

**CLARK UNIVERSITY**

**MSCS-3999**

**CAPSTONE**

**PROJECT PLAN**

*for*

**Team – 6**

**Smart StudyMate – AI-Powered Learning Assistant**

Submitted By:

Vikas Falke

Kevin Joseph

Sophiya Baniya

**1. PROJECT BACKGROUND**

**1.1 Problem / Opportunity Description**

**Problem:**  
In the present day of digital learning, students, faculty, and organizations face the overwhelming challenge of managing vast quantities of study materials, lecture notes, research papers, and online references. While accessibility to such information can be valuable, it has also led to information overload. Traditional study methods remain draining, tedious, and time-consuming. Learners often struggle to filter important insights buried within lengthy documents, leading to reduced productivity, poor knowledge retention, and difficulty in preparing effectively within limited timeframes. These barriers are particularly concerning for students preparing for exams and for organizations involved in education, training, or professional development, where inefficiency directly impacts outcomes.

**Opportunity:**  
With advancements in artificial intelligence (natural language processing, summarization, and video generation), there is a unique opportunity to transform studying into a personalized, interactive, and engaging process. Instead of passively consuming text, learners could receive concise summaries, explanatory videos, and real-time interactive support. This approach would not only reduce information overload but also help learners comprehend and retain knowledge more efficiently. For institutions and educators, it presents an opportunity to enhance digital transformation initiatives, support student success, and build innovative reputations while streamlining repetitive tasks.

**1.2 Benefits**

* **For Students / Learners:** Faster revision through concise summaries, real-time interactive Q&A, and video-based explanations that make studying engaging and personalized.
* **For Educators / Faculty:** Reduced repetitive workload by offloading tasks such as summarizing and explaining basics, enabling more time for mentoring, research, and curriculum development.
* **For Institutions:** Improved student engagement, alignment with organizational goals of digital transformation, and enhanced academic performance outcomes. Institutions may also gain reputational benefits by adopting innovative educational technologies.
* **For EdTech & Organizations:** Opportunities to collaborate, integrate, or monetize through licensing and premium features, creating new revenue streams while supporting internal efficiencies.

**1.3 Goals**

The overall goal of the project is to develop an **AI-powered learning assistant** that improves efficiency, productivity, and comprehension. The platform will:

* **Build a web-based solution** that converts study materials into interactive notes and explanations.
* **Integrate an AI tutor for real-time Q&A,** enabling learners to clarify concepts directly from their uploaded materials.
* **Provide summarization tools** that automatically condense lengthy documents into concise revision notes with key takeaways.
* **Generate video explanations** from study materials, turning complex concepts into narrated, engaging, and interactive learning experiences.
* **Deliver a functional MVP** within the project timeline, focusing on the features most impactful to student learning.

**1.4 Stakeholders and Clients**

* **Primary Stakeholders (direct users):**
  + **Students / Learners** who will directly use the platform for faster, more interactive, and more personalized study experiences.
  + **Teachers / Faculty / Trainers** who can recommend the tool, use it to support student learning, and benefit from reduced repetitive teaching tasks.
* **Secondary Stakeholders (indirect users):**
  + **Educational Institutions** that may adopt the tool to support student success, integrate it into digital learning strategies, or share best practices.
  + **EdTech Companies** as potential collaborators or technology partners who could assist in integration, provide technical support, or extend the solution commercially.
  + **Libraries & Learning Resource Centers** that could host or recommend the tool for students seeking academic support.
* **Clients (initial focus):**
  + **University and college students** preparing for exams, managing heavy course loads, and seeking efficient study aids.

**2. PROJECT SCOPE**

The focus of this project will be in the academic and training contexts of universities, colleges, and educational institutions. Smart StudyMate is designed to support students by transforming textbooks and notes into more interactive and accessible learning resources. While the project could eventually scale to other institutions or edtech partners, the initial MVP will prioritize functionality for individual learners.

The scope does **not** include creating non-academic content, implementing live tutoring, or generating fully personalized curricula for every uploaded material. Instead, the project centers on summarization, interactive Q&A, and video-based explanations built on top of uploaded study documents.

**2.1 Objectives**

* **Enable Upload of Study Materials:** Allow students to upload PDFs and text-based notes for processing.
* **AI-Powered Summarization and Q&A:** Provide concise summaries of key content and enable interactive Q&A directly from uploaded documents using retrieval-augmented generation (RAG).
* **Video Explanations:** Transform summaries and explanations into narrated video formats for easier understanding of complex concepts.
* **Improved Learning Efficiency:** Help students access and comprehend study materials quickly, reducing manual note-taking and reading time.
* **Enhanced Retention and Engagement:** Use multimodal outputs (text + video + interactive chat) to reinforce knowledge retention and make learning more engaging.
* **Build Scalable System:** Create a user-friendly React-based frontend and a scalable Python backend with PostgreSQL for document handling and AI processing.

**2.2 Deliverables**

**MVP:**

* **File Upload & Summarization Module**
  + Extract text from PDFs and generate concise summaries highlighting key ideas.
  + Text processing pipeline (chunking/embedding) to quickly retrieve relevant sections.
  + Dashboard/Interface for uploads and summaries.
* **Interactive Q&A Chatbot**
  + AI-powered Q&A system using RAG to answer student queries from uploaded documents.
  + Backend mechanisms to process queries and return reliable, context-aware responses.
  + Logging and reporting features to monitor usage and question patterns (future extension).
* **Video Explanation Feature**
  + Module to convert generated summaries and explanations into narrated video lessons.
  + Accessible via the frontend dashboard alongside text summaries and Q&A interface.

**Documentation:**

* User guide.
* Technical documentation.
* Final project report.

**2.3 Out of Scope**

* **Integration with External LMS (Learning Management Systems):** Connecting to platforms like Moodle, Blackboard, or Canvas is not included in the MVP but may be a future enhancement.
* **Live Tutoring or Human-Led Assistance:** The assistant will only provide automated responses; real-time mentoring is excluded.
* **Personalized Lesson Plans:** No auto-generated curricula beyond processing uploaded materials.
* **Offline or Local Installation:** The system will run as a web-based cloud platform only; desktop/local deployment is not planned.
* **Advanced Gamification:** Features like leaderboards, streaks, and XP points are excluded from the MVP.

**3. PROJECT PLAN**

**3.1 Approach and Methodology**

The project will follow an **Agile Scrum methodology**, ensuring iterative development and continuous feedback. The work will be divided into 2-week sprints, with each sprint delivering incremental progress toward the MVP. This approach allows the team to adapt to challenges, refine features based on testing, and maintain steady momentum.

Collaboration will be supported through modern project management and development tools:

* **GitHub** for version control and collaborative coding.
* **Jira** for sprint planning, backlog management, and task tracking.

The **technical stack** will include:

* **Frontend:** React – chosen for its component-based architecture and interactive UI capabilities.
* **Backend:** Python – selected for its strong ecosystem and compatibility with AI/ML libraries.
* **Database:** PostgreSQL – an open-source relational database for reliable data storage.
* **AI Models:** HuggingFace pre-trained models – leveraged for summarization, document Q&A, and NLP tasks.

**3.2 Project Timeline (10 Weeks)**

The project will be executed in **parallel tracks** (frontend, backend, and database) to maximize efficiency and teamwork.

* **Week 1:** Requirements gathering, high-level system design, and environment setup.

**Sprint 1:**

* **Weeks 2–4:**
  + **Frontend:** Dashboard design, file upload module, and chatbot interface implementation.
  + **Backend:** Development of document parsing, summarization, and interactive Q&A.
  + **Database:** Setup of PostgreSQL schema, initial integration with backend services.

**Sprint 2:**

* **Weeks 5–6:**
  + **Frontend:** Development of notes display and video explanation panels.
  + **Backend:** Refinement of summarization and Q&A logic.
  + **Database:** Optimization of queries and integration with backend services.

**Sprint 3:**

* **Weeks 7–8:**
  + Development of the video explanation feature.
  + End-to-end testing of frontend, backend, and database workflows.

**Sprint 4:**

* **Week 9:** Full integration of all modules, bug fixing, and quality assurance testing.
* **Week 10:** Preparation of documentation, deployment and presentation.

**3.3 Success Criteria**

* The MVP features (upload, summarization, Q&A, video explanations) are functional.
* The AI Q&A achieves at least 80% accuracy when answering questions compared to the uploaded content.
* The video explanation feature successfully generates coherent and accessible narrated outputs.
* Students and testers provide positive feedback, indicating that the tool improves learning efficiency and makes revision easier.

**3.4 Issues and Policy Implications**

The project must address several policy and ethical considerations:

* **Ethical Use of AI:** All responses must be grounded in uploaded material to avoid plagiarism and misinformation. The system will cite sources when possible.
* **Data Privacy:** Uploaded documents will be stored securely in PostgreSQL with restricted access. Sensitive data will not be shared with third parties.
* **Accessibility:** The platform will adopt universal design principles, such as a clean, distraction-free interface, adequate color contrast, large fonts, and mobile-friendly layouts, to support diverse learners.

**3.5 Risk Management Plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk Category** | **Risk Scenario** | **Potential Impact** | **Probability** | **Impact** | **Mitigation Strategy** |
| **Technical** | **AI Model Performance Issues:** The selected HuggingFace models for summarization or Q&A produce low-quality results or are too slow, leading to a poor user experience. | Degraded core functionality, user frustration, failure to meet project goals. | Medium | High | **Mitigation:** Test multiple pre-trained models early in the development cycle. Start with smaller, faster models and scale up if necessary. Implement asynchronous processing for AI tasks so the user interface doesn't freeze. |
| **Technical** | **Video Generation Complexity:** The MoviePy/ffmpeg pipeline for creating narrated videos is more complex or resource-intensive than anticipated, causing significant delays. | Feature may be delayed or have to be simplified, impacting the MVP deliverables. | Medium | Medium | **Mitigation:** Develop a basic proof-of-concept for the video feature in an early sprint. If it proves too complex, the plan is to pivot to a simpler alternative, like generating a slideshow with text-to-speech audio instead of a full video. |
| **Technical** | **Third-Party Dependency Failures:** A critical library (e.g., for PDF parsing, UI components) becomes deprecated, has a breaking change, or contains a security vulnerability. | Development blockage, unexpected bugs, security risks. | Low | Medium | **Mitigation:** Lock dependency versions using packagelock.json. Regularly audit packages for vulnerabilities. Choose well-maintained and popular libraries to minimize the risk of abandonment. |
| **Technical** | **Deployment & Integration Challenges:** Difficulties in configuring the environment on Render (backend) and Vercel (frontend), causing delays in deployment and testing. | Delays in delivering a testable product and completing end-to-end testing sprints. | Medium | Medium | **Avoidance/Mitigation:** Document all environment variables and deployment steps. Perform a simple "hello world" deployment for both frontend and backend in Sprint 1 to identify and resolve configuration issues early. |
| **Project Manage-ment** | **Scope Creep:** The team is tempted to add unplanned features (e.g., advanced quiz generation, gamification) that are out of the initial MVP scope. | Project timeline slips, team gets distracted from core deliverables, MVP is not completed on time. | High | High | **Avoidance:** Strictly adhere to the feature set defined in the MVP. All new feature ideas will be added to the Jira backlog for consideration after the capstone project is delivered. The Project Manager will be responsible for enforcing the scope. |
| **Project Manage-ment** | **Inaccurate Time Estimations:** Tasks, particularly backend AI integration, take longer than the 2-week sprint allows, creating a cascading delay effect. | Sprints fail to deliver their intended functionality, leading to a rushed and incomplete project in the final weeks. | Medium | High | **Mitigation:** Break down large tasks (e.g., "Summarization Pipeline") into smaller, more manageable sub-tasks in Jira. Re-evaluate and adjust the release plan at the end of each sprint based on the team's velocity. |
| **Team/ People** | **Key Person Unavailability:** A team member with specialized knowledge (e.g., backend AI integration) becomes unavailable due to illness or competing academic priorities. | A critical part of the project is stalled, jeopardizing the timeline. | Low | High | **Mitigation:** Promote knowledge sharing through pair programming on complex tasks. Ensure all code is well-documented and pushed to the shared GitHub repository frequently. The Project Manager will track progress to quickly identify bottlenecks. |
| **External/ Data** | **Data Privacy Breach:** A user uploads sensitive or personal information, and there is an unauthorized access to the PostgreSQL database. | Ethical and reputational damage. Failure to meet data privacy policy goals. | Low | High | **Mitigation/Transference:** Store uploaded files with strict access controls in PostgreSQL. The application will not require personal user accounts for the MVP, reducing data storage needs. Add a disclaimer that users should not upload sensitive documents. |
| **External/ Data** | **Copyright Infringement Issues:** Users upload copyrighted textbook chapters, potentially putting the platform at legal risk. | The project could face legal challenges or be required to take down publicly accessible instances. | Medium | Medium | **Avoidance/Mitigation:** For the pilot, the tool will be used in a closed academic context. Implement a "Terms of Use" checkbox during file upload where users must affirm they have the right to use the uploaded material for personal study purposes. |

**3.6 Service Transition**

Upon project completion, the MVP will be deployed, leveraging free-tier hosting services. This ensures accessibility for demonstration and testing purposes without cost.

The team will produce:

* **User Documentation:** Explaining how to upload files, generate summaries, and use the Q&A and video features.
* **Technical Documentation:** Covering architecture, APIs, database schema, and deployment processes.

The platform will initially be rolled out as a **pilot** for university students. Feedback will inform further iterations, ensuring smooth transition from development to limited real-world use.

**3.7 Options Analysis**

Several approaches were considered:

* **Option A (Chosen): Build from Scratch with React + Python + HuggingFace**  
  Provides maximum flexibility, control over architecture, and scalability. Best suited for a capstone project that emphasizes learning and technical growth.
* **Option B: Use Firebase + Third-Party AI APIs**  
  Would allow faster prototyping but limits customization, risks vendor lock-in, and could become costly in the long term.
* **Option C: Build as a Browser Extension**  
  A lightweight solution that overlays study content with AI-generated summaries. While simpler, it would not scale well for advanced features like video explanations and integrated dashboards.

**Option A** was selected as the most viable path to achieve both academic rigor and meaningful outcomes within the project scope.

**4. TECHNICAL FEATURES**

Smart StudyMate will include the following technical features:

* **File Upload with PDF Support:** Students can upload textbooks, notes, or research papers.
* **AI-Powered Summarization and Q&A:** Summaries and interactive Q&A powered by retrieval-augmented generation (RAG).
* **Video Explanation Feature:** Narrated videos generated from summaries to make content more engaging.
* **PostgreSQL Database:** For storing processed text, summaries, and user interaction logs.
* **Secure Backend APIs with Authentication:** Ensuring safe communication between frontend and backend services.

**5. PROJECT ORGANIZATION AND STAFFING**

The team consists of three members with defined roles and shared responsibilities:

|  |  |  |
| --- | --- | --- |
| **Role** | **Member** | **Responsibilities** |
| **Frontend Developer** | Vikas Falke | Development of the React-based dashboard, chatbot UI, video display integration, and user experience design. |
| **Backend Developer** | Kevin Joseph | Core backend development including document parsing, AI model integration, and API endpoint design. |
| **Database Engineer / Project Manager** | Sophiya Baniya | Database management (PostgreSQL setup, schema design, optimization), backend support tasks such as testing and performance tuning. As Project Manager, Sophiya will coordinate sprints, oversee Agile ceremonies (planning, standups, reviews), manage Jira boards, and ensure timely delivery. |
| **Shared** | All Members | Deployment on Render/Vercel, system testing, integration, quality assurance, documentation, and final presentation. |

This structure ensures clear ownership of tasks while encouraging collaboration across roles.

**6. PROJECT BUDGET**

**Personnel Costs:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Role** | **Team Member** | **Total Hours (10 Weeks)** | **Blended Rate** | **Total Cost** |
| Frontend Developer | Vikas Falke | 150 | $40/hr | $6,000 |
| Backend Developer | Kevin Joseph | 150 | $40/hr | $6,000 |
| DB Engineer / PM | Sophiya Baniya | 150 | $45/hr | $6,750 |
| **Subtotal** |  | **450** |  | **$18,750** |

**Infrastructure & Hosting:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Service** | **Purpose** | **Plan Tier** | **Monthly Cost** | **Project Duration in mo** | **Total Cost** |
| Vercel | Frontend Hosting | Pro Plan | $20 | 3 | $60 |
| Render | Backend Hosting | Starter | $10 | 3 | $30 |
| Render | PostgreSQL Database | Starter | $10 | 3 | $30 |
| **Subtotal** |  |  |  |  | **$120** |

**Software & Tooling:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tool** | **Purpose** | **Cost Structure** | **Project Duration (3 Mos)** | **Total Cost** |
| Jira | Project Management | $8.15 / user / mo | 3 Users | $73 |
| GitHub | Version Control | Team Plan ($4 / user / mo) | 3 Users | $36 |
| Figma | UI/UX Design | Professional ($12 / editor / mo) | 1 Editor | $36 |
| **Subtotal** |  |  |  | **$145** |

**Third-Party API Costs:**

|  |  |  |
| --- | --- | --- |
| **Service** | **Purpose** | **Estimated Cost** |
| AI Model APIs | LLM for Q&A and Summarization | $150 (initial usage credit) |
| Video Generation API | Text-to-Video Explanations | $100 (based on minutes generated) |
| **Subtotal** |  | **$250** |

**Grand Total:**

|  |  |  |
| --- | --- | --- |
| **Category** | **Estimated Cost** | **Notes** |
| Personnel Costs | $18,750 | Based on 450 hours of development time. |
| Infrastructure & Hosting | $120 | Commercial-tier cloud services. |
| Software & Tooling | $145 | Paid plans for professional tools. |
| Third-Party API Costs | $250 | For production-grade AI/video services. |
| **Subtotal** | **$19,265** |  |
| Contingency (17%) | $3,279 | Industry standard for managing risk. |
| **Grand Total** | **$22,544** | Total Estimated Commercial Value |