# AI-Study Chrome Extension

Version: 1.0  
Date: 2025.3.12

## 1. Project Objectives

Develop a Chrome browser extension leveraging AI to enhance vocabulary learning efficiency through:

• Contextual Learning: Automatically generate example sentences, synonyms, and exercises after users highlight text on webpages.

• Personalized Recommendations: Dynamically adjust review schedules based on user progress and forgetting curves.

• Cross-Device Sync: Ensure seamless data synchronization across devices via AWS cloud services.

## 2. Key Deliverables

|  |  |  |
| --- | --- | --- |
| Deliverable | Description | Priority |
| Functional System | Core Chrome extension features (text highlighting, AI-generated content, reminders). | High |
| AI Service System | OpenAI API-integrated vocabulary analysis engine deployed on AWS Lambda. | High |
| Data Sync System | Local storage (IndexedDB) + AWS cloud sync with conflict resolution. | Medium |
| User Documentation | Installation guide and feature documentation (Markdown format). | Low |

## 3. Project Boundaries

In-Scope:  
• Chrome extension development (prioritizing Chrome, compatible with Chromium-based browsers).  
• AI services and local/cloud data synchronization.  
• User testing and performance optimization (response time ≤ 2 seconds).

Out-of-Scope:  
• Mobile app development (iOS/Android).  
• Multilingual support (English vocabulary only).  
• Enterprise-level user management (e.g., team collaboration).

## 4. Acceptance Criteria

Functional Acceptance:  
• Users can highlight words on any webpage and generate AI explanations (via OpenAI API).  
• Local offline storage (IndexedDB) with cloud sync error rate < 1%.  
• Daily review reminder accuracy ≥ 95%.

Technical Acceptance:  
• AWS Lambda service latency ≤ 500ms (90% of requests).  
• Extension memory usage ≤ 50MB.

## 5. Project Dependencies & Critical Path

Dependencies:  
• Technology Selection Report (Document D112) must be finalized before feature development (B2).  
• Local Storage Optimization (C24) directly impacts sync functionality (B5→B6).

Critical Path:  
Technical Validation (B1) → Core Feature Development (B2) → Algorithm Optimization (B3) → Testing & Release (B7).

Flowchart Color Coding:  
• Features: Blue  
• Milestones: Orange  
• User Stories: Green  
• Tasks: Gray

## 6. Constraints & Assumptions

Constraints:  
• Budget: Monthly AWS service cost ≤ $200.  
• Timeline: Must complete release by April 16.

Assumptions:  
• OpenAI API stability meets requirements (error rate < 5%).  
• Timely recruitment of user testers (at least 20 students).

## 7. Scope Change Control

All changes require submission via a Change Request Form, evaluated by the PM and technical lead for:  
• Impact on existing features (e.g., disruption to the critical path).  
• Resource and timeline adjustments (update Gantt charts accordingly).  
• Approved changes must be reflected in the Notion documentation scope baseline.

## Attachments

Flowchart References:  
• Technical validation (B1) depends on Document D112; local storage optimization (C24) links to sync functionality (B6).  
• Use the color scheme from Document 2 to prioritize deliverables (High/Medium/Low).