ADA University
School of Information Technologies and Engineering
CSCI 3509: Intro to Software Engineering
Fall 2023

Homework 1: **Project Proposal and Software Development Plan**

Imba Learn

Project team: Imba gang (Team 8)
Instructor: Kamila Ismayilova

Submitted in partial fulfillment of the requirements of the CSCI 3509: Intro to Software Engineering course project

Version date	Version information
23 Oct 2025	**Initial draft**
25 Oct 2025	**The timeline of the project is discussed**
26 Oct 2025	**Final version is prepared**

Team member	Contribution to the homework
Leyla Aliyeva	Worked on Project Deliverables, Assumptions and Constraints, References and Definitions
Diana Kuanyshkyzy	Worked on the Project organization part, planning the development timeline
Dariya Kairkhanova	Worked on the overview, scope and objectives parts
Emin Dabakhov	Worked on the Project organization part. Worked on selecting the appropriate technologies for the web application

Table of Contents

1. Overview

This is a project proposal for partial fulfillment of the requirements of the *Introduction to Software Engineering* course in the School of Information Technologies and Engineering at ADA University, Baku, Azerbaijan.

We propose the development of a web-based educational platform inspired by Quizlet, designed to help students and educators with learning through interactive study tools. The platform will allow users to create different modules on different topics. The platform will enable users to create customizable study modules and flashcards on various topics featuring three primary modes: flashcards, tests and quizzes.
Additionally, we are considering the potential to make a mobile application version of this project to complement the web version. Our goal is to create an intuitive, user-friendly tool that optimizes the learning experience and makes it more interesting and convenient.

2. Project Description

Scope and Objectives

Potential audience:

Our project targets two primary user categories. The first group are the students who will use the website or application as a learning tool to memorize new terms and assess their knowledge independently. The second category is teachers who may use it as a tool for teaching, to make the lessons more interactive and interesting, and help their students to memorize new terms effectively. While both groups share the common objective of leveraging the platform for learning, their motivations differ: students focus on personal learning, whereas teachers aim to enhance their instructional methods.

Aspects/mechanics that should be covered in the system:

It must provide a convenient and pleasant user experience, must be easy to understand and effective to use. Users will be able to use such modes like flashcards, tests, or quizzes based on their needs. Tests and quizzes will provide feedback on the results. The project must have a convenient and pleasant design. Also, the project must enable users to log in and save modules they created and results of the tests and quizzes.

Main objectives of the project:

The main objectives of the project are creating a website that helps to memorize new terms in different modes, and providing users with the opportunity to personalize their modules and use the website as the tool to learn or teach effectively.

Major functionalities and features:

With this software we want to satisfy goals such as providing stipends and teachers with the tool that makes the learning process convenient, effective and interesting. Using flashcards, tests and quizzes will help people to memorize and check their own knowledge and progress. We aim to provide convenient and reliable software to maintain the necessary functionality to help people learn.

List of features:

1. **User account:** Users must be able to create, edit and manage their account. Register and log in, save changes, create personalized modules and save them. Set and change password and username. View personal statistics and progress.(Optional)
2. **Modules:** Create modules, flashcards; edit and save them; delete modules; organize, filter and search for specific modules.
3. **Flashcards Mode**: Flip flashcards, shuffle and make them appear in random order; track not studied and learned flash cards.
4. **Quiz mode:** Automatically generated quizzes based on the flashcards; Provide scores,, time of completion and feedback; different modes of quizzes: multiple choice, true/false, fill-in.
5. **Test mode:** Automatically created practice test that will consist of the majority of terms in the module. The score and the feedback will be provided at the end of the test.
6. **Learning progress (Optional):** Progress of completion of learning the module; history of quizzes and tests; visual progress bar showing the status of each term: not studied, learning, learned.
7. **User interface:** Responsive, convenient design and simple navigation and instructions.

Data collection and Confidentiality

The website will require users to sign up and log in before using it. During the registration, the user must provide their email address to enable authentication, account creation, and password recovery if needed.. Also, their names and usernames to personalize their experience and associate their created modules with their accounts. In addition, we will optionally ask users to fill out their personal information as age, gender, and purpose of using

the website for statistical analysis. All collected data will be securely stored and used in accordance with privacy and data protection policies.

Project Deliverables

Deliverables include:

- **Project Proposal and Software Development Plan** – outlines the project goals, description, architecture, timeline, tools, team roles, and technologies that will be used.
- **Requirements Specification Document (SRS)** – details functional and non-functional requirements, user stories, and use cases. Includes a detailed plan of user interaction with the system, as well as use case diagrams to clearly illustrate how users will interact with the application.
- **System Design Documents / UI/UX Design** – includes web and mobile design mockups created in **Figma**. The designs will follow a style suitable for the project's features, with inspiration from existing flashcard and educational apps such as **Quizlet**, for elements like layout, cards, and navigation. This ensures a user-friendly and familiar interface without designing entirely from scratch.
- **Source Code** – complete, well-documented code for:
    - Web application frontend (React.js + Next.js, Tailwind CSS)
    - Mobile application frontend (Swift + UIKit)
    - Backend API (Node.js + NestJS, PostgreSQL)

  This includes the implementation of all core features such as user authentication, flashcard management, study mode, and API integration.
- **Stretch Goal** – adding optional or additional features beyond the core functionality, such as advanced sorting/filtering of flashcards, Basic Gamification,
- **Deployment Files and Instructions** – Docker configuration and hosting setup for the web application on **Vercel**, with clear instructions for running the system.(Optional) For the mobile app, deployment instructions will include setting up the project in **Xcode**, running it on a simulator or physical device.
- **Final Report / Documentation** – summarizes the project process, architecture, challenges, and outcomes. Provides detailed explanations of the code, features, and the system workflow.
- **Presentation / Demo** – a presentation using slides to demonstrate the project, including its purpose, functionality, features, user interactions, and challenges faced during development.

Assumptions and Constraints

<List here any assumptions on which the project is based and imposed *constraints* such as schedule, budget, resources, software to be reused, customer software to be incorporated, techniques to be employed and product interfaces to other products. Include performance or language/platform issues.>

Assumptions: The app targets high school and college students familiar with digital flashcard tools like Quizlet. Users will have reliable internet access. Team members possess the necessary skills for web and mobile development. The backend will support up to 500 concurrent users initially. UI designs will be inspired by Quizlet's layout, created in Figma, ensuring no proprietary assets are reused.

Schedule and Budget: The project will be completed within a 4-month semester timeline (September–December 2025). No budget is required, and free tools and platforms (Figma free

plan, Vercel hosting, Xcode, VS Code) will be used. It is possible that there might me some changes in the project timeline, but in general all of the work will be completed till the end of the semester.

Resources and Software: Development will use personal computers and GitHub for version control. Reusable software includes Figma templates and Quizlet-inspired UI/UX (no proprietary assets). All APIs, libraries, and frameworks (React.js, Next.js, Node.js/NestJS, Swift/UIKit) are open-source or freely available.

Techniques and Interfaces: REST APIs with JWT authentication will enable secure communication between frontend and backend. The shared backend will use PostgreSQL with a normalized schema for data consistency across web and mobile apps.

Performance and Platform Constraints: The web app will be responsive, supporting the latest versions of Chrome, Firefox, and Safari, with page load times under 2 seconds. The iOS app will support iOS 16+ with core features like flashcards and modules, aiming for load times under 1 second.(Optional)

Where it will be used: The application will be primarily used as an educational study tool for students to create, study, and share learning modules containing flashcards. It will help students learn foreign languages, revise terms and definitions, and prepare for school or university assignments that require memorization and review of study materials.

Hardware required: The only hardware requirement is a device capable of running a modern web browser (e.g., Google Chrome, Safari, or Firefox) and, for mobile access, an iPhone compatible with the mobile app.

Target platforms:

- Web application (main): All features of the system will be available here. Accessible on desktop and mobile browsers.
- Mobile application (optional): iPhone (iOS, built in Swift/UIKit). Will include most of the core features but may not have all functionalities available as on the web version.

3. Project Organization

Development Process

For this project, we will follow an Extreme Programming model. XP is an agile development approach that focuses on flexibility, adaptability based on the users' feedback, and frequent updates. We will use XP because it is highly effective and adaptive, and fits our workflow very well. It also will make our communication easier. XP approach aligns perfectly with our series of assignments and ongoing teacher feedback, as it allows our team to continuously refine our project after each submission. At the beginning we plan to create a basic version of our project - a prototype that includes a core functionality. We will later gradually add more new features, refine the design, and enhance the functionality of the software.

<What is your choice for the team communication process? Any tools?>

We will stay organized and ensure that everything is up to date. We use WhatsApp for quick daily communication, updates, and both formal and informal discussions. We also meet after class and discuss our progress and work done. For online meetings, Google Meet platforms will be used. For managing tasks, we use Notion or ClickUp. For seeing the changes made in software development, we will use GitHub to review and merge code efficiently.

Development Tools

Frontend: React.js and Next.js, as well as Tailwind CSS for styling&UI. React and Next in combination are fast and reliable, very popular, with a lot of ready-to-use libraries.

Mobile Application: The mobile version will be developed for iOS using UIKit, providing a smooth and responsive native experience.

Backend: Node.js and NestJS for server-side logic and API development. PostgreSQL managed through Prisma ORM for efficient data modeling and querying.

Version Control & Collaboration: Git and GitHub for source control, collaboration, and issue tracking.

Development Environment: Visual Studio Code as the primary IDE, along with Node Package Manager (npm) or Yarn for dependency management in the frontend and backend systems.

Testing & Debugging: Jest and Cypress will be used for testing and verifying the functionality of the backend system. Jest will primarily handle unit and integration testing, while Cypress will be used for end-to-end testing of user interactions. (Optional)
Swagger will be used for API documentation and to provide an interactive interface for testing and validating API endpoints during development. (Optional)

Deployment & Hosting: Docker will be used to containerize the application and ensure environment consistency across development and production. The project will be hosted on Vercel, utilizing the pre-generated domain provided by the platform for deployment.

Project Schedule

Project task	Expected date	Expected duration	Key Tasks
Planning & Design	**27.10.2025**	**7**	**Project Proposal & Plan; Initial UI/UX Mockups (Figma)**
Core Development: Backend	**03.11.2025**	**7**	**Backend Foundation: Database setup, User Auth API, Basic Flashcard CRUD APIs**
Core Development: Frontend	**10.11.2025**	**7**	**Frontend Foundation: Set up components, implement Auth, connect to Backend APIs**

Core Feature Polish	**16.11.2025**	**6**	**Implement Study Mode; fix bugs from initial integration; Internal Testing**
Stretch Goal / Feedback	**23.11.2025**	**7**	**Implement stretch features; Conduct user testing with peers**
Finalization	**29.11.2025**	**5**	**Final bug fixes, performance optimization, deployment, and preparation of Final Report & Demo**

Team Profile

Emin Dabakhov - have an experience working with the web applications both in frontend and backend. Worked with technologies such as TypeScript, Next.js, React for the frontend and Node.js, Express.js, NestJS, and Prisma for the backend. Comfortable with developing and integrating APIs, managing project structure, and maintaining clear documentation

Diana Kuanyshkyzy - I have experience in full stack development. I worked on multiple projects that involved knowledge in system design, frontend development on React.Js, Next.js, Tailwind CSS, Typescript, backend development on Node.js, Express.js, databases on PostgreSQL, MongoDB, artificial intelligence using various APIs. Worked as a frontend developer in a local startup.

Leyla Aliyeva - I have experience building the frontend for small mobile apps using Swift and UIKit. I have also worked on simple web applications using HTML and CSS, and completed a project in Java, which strengthened my programming and problem-solving skills. I write well-organized, well-documented code, making it easier for the team to collaborate. Additionally, I am confident in presenting ideas and explaining technical concepts clearly.

Dariya Kairkhanova - I have knowledge about using different programming languages such as C, C++, C#, Java, Python. Completed programming languages, data structures, etc courses while studying at university. Used database tools such as PostgreSQL. Practiced creating designing websites using Tilda, HTML, CSS. Also, I have practiced graphic designing for personal purposes and am comfortable with providing clear documentation.

Stakeholder Identification and Involvement

There is no need for subject matter experts or external contributors for this project. All required technical and domain knowledge is already covered within the team's existing skill set and the team is the only side that is responsible to deliver the project.

4. Budget

No budget required. The website will be hosted via Vercel, which is a free platform that doesn't require any spending.

5. References

Quizlet. *Quizlet – Learning Tools & Flashcards.* https://quizlet.com/gb
Vercel: https://vercel.com/

6. Definitions

Term	Definition
Flashcard	**A digital card containing information on both sides — one side typically holds a question, term, or prompt, and the other side contains the answer or definition. Used as a learning tool for memorization and self-testing.**
Module	**A collection of related flashcards grouped under a specific topic or subject (e.g., "Biology Terms" or "Spanish Vocabulary"). Each module can be created, edited, and shared by users.**
User Account	**A digital identity that allows a person to log in to the system, access their saved modules, and personalize their learning experience. Each user account stores data such as name, email, and progress.**
Authentication	**The process of verifying a user's identity when logging in, usually through credentials like an email and password. Ensures that only authorized users can access their personal data.**
SaaS (Software as a Service)	**A software distribution model in which applications are hosted by a service provider and made available to users over the internet, rather than installed on local computers.**
Web Application	**A software application that runs in a web browser and can be accessed on any device with an internet connection. It doesn't require installation like traditional desktop software.**
Mobile Application (App)	**A version of the software designed to run on mobile devices, such as smartphones and tablets. In this case, it refers to the iOS app developed for iPhone users.**
Database	**A structured collection of data that stores user information, flashcards, and modules. It allows the system to efficiently retrieve, modify, and manage data.**
Cloud Hosting	**A hosting method that uses remote servers connected to the internet to store and manage data and applications. It provides scalability, reliability, and accessibility from anywhere.**

API (Application Programming Interface)	**A set of rules and endpoints that allows different software systems (e.g., web and mobile apps) to communicate with the backend server and retrieve or update data.**
Backend	**The part of the application that runs on the server and handles data processing, authentication, and business logic. It communicates with the database and provides information to the frontend.**
Frontend	**The part of the application that users interact with directly — the visual interface built using web technologies like HTML, CSS, and JavaScript.**
Vercel	**A cloud platform used for hosting web applications. It supports modern frameworks like Next.js and provides tools for easy deployment, scaling, and performance optimization.**
Swift	**Swift is a programming language developed by Apple for building iOS applications.**
UIKit	**UIKit is a framework that provides the tools and components for creating the graphical user interface of iOS apps.** **Adding**
Stretch Goal	**Ambitious, secondary target you aim for after the main project goals are successfully completed.**