MACAU UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Computer Science and Engineering

Faculty of Innovation Engineering

<<Proposal for Software Project Management >>

Homework ID : Project Proposal

Report Title : AI Study Buddy_Browser Extension

Student Name : Zhaoyi, Liushixian, Wanghaoyang

Student No. : 1220005384、1220004128、1220001359

Date : 2025/2/10

1.Background

Project Purpose: AI-powered browser extension for students, integrating error analysis, vocabulary learning, and knowledge review.

Core Problem: Students often struggle with fragmented learning tools and lack personalized feedback during online study.

Technical Foundation: Combines lightweight browser extension architecture (HTML/CSS/JS) with AI services (OpenAI, Hugging Face) and 3D visualization (Three.js).

2. Motivation

User Needs: Students: Unified tool for error debugging, vocabulary building, and progress tracking.

Educators: Efficient quiz/question generation for classrooms.

Value Proposition:

Democratize AI-driven personalized learning.

Freemium model balancing accessibility and sustainability.

3. Revised Schedule

Weeks 1-2: Planning & Design

User research, AI/3D technical specs finalization, and UI prototyping for key features.

Weeks 3-4: Frontend Core

Base extension setup, text capture/highlight system, and vocabulary dashboard.

Weeks 5-6: Backend & AI

Deploy error/vocabulary APIs, integrate spaced repetition, and enable data sync.

Week 7: Advanced Features

3D model visualization, interactive exercises, and cross-device sync.

Week 8: Launch Prep

Stress testing, performance optimization, and Chrome Web Store packaging.

5. Team Members (3)

1. **Frontend Developer**: Extension UI/UX, text highlighting logic, 3D model rendering (Three.js).
2. **Backend Developer**:API development (FastAPI), AI integration, SQLite database design.
3. **Full-stack Developer**:Cross-module integration, testing, deployment pipeline.

6. How to Satisfy "Time-Cost-Quality"?

Time: Accelerate timelines via pre-trained AI models (OpenAI/Hugging Face) and parallel frontend/backend development.

Cost: Minimize expenses using free-tier APIs (e.g., Wordnik) and lightweight infrastructure (SQLite/IndexedDB).

Quality: Focused prioritization of core features with modular design allows optimal trade-off balancing.

7. Possible Risks

1. AI API Latency in Real-World Usage

Mitigation: Implement request queuing with client-side caching for frequent queries (e.g., common math errors). Use distilled smaller models (e.g., TinyBERT) as fallback.

2. Browser Compatibility Issues

Mitigation: Rigorously test on Chromium variants (Chrome, Edge, Brave) during Week 8. Provide a degraded mode for unsupported browsers.

3. Vocabulary Data Loss During Sync

Mitigation: Design a versioned backup system with manual restore points. Use checksums to detect corruption.

4. Overwhelming User Feedback Post-Launch

Mitigation: Prepare a priority framework for feature requests (e.g., "Bug fixes > Performance > New features"). Use automated tagging for user-submitted issues.