HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

SOFTWARE ENGINERRING

FEASIBILITY REPORT

FLASHCARD WEBSITE

GROUP 17

Name	Student's ID
Ngô Duy An	20235883
Nguyễn Trọng Minh	20235976
Đồng Đại Huy	20235945
Nguyễn Việt Hòa	20235936
Phan Đăng Đạt	20235908

Instructor: Dr Nguyễn Quốc Tuấn

CONTENTS

1. Problem Definition & Objectives

1.1 Problem

In today's fast-paced educational environment, learners are often overwhelmed by the volume of information they need to memorize and retain for exams, professional certifications, or personal knowledge growth. Traditional study methods such as note-taking, rereading, and passive review have been proven to be inefficient for long-term memory retention. Students frequently forget learned materials due to the lack of systematic repetition and active engagement with the content.

While digital tools like Quizlet and Anki offer flashcard-based learning, they often come with limitations such as complex user interfaces, lack of collaboration features, or steep learning curves for new users. Many also fail to integrate emerging technologies like artificial intelligence (AI) to personalize learning experiences or assist users in content creation.

The proposed project aims to solve these issues by developing a user-friendly flashcard website that enhances memory retention using spaced repetition and active recall—two proven learning techniques. The system will allow users to create, organize, and share flashcard decks easily, while integrating AI assistance to help generate questions or answers and track user progress effectively

1.2 Objective

The main objective of this project is to design and develop a **web-based flashcard platform** that simplifies the memorization process while encouraging consistent and active learning.

Specific goals include:

- **Enhance learning efficiency:** Use evidence-based methods like spaced repetition and active recall to improve users' long-term memory retention.
- **Provide a simple and intuitive interface:** Ensure that users of all technical levels can easily create, edit, and review flashcards through a responsive and accessible design.
- **Integrate AI support:** Offer an intelligent assistant to help users generate flashcards, suggest improvements, and provide personalized study feedback.
- **Encourage collaboration and knowledge sharing:** Allow users to share decks publicly, browse decks created by others, and clone them for personal use.
- **Enable performance tracking:** Implement progress analytics that help users monitor their learning journey and identify weak areas.
- **Ensure scalability and maintainability:** Use modern web frameworks and databases to build a platform that can evolve with user needs and future improvements.

1.3 Scope of project

In-Scope Features:

- Simple user interface with responsive design so that users can easily use the website through the computer and mobile phone interface.
- Creating, editing, and organizing flashcards.
- Integrated AI assistant to help generate answers.
- Progress tracking and performance analytics.
- User authentication and cloud synchronization.
- Sharing with other users.
- Browse & Clone Public Decks: Find public decks created by others and create a personal copy to study.
- Basic search/filter for public decks by title or topic.
- Integration of spaced repetition algorithms.

Out-of-Scope Features: The following features are intentionally excluded from this phase of development:

- Advanced or adaptive Spaced Repetition algorithms (e.g., Anki-style SRS tuning).
- Real-time collaborative deck editing.
- Native mobile applications (iOS and Android).
- Gamification features (leaderboards, badges, rewards).
- Multimedia embedding (audio, video, complex image support).

1.4 Target Users

The primary users of this system are **students, educators, and lifelong learners** who seek effective ways to memorize information and reinforce learning.

- **Students (primary target):** High school, college, and university students preparing for exams, quizzes, or standardized tests.
- **Educators and Tutors:** Teachers who want to create shared decks for class use or recommend curated flashcard sets to their students.
- **Self-learners and Professionals:** Individuals preparing for certifications, language learners, or anyone pursuing self-improvement through consistent study habits.

Additionally, **educational institutions** can adopt the platform to support blended or online learning, enabling collaborative study environments. The design prioritizes inclusivity, ensuring accessibility for users across different devices and technical backgrounds.

2. Background Research & Context

2.1 Overview

Learning and memorization are core aspects of education, yet traditional study methods such as reading notes repeatedly or highlighting text often result in short-term recall rather than long-term understanding. Research in cognitive psychology shows that **active recall** (actively retrieving information rather than passively reviewing it) and **spaced repetition** (reviewing information at increasing intervals) significantly improve memory retention.

Flashcard-based learning tools implement these techniques by prompting learners to engage with information repeatedly and actively. This makes them particularly useful for students studying vocabulary, formulas, or factual material that requires memorization.

However, despite the proven effectiveness of this learning model, many existing tools lack simplicity, accessibility, or adaptability for diverse learning contexts.

2.2 Existing Solutions

There are several established flashcard applications in the market, such as **Anki**, **Quizlet**, and **Brainscape**. Each has its strengths, but also limitations that create an opportunity for a more balanced and user-friendly alternative.

Platform	**Strengths**	**Limitations / Gaps**
Anki	Uses an advanced spaced repetition algorithm (SM-2), supports large decks and plugins, strong customization.	Steep learning curve, outdated interface, less intuitive for beginners, limited collaboration/sharing for free users.
Quizlet	Modern UI, easy deck creation and sharing, mobile apps, supports images and audio.	Many features are locked behind a paywall, recent restrictions on AI-generated content, limited personalization in learning algorithms.
Brainscape	Clean design, adaptive repetition intervals, analytics for progress tracking.	Subscription model limits free use, fewer customization options compared to Anki.

From these comparisons, it becomes clear that while existing tools are powerful, they often compromise between **usability, accessibility, and adaptability**. Some prioritize advanced functionality but are too complex for casual users; others focus on user experience but restrict access or personalization through premium models.

2.3 Identified Gaps and Justification for the Proposed Solution

From the review of existing systems and user needs, several **key gaps** were identified that justify the creation of a new flashcard web platform:

1. **Complex user interfaces** deter beginners from effectively using existing tools. → The proposed system prioritizes **simplicity and usability** through a clean, responsive interface accessible across desktop and mobile browsers.
2. **Limited collaboration and sharing features** restrict community-based learning. → The project introduces **public deck sharing and cloning**, enabling users to learn from others and contribute their own study materials.
3. **Lack of AI-driven learning support** in most existing platforms. → Integration of an **AI assistant** will assist users in generating flashcards, summarizing content, and improving study efficiency.
4. **Accessibility barriers and paywalls** reduce equal access to learning tools. → This project ensures a **free, open-access platform**, making it inclusive for students and educators worldwide.
5. **Overly complex spaced repetition systems** are difficult to configure. → The proposed version will use a **simplified review system** based on known/repeat logic, offering the benefits of spaced repetition without technical complexity.

By addressing these issues, the proposed flashcard website offers a **practical, user-centered, and educationally sound solution**. It combines the effectiveness of proven learning methods with the convenience of modern web technologies.

3. Technical Feasibility

3.1 Technology Stack Selection

- **Frontend: Next.js** – A modern React framework that provides built-in routing, server-side rendering, and API route capabilities. It enables fast, scalable, and SEO-friendly web applications, ideal for an interactive learning platform.
- **Backend: Express.js** – A lightweight and flexible Node.js framework used to build RESTful APIs efficiently. It simplifies server-side logic, integrates smoothly with Supabase, and supports future scalability.
- **Database: Supabase (PostgreSQL-based)** – A robust open-source backend-as-a-service providing authentication, database, and storage features. Built on PostgreSQL, it ensures reliable data management, real-time synchronization, and secure access control.
- **Deployment: Vercel & Render** – The frontend (Next.js) is hosted on **Vercel**, enabling automatic builds, previews, and instant deployment from GitHub. The backend (Express.js) is hosted on **Render**, which offers a free, reliable environment for Node.js applications. This combination ensures smooth communication between services and efficient CI/CD integration.

3.2 Evaluation

These technologies are modern, open-source, and widely adopted in the web development community. They offer comprehensive documentation, active community support, and smooth integration with each other. The selected stack provides a balance of simplicity, performance, and scalability — suitable for a 12-week academic project focused on learning and experimentation.

3.3 Team Capability

The development team has foundational knowledge of JavaScript, web frameworks, and database integration. Previous projects provide sufficient familiarity with React and Node.js. The primary challenge lies in coordinating frontend-backend communication and managing asynchronous data with Supabase. However, with structured planning, consistent communication, and version control tools (e.g., GitHub, Trello), the team is fully capable of implementing and completing the project successfully within the scheduled timeframe.

4. Operational Feasibility

The Flashcard Learning Web App project is proven to be feasible in adoption, use, and maintenance in the real-world context:

- **User Readiness:** Students and self-learners are already familiar with online learning tools and mobile study apps. The proposed system builds on familiar concepts such as creating study materials, flipping cards, and tracking progress.
- **Operational Workflow:** The platform operates as a web-based system accessible through any modern browser. Users can register, log in, create flashcard decks (with a front side for questions and a back side for answers), and practice through randomized sessions. The system also allows users to mark learning progress and optionally share decks publicly.
- **Maintenance and Support:** The development team will oversee application updates, API versioning, and bug fixes. Since the hosting platforms (Vercel, Render, and Supabase) manage scaling and uptime automatically, the maintenance workload is minimized. Environment variables and automated CI/CD pipelines ensure consistent deployment across environments. Routine monitoring can be performed using free tools such as **UptimeRobot** or **Sentry** for uptime and error tracking.
- **Usability and Adoption:** The system emphasizes an intuitive interface, fast response time, and minimal distractions. With easy access and built-in progress tracking, users are encouraged to practice regularly and adopt the app as a daily learning tool. Little to no training is required for operation.
- **Sustainability:** By using **free hosting services**, the platform remains cost-effective and sustainable. The combination of **Vercel**, **Render**, and **Supabase** ensures continuous availability without infrastructure costs. Future scalability—such as migrating to paid plans or integrating additional services—can be achieved with minimal reconfiguration. This ensures the project can continue running efficiently in both educational and research environments.

5. Economic Feasibility

5.1 Cost estimation

Resource Type	Description	Estimate
Team Members	5 developers	5 persons
Tools	VS Code, IntelliJ Idea, Figma, GitHub, Supabase CLI	Open source
Hardware	Individual laptops	Self-brought
Hosting & Deployment	Vercel (frontend), Render (backend), Supabase (database)	Free plans

5.2 Benefits

For users:

- Convenience and affordability for students.
- Increased safety and trust through verified users.
- Networking and collaboration among students.
- Accessible across devices with no installation required.

For team members:

- Opportunity to gain hands-on experience in full-stack web development using modern technologies (Next.js, Node.js, and Supabase).
- Provides a portfolio-quality project for student developers.

As the cost of resources is low while the benefits for users and team members are multiple, this project is economically feasible for development and maintenance.

6. Schedule feasibility

Approach: Lightweight Agile with short sprints, continuous integration.

Milestones:

- Milestone 1. Feasibility Study & Analysis Report (End of Week 4)
- Milestone 2. Requirement Analysis (End of Week 7)
- Milestone 3. Design, Implementation & Testing Progress (End of Week 12)
- Milestone 4. Final Project Report & Completion Assessment (End of Week 15)

6.1 Schedule for milestone 1 & 2

Weeks 1–7: Timeline & Tasks:

Week 1:

- **Project kickoff & planning**
  - Define objectives, success criteria, constraints
  - Draft **Problem Statement** & **Goals** (from feasibility)
  - Pick toolchain: GitHub Projects, Issues, Labels, Milestones
- **Feasibility Study – Outline**
  - Skeleton for: Technical, Operational, Economic, Schedule, Legal/Ethical & Risks

Week 2:

- **Background & Context**
  - Survey **Anki, Quizlet**
  - Identify differentiators (deck sharing, AI helper, simple SRS)
- **Features**
  - Define **In-Scope MVP** features:
    - Auth & profile; Deck/Card CRUD; Study session with simple SRS (Known/Repeat)
    - Progress tracking; Browse public decks & **Clone**; Basic search/filter
    - Share public deck link; Responsive UI
  - **Out-of-Scope**: advanced SRS (Anki-like), real-time co-edit, native apps, gamification, complex media
- **Technical Feasibility (Draft)**
  - Architecture overview: Next.js + Express API + Supabase (schema draft)
  - Risks log (SRS complexity, scope creep, testing debt, VCS conflicts)

Week 3:

- **Operational Feasibility**
    - Define primary personas: Student, Teacher
    - High-level workflows: create deck, study, share
- **Economic Feasibility**
    - Rough cost-benefit (domain/hosting/time vs. learning value, future premium)
- **Legal & Ethical**
    - Data privacy (emails, study history), copyright for shared decks, fair use
- **Feasibility report**

Week 4:

- **Finalize Feasibility Report**
- **Lightweight UI research**
    - Paper/Figma sketches for key screens (Home, Deck, Study)

Week 5:

- **Requirements Discovery**
    - Team members learn about requirements analysis documents.

Week 6:

- **User Stories & Acceptance Criteria**
    - Backlog by epic: Authentication, Decks & Cards, Study/SRS, Public Browse & Clone, Search/Filter, Sharing, Progress
- **Non-Functional Requirements**
    - Performance (TTFB, FID), Security (auth/session), Accessibility (WCAG AA), Reliability
- **System Context & API**
    - Context diagram; initial endpoints; Supabase schema v0.1

Week 7:

- **Backlog Grooming & Prioritization for Sprints**
- **Requirements Package** finalized (stories, NFRs, wireframes, schema)
- **Go/No-Go for implementation**

6.2 Schedule for milestone 3 & 4

Sprint Roadmap (2-Week Sprints) for coding application features:

Sprint 1: *Weeks 7–8: Foundations & Core CRUD (MVP Skeleton)*

- **Goals**
  - Working vertical slice: Auth → Create deck/card → List view → Persist
  - CI set, preview deploys, baseline test scaffold
- **Scope / Stories**
  - Auth: email/password sign-up, sign-in, sign-out
  - Decks: create/read/update/delete; list own decks
  - Cards: add/edit/delete cards within a deck
  - Project scaffolds: Next.js app, Express API, Supabase connection
  - DB schema v1: users, decks, cards, study_reviews (skeleton)
- **Tasks**
  - FE: Next.js app shell, routing, form components, responsive layout
  - BE: Express project, REST endpoints for decks/cards, validation
  - DB: Supabase SQL migration v1, row-level security basics, indexes
  - DevEx: GitHub Actions CI (lint/test/build), env secrets, preview deploy (e.g., Vercel + Render)
  - QA: Unit tests for deck/card reducers/services; Postman collection; smoke E2E (Playwright/Cypress)
- **Usability Test**
  - 3–5 users create a deck and 10 cards; measure task success & time
- **Definition of Done**
  - All stories have AC met; tests: unit ≥ 60% for touched modules; E2E smoke green
  - Lint/typecheck clean; accessible forms (labels, focus, keyboard)
  - Deployed **v0.1** to preview with release notes; docs updated (README, API)

Sprint 2: *Weeks 9–10: Study Flow & Simple SRS; Public Browse/Clone*

- **Goals**
  - First usable study session with Known/Repeat logic
  - Public deck browse + Clone to My Decks; basic search/filter
- **Scope / Stories**
  - Study session: flip card, mark Known/Repeat; next-card selection
  - Progress basics: per-deck stats (cards learned, due today)
  - Public area: list public decks, deck details, Clone
  - Search/filter by title/topic (simple text search)

- **Tasks**
    - FE: Study UI (keyboard shortcuts), progress badge, public pages
    - BE: endpoints for public decks, clone, simple ranking
    - DB: study_reviews table finalized (interval, last_seen, status)
    - QA: E2E for study happy path; API contract tests
    - Analytics: basic event logging (study start/finish)
- **Usability Test #1**
    - 5 users complete a 20-card study; collect top 5 UX issues → backlog
- **Definition of done**
    - E2E subject "study" green; progress persists across sessions
    - Search returns relevant decks within 200ms p95 (seeded dataset)
    - **v0.2** deployed with changelog; top-3 UX issues addressed in-sprint

Sprint 3: *Weeks 11–12 (ends M3): Sharing & AI assistance + Hardening*

- **Goals**
    - Share a deck publicly with stable link; AI helper prototype for generating sample Q/A
- **Scope / Stories**
    - Sharing: toggle public/private; copy public link
    - AI assistance: generate suggestions for answers/alternatives (using Gemini's API)
    - Progress dashboard v2: streak, due counts, last 7 days
    - Error handling & observability: basic logs, rate limits
- **Tasks**
    - FE: share controls, dashboard charts, AI prompt UI (opt-in)
    - BE: AI endpoint (stub or provider), quotas; logging & rate-limiter
    - QA: Security passes (authZ on deck access); negative tests
    - Docs: Architecture diagram v2; API doc synced; risk register update
- **Usability Test #2**
    - Focus on sharing & AI prompt clarity; record confusion points
- **Definition of done (Milestone 3)**
    - All new endpoints authenticated & authorized; logs for errors
    - AI helper flagged as **beta**; can be disabled; no PII leakage
    - **v0.3** deployed; **Milestone 3 package**: demo deck, design artifacts, test summary, updated risks
- **Deliverable**: M3 progress bundle + demo

Sprint 4: *Weeks 13–14: Refinement, Performance & Accessibility*

- **Goals**

    - Final-polish the product for the final demo.
    - Improve page speed and usability.
- **Tasks**
    - **Frontend:** run performance & accessibility checks; optimize loading (lazy load, pagination); Completing UI
    - **Backend:** add API pagination and basic caching headers
    - **QA:** run regression suite; test on mobile & desktop; log bugs and verify fixes.
    - **Docs:** update User Guide, README setup, and Changelog; write final test plan.
- **Usability Test #3**
    - Perform a full study session test focusing on usability and accessibility; verify navigation, and record any issues.
- **Definition of Done (DoD)**
    - All planned features are fully implemented, tested, and stable.
    - No open issues or known data-loss bugs remain.
    - Product is complete, meeting all sprint goals and acceptance criteria.
    - The final build is deployed and ready for handover to stakeholders.
    - Documentation and user guide are finalized and up to date.

Week 15: *Finalization & Completion Assessment*

- **Release v1.0** cut & deploy; tag + release notes
- **Final Report** (consolidate feasibility + requirements + design + test/UX findings + risks + lessons learned)
- **Demo day** script & dataset; contingency checklist
- **Handover**: credentials rotation, backlog state, post-project recommendations

Deliverable: Final Report (PDF), Slide deck, v1.0 release notes, Demo script, Handover package.

6.3 Risks

These are risks to the development team and the technology choices.

Internal risks:

Risk 1: *Over-engineering a complex algorithm for the Spaced Repetition System.*

- **Description:** Spaced Repetition System is a memorization technique.
- **Visibility:** Monitor sprint goals; check if SRS work delays MVP features.
- **Fallback:** Use simple “Known/Repeat” system;

Risk 2: *Scope Creep and Feature Bloat*

- **Description:** Constantly adding features → unfinished, buggy product.
- **Visibility:** Project Manager enforces MVP scope; backlog maintained separately.
- **Fallback:** Features freeze 3 weeks before deadline; focus only on core.

Risk 3*: Insufficient Testing Leading to a Buggy User Experience*

- **Description:** With a short timeline, testing often gets cut or delayed. Critical bugs in core flows (like saving a card, user login) could make the application unusable and ruin the demo.
- **Visibility:** Track test coverage per sprint; assign rotating tester role weekly.
- **Fallback:** fix critical bugs before adding new features.

Risk 4: *Version Control Catastrophe*

- **Description:** Git conflicts, overwritten/lost code.
- **Plan:** Establish and enforce a **Git workflow** from day one. All team members must agree to and follow the rules (e.g., main branch is protected, all work is done in branches, pull requests require a review from another teammate).
- **Fallback:** Protected main branch; restores from backups if needed.

External Risks:

Risk 1: *Academic Timeline Pressures and Conflicting Commitments*

- **Description**: Team members have other courses, assignments, and exams. This can lead to missed deadlines, reduced contribution, or burnout during peak academic periods (e.g., midterms).
- **Visibility:** Shared team calendar highlights peak busy periods.
- **Fallback:** Build two-week buffer; reduce sprint scope during exam weeks.

Risk 2: *User Adoption and Feedback*

- **Description**: The team builds the app in isolation. During the final demo, it's revealed that the user interface is confusing, or the sharing feature doesn't work as intended from a user's perspective.
- **Visibility:** Conduct early and continuous user testing.
- **Fallback:** Collect detailed feedback at the Demo, put the problem into Product Backlog. Plan for the next Sprint, instead of writing the code immediately, it is possible to do a prototype to confirm the UI/UX solution before re -development.

6.4 Alternatives

These are the alternative options if the project is not done.

Alternative 1: *Build a Simpler Prototype*

- **Description:** Scale back the scope dramatically to a single-user, local-storage-only.
- **Pros:** Much lower technical risk and development time.
- **Cons:** Fails to meet the core requirement of being "used and share among many users," significantly reducing the project's ambition and learning value.

Alternative 2: *Fork and Modify an Open-Source Project*

- **Description:** Find an existing open-source flashcard app and add the desired features.
- **Pros:** Could be faster than building from scratch.
- **Cons:** Requires understanding a potentially complex foreign codebase, which can be more time-consuming than building a simple. May not fulfill the "build from scratch" requirement of the course.

7. Legal, Ethical & Risk Considerations

- Privacy and data protection:
    - **Risk:** Unlawful/unclear basis for processing; over-collection.
      Mitigations: Privacy Notice; just-in-time consent where needed; strict purpose limitation; configurable data retention; delete inactive accounts after X months; **do not** use private decks for model training by default; obtain explicit opt-in.
    - **Risk:** Data subject rights not met (access, deletion, portability).
      Mitigations: Build **Self-Service Privacy Center** (export, delete, change email); document manual process during MVP.
    - **Risk:** Cross-border data transfers via cloud or AI API.
      Mitigations: Prefer regional hosting; execute DPAs with vendors; restrict logging of personal data; pseudonymize analytics.
- Security:
    - **Risk:** Compromise of accounts (weak passwords, credential stuffing).
      Mitigations: MFA (optional at MVP, recommended later), rate limiting, password strength checks, breached-password blocklists, step-up verification for sensitive actions.
    - **Risk:** Secrets leakage (API keys, database creds) in repo or logs.
      Mitigations: Secrets in environment store; never commit to VCS; short-lived tokens; rotate keys; least-privilege service accounts.
    - **Risk:** Data loss/unavailability (cloud outage, accidental deletion).
      Mitigations: Automated backups (daily), point-in-time restore, tested recovery drills; read-replica or multi-AZ later.
    - **Risk:** Third-party/vulnerability exposure.
      Mitigations: Dependency scanning (SCA), SAST/DAST in CI; patch SLAs; security.txt and disclosure policy; emergency rollback plan.
- Copyright & IP (UGC, Sharing & Cloning):
    - **Risk:** Users upload copyrighted textbooks/exam banks; public cloning propagates infringement.
      Mitigations: ToS prohibiting infringement; **Notice-and-Takedown** workflow; repeat-infringer policy; visible **License Picker** for public decks (CC-BY, CC-BY-SA, CC-BY-NC, or "All Rights Reserved"); provenance/attribution fields; hash-based duplicate hints (advisory only).
    - **Risk:** Ambiguous ownership of AI-generated suggestions.
      Mitigations: Terms clarify user's rights to AI outputs they accept into decks; require users to ensure lawful use.
- AI Ethics & Academic Integrity
    - **Risk:** Hallucinated or biased answers; over-reliance undermines learning.
      Mitigations: Label AI as assistive; show confidence cues and sources when available; allow instructors to **disable AI** for classes; encourage active-recall workflows; provide "verify with source" prompts.

  - **Risk:** Cheating/misuse (exam answers, leaked content).
    Mitigations: Prohibit assessment content uploads where prohibited; educator reporting channel; moderation; rate limits; content filters on obvious exam keywords when flagged by classroom mode.
  - **Risk:** Toxic/unsafe content in public decks.
    Mitigations: Community Guidelines; report/flag system; automated moderation for hate/violence/self-harm; human review queue for trending decks.
- Policies & User-Facing Documents
  - **Terms of Service (ToS):** UGC ownership and license, prohibited content, repeat-infringer policy, AI output responsibility, termination rules.
  - **Privacy Notice:** What we collect, why, lawful bases, retention, third parties, cross-border disclosures, user rights, contact.
  - **Community Guidelines:** Respectful conduct, no harassment, no illegal content, academic integrity.
  - **Moderation & Takedown SOP:** Intake → triage → action → notification → appeal → recordkeeping.
  - **Security Disclosure (security.txt):** How to report vulnerabilities; safe-harbor language.

References

- AlliedUin. (2023). *Anki vs Quizlet vs Brainscape: Which flashcard app is the best?* Medium. Retrieved from https://alliedunin.medium.com/anki-vs-quizlet-vs-brainscape-which-flashcard-app-is-the-best-570f2d3d176f
- FlashcardLab. (2023). *Choosing the best flashcard app: A thorough review of Anki, Quizlet, FlashcardLab, Cram and Brainscape.* Retrieved from https://flashcardlab.co/blog/choosing-the-best-flashcard-app-a-thorough-review-of-anki-quizlet-flashcard
- KidMedNC. (2023). *Unlocking your child's brain power: The magic of active recall and spaced repetition.* Retrieved from https://kidmednc.com/unlocking-your-childs-brain-power-the-magic-of-active-recall-and-spaced-repetition
- Osmosis. (2023). *Active recall: The most effective high-yield learning technique.* Retrieved from https://www.osmosis.org/blog/active-recall-the-most-effective-high-yield-learning-technique