenshots of: Login Page, Quiz Page, Feedback Results, Analytics Dashboard)

**8. ADVANTAGES & DISADVANTAGES**

**Advantages**

1. **AI-Powered Insights** – Automates quiz generation and feedback.
2. **Policy Simplification** – Easily integrates with educational or civic content.
3. **Enhanced Engagement** – Personalized feedback improves learning outcomes.
4. **Data-Driven** – Progress analytics inform educators.
5. **Modular & Scalable** – Streamlit-based micro-modules.

**Disadvantages**

1. **Offline Limitation** – Requires local runtime setup.
2. **No LLM Support Yet** – Currently rule-based, not LLM integrated.
3. **Limited Language Support** – English-only UI.
4. **Static Question Bank** – Manual curation required.
5. **Basic Security** – No advanced auth or encryption.

**9. CONCLUSION**

EduTutor AI is an impactful educational platform built for students and educators seeking personalized, adaptive, and interactive learning experiences. It showcases how a well-designed, non-LLM solution can still deliver high-quality assessments and insights. With modular components and open architecture, the system is future-ready for LLM integration, mobile deployment, and broader institutional adoption.

**10. FUTURE SCOPE**

1. **Mobile App** – Build Flutter app for broader reach.
2. **LLM Integration** – Add GPT/Granite for smarter quizzes.
3. **IoT Integration** – Link with attendance or biometrics.
4. **Multi-language UI** – Extend to regional languages.
5. **Blockchain Security** – Safeguard user analytics.
6. **Dynamic Content Upload** – Allow educators to input custom material.
7. **Gamified Learning** – Reward systems and badges.

**11. APPENDIX**

**Source Code Snippet**

# Sample Quiz Evaluation Logic

from ai\_quiz import AIQuizGenerator

quiz\_engine = AIQuizGenerator()

questions = quiz\_engine.generate\_quiz("Algebra", difficulty\_level=2, subject="mathematics")

user\_answers = [0, 2, 1, 1, 3]

feedback = quiz\_engine.evaluate\_answers(questions, user\_answers)

print(feedback["performance\_level"])

**End of Document**