ASPS SPMP Adaptive Study Planning System Software Project Management Plan

1. Overview

1.1 Introduction to ASPS The Adaptive Study Planning System (ASPS) is a decision-support software system designed to assist students in planning and managing their academic preparation dynamically. Unlike static planners or fixed timetables, ASPS adapts study plans over time by considering syllabus structure, topic difficulty, learning decay, and learner engagement.

The core objective of ASPS is to help students decide what to study next given limited time, syllabus importance, and their current learning state. The system is developed as part of the B.Tech minor project with scope for further extension.

2. System Structure and Architecture

ASPS follows a modular layered architecture to ensure clarity, scalability, and maintainability.

2.1 High-Level Architecture User Interface (Frontend) API Layer (Node.js / FastAPI) Decision Engine (Python – ASPS Core) Data Layer (JSON / Database – Planned)

2.2 Frontend Layer (Proposed) Purpose: - Collect user inputs such as syllabus sources and study availability - Display structured syllabus, checklists, and daily plans - Provide a clean and simple interface

Proposed Technologies: - React.js or basic web frontend - Markdown or checklist-based rendering

Current Status: Not implemented

2.3 Backend / API Layer (Planned) Purpose: - Handle communication between frontend and decision engine - Manage requests and responses

Proposed Technologies: - Node.js as API gateway - Python FastAPI for decision logic exposure

Current Status: Architecture finalized, implementation pending

2.4 Decision Engine (Implemented) Purpose: - Core logic for adaptive planning - Priority calculation - Daily plan generation - Learning decay updates

Technology Used: - Python 3.12

Current Status: Implemented and tested locally

2.5 Data Layer (Planned) Current Approach: - In-memory structures and JSON files

Planned: - Database integration for persistence and analytics

3. Core Functional Design

3.1 Priority Calculation Each topic is assigned a priority score based on: - Credits (syllabus importance) - Complexity (difficulty level) - Time since last revision - Exposure level - Optional test performance

3.2 Learning Decay Topics not revised for long durations automatically gain higher priority, while recently studied topics are deprioritized.

3.3 Adaptive Planning Loop Plan generation follows a continuous loop: Plan $\rightarrow$ Study $\rightarrow$ Update $\rightarrow$ Replan

4. Ingestion and Syllabus Handling (Planned)

ASPS aims to provide a single structured representation of syllabus data.

Proposed pipeline: - Accept syllabus input (PDF, DOCX, text, links) - Extract plain text - Assist in structuring units and topics - Store confirmed syllabus in structured format

5. Machine Learning Integration (Planned)

ML will be used only to assist in estimating topic difficulty. It will not replace the core planning logic.

6. Progress Report

Completed: - Core planning engine implemented - Priority calculation refined - Learning decay mechanism added - Engine structured for API use - Version-controlled repository set up

7. Pending Discussions - Final syllabus data model - Frontend interaction design - Evaluation metrics - Database strategy

8. Task Checklist

To be completed: - Ingestion prototype - API implementation - ML difficulty estimator - Frontend integration - Documentation and presentation

9. Conclusion

ASPS is being developed as a structured and adaptive study planning system with a clear separation of components. The current progress establishes a strong foundation for further development.