Web App | Task Specification

Overview

EDUCATIONAL WEB APP TO SUPPORT STUDENT RETENTION OF KEY INFORMATION
The goal is to create a flashcard application where each signed-in user can create flashcards and be tested on them by the app to maximize memory retention and minimize study time.

Key functionality

1. Creation of study flashcards with front and back sides
2. Ability to edit or delete already created cards
3. Interactive test mode for learning the cards

Web Pages

1. Introductory page with sign-up and sign-in forms
2. Study dashboard page with stats and navigation
3. Creator mode page with cards in the database and ability to create/edit/delete them
4. Test mode page with the ability to test the user from their study flashcards

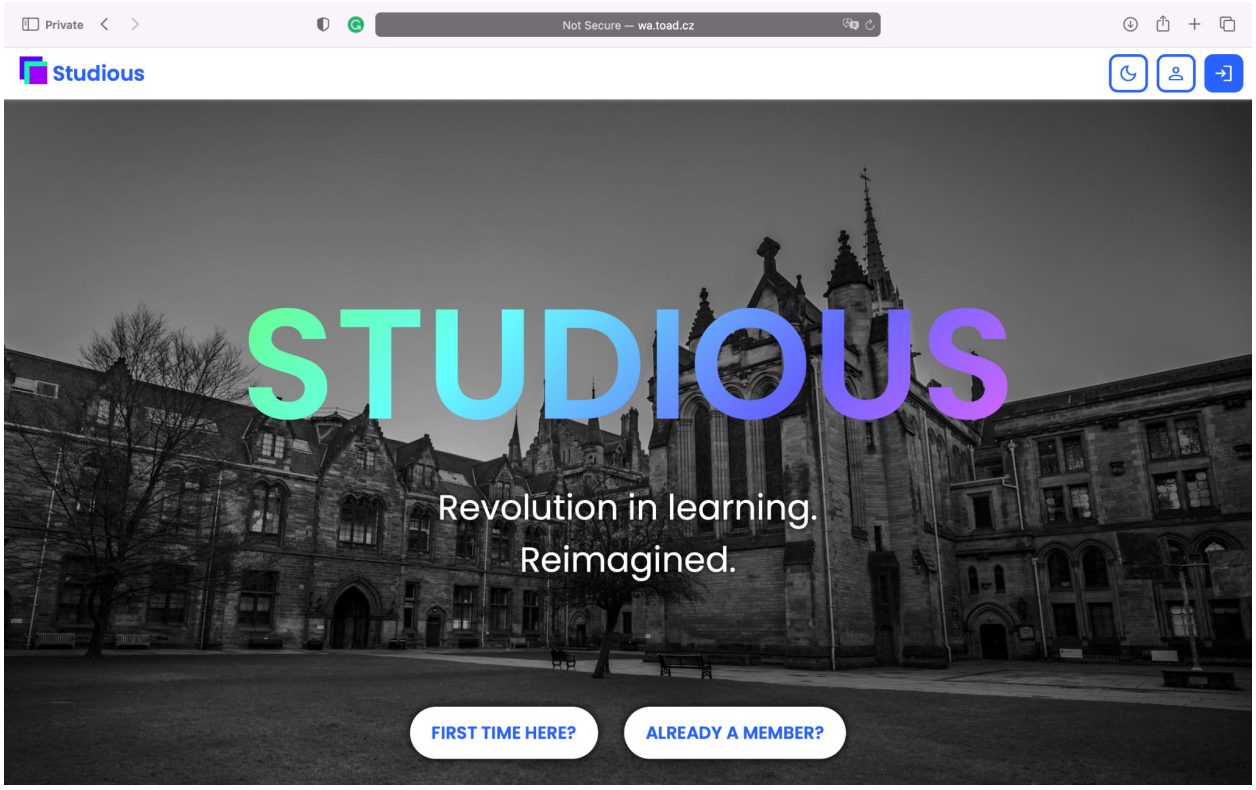
User Roles

1. Not signed-in user - can access only the introductory page to sign-in/sign-up
2. Signed-in user - can benefit from the full functionality of the app (create, read, update, delete operations on flashcards, view stats, and use study mode)

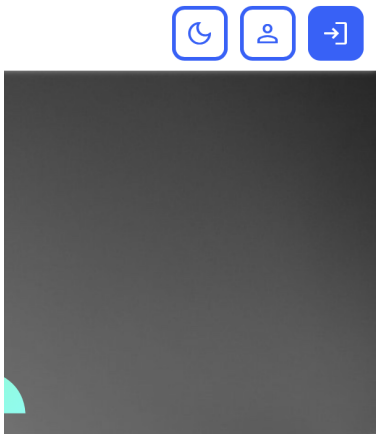
Studious | User Manual

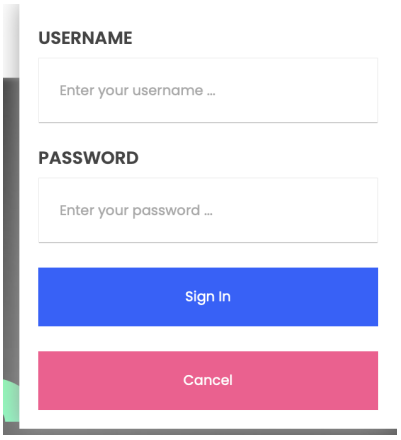
Studious is a web app to help you learn and remember efficiently.
As a user, you can create flashcards and subsequently study, edit, and delete them. Studious chooses an efficient review date whenever you rate your memory response for a particular card.

STEP 0: Welcome on studious

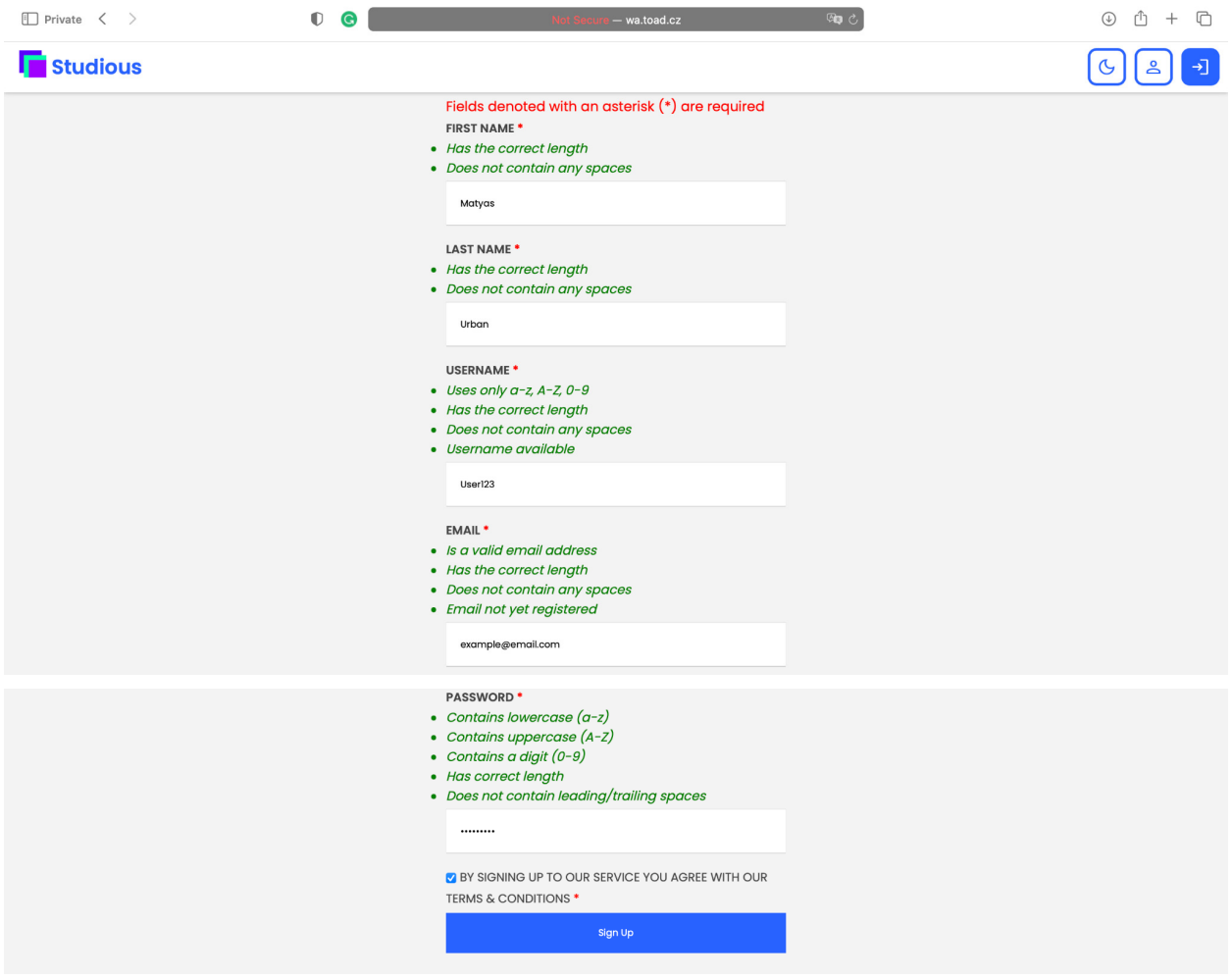


Above, you can see the landing page of Studious. If you are new here, click the “FIRST TIME HERE” button to sign up. Otherwise, choose the “ALREADY A MEMBER” button or the square blue button in the upper-right corner to reveal the sign-in window.



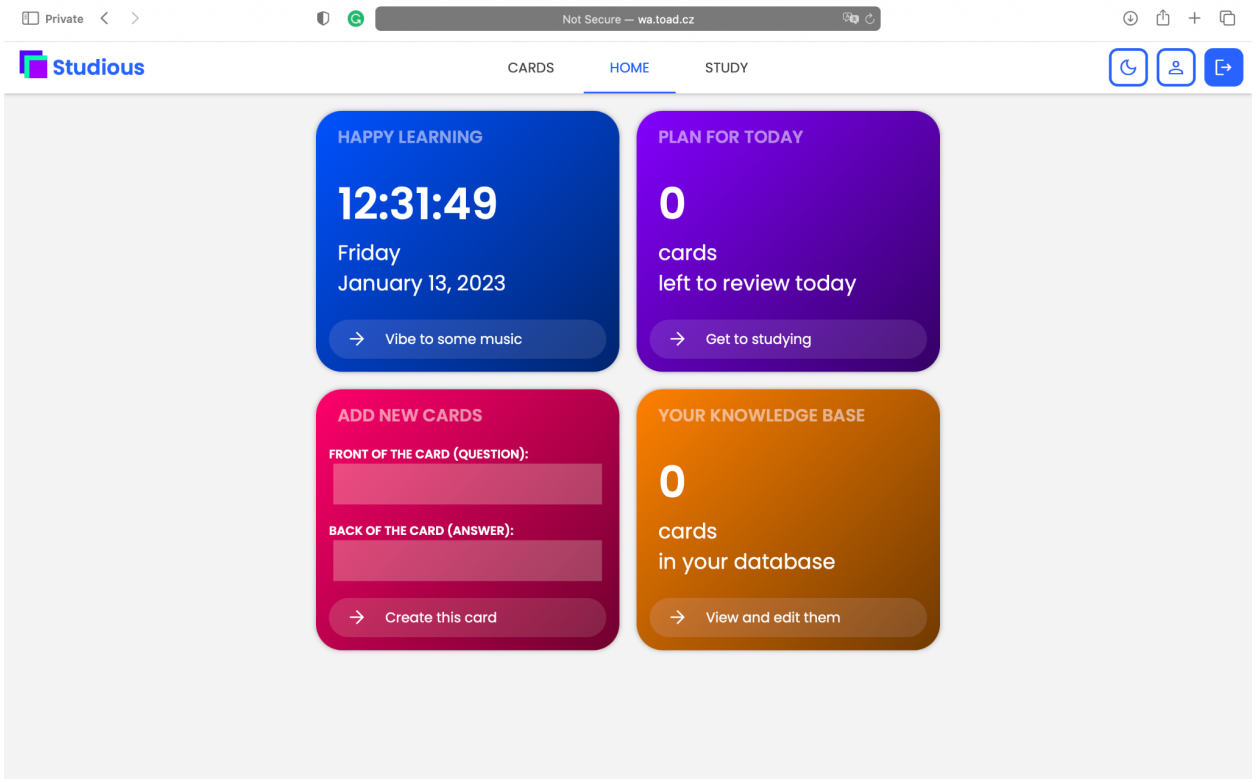


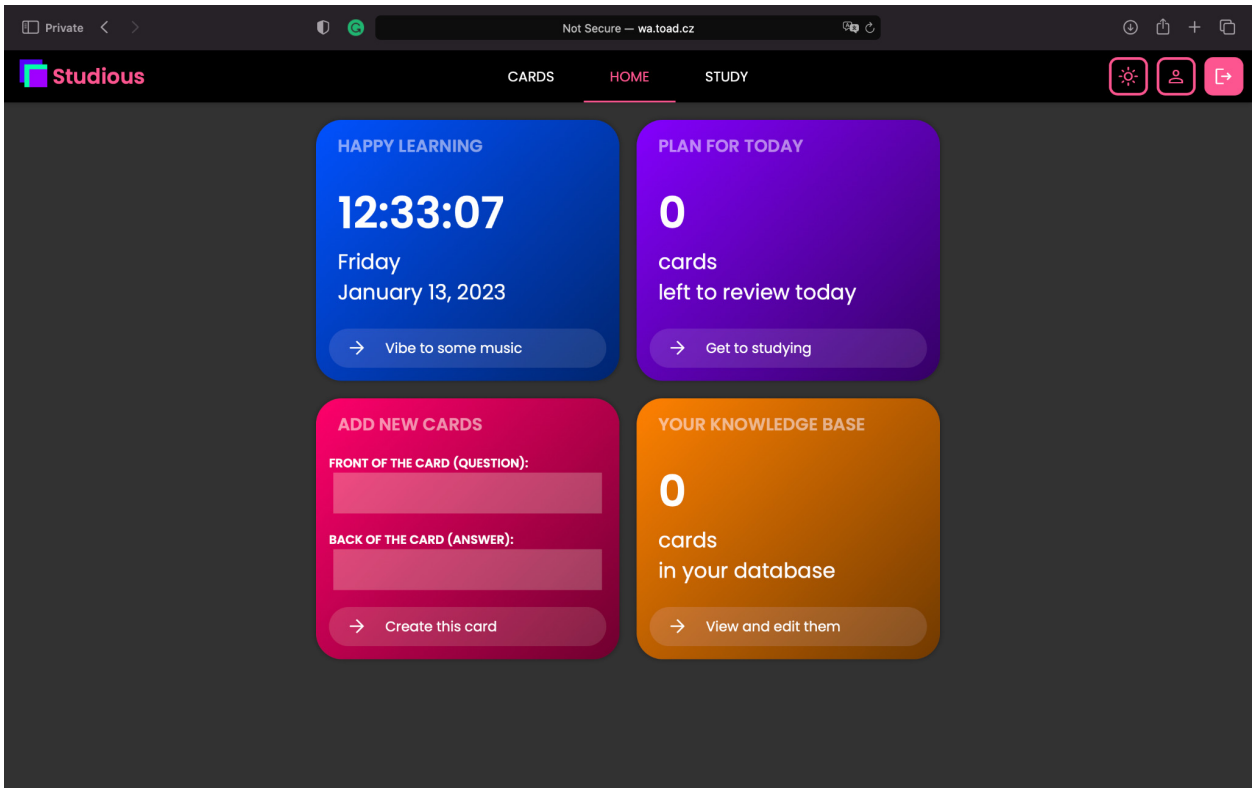
STEP 1: Create an account



Fill in the registration form at the bottom of the page. All fields are required. Pay attention to requirements for individual text fields.

STEP 3: Welcome home





Let's explore the options…
In the upper center, you have the navigation bar:

1) CARDS - A place to browse, add, edit, delete cards that are currently in your database
2) HOME - Dashboard with information and preferred form to create new flashcards
3) STUDY - You'll see there cards that are scheduled for review and rate your performance

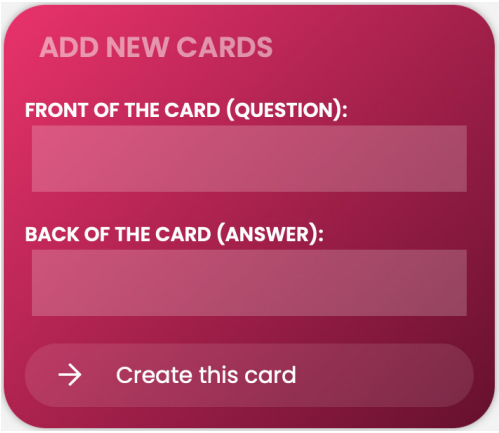
There are also some buttons in the upper-right corner:

1) Switch between light and dark mode
2) Display the username (to make sure you are signed in to the right account)
3) Sign out of your account

There are four colored cards with various buttons:

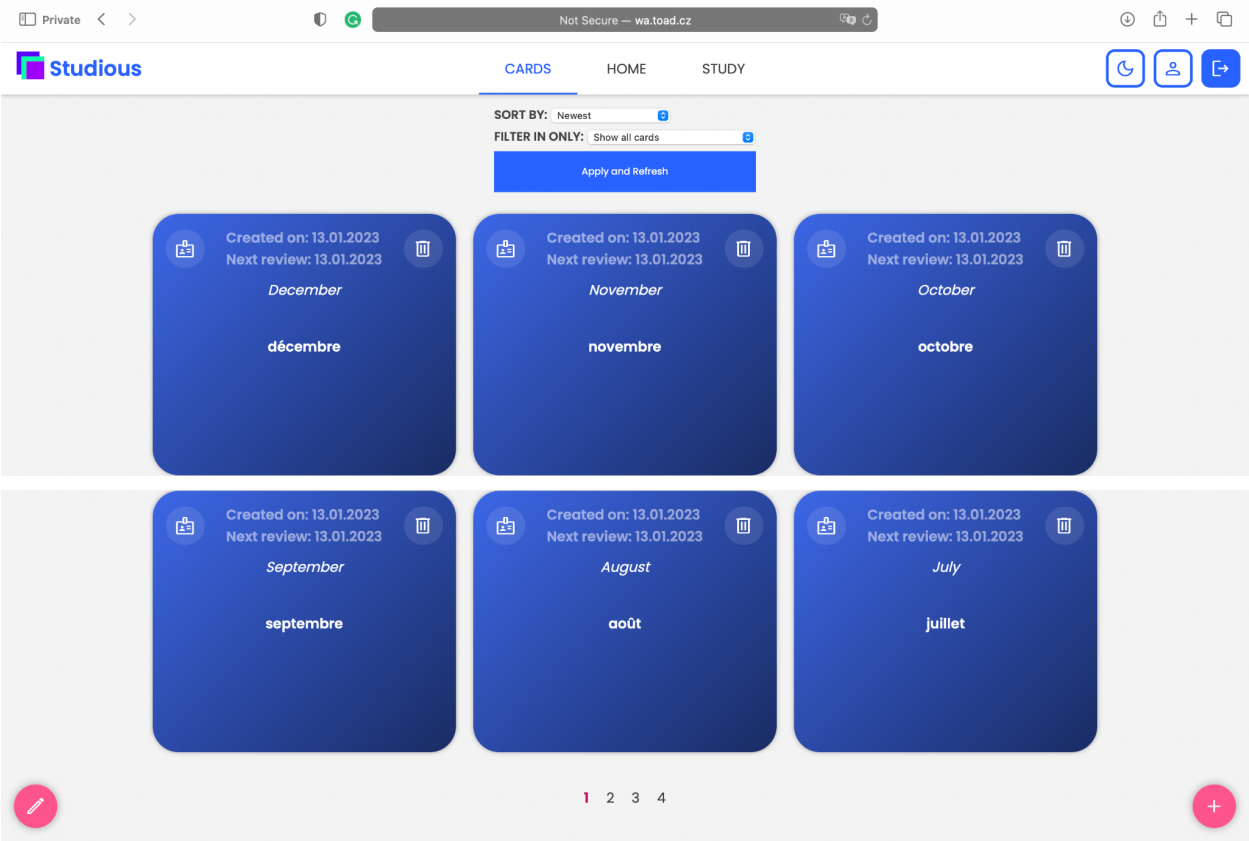
1) VIBE TO SOME MUSIC will open a lo-fi youtube radio to listen to
2) GET TO STUDYING will navigate you to the STUDY page
3) VIEW AND EDIT THEM will navigate you to the CARDS page

STEP 3: Add some cards



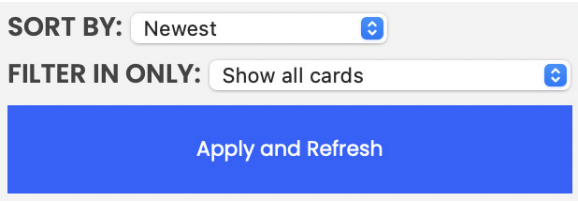
The fastest and most convenient way to create your flashcards is this form. **Note that both fields are mandatory.**
FRONT means the thing you are tested on; the BACK means an answer or a hint. When you fill in both fields, hit enter or click CREATE THIS CARD.
We will create 19 cards to learn the days of the week and months in French. The feedback for your cards being correctly added to your database is that you see the number of cards in your knowledge base continuously increasing.

STEP 4: Browse your cards

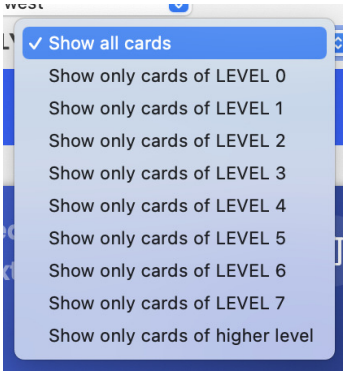


On the CARDS page, you can browse your cards. Let's go through all the options.
There will always be six cards displayed on the screen. To view further cards, select the desired page here: 

In order to add even more cards, click the plus  button and insert a new card the same way you did on the HOME page.



You can even change the order in which the cards are displayed using the order SORT BY option. And to display just some of the cards, change FILTER IN ONLY. You can filter cards by level. What is that?

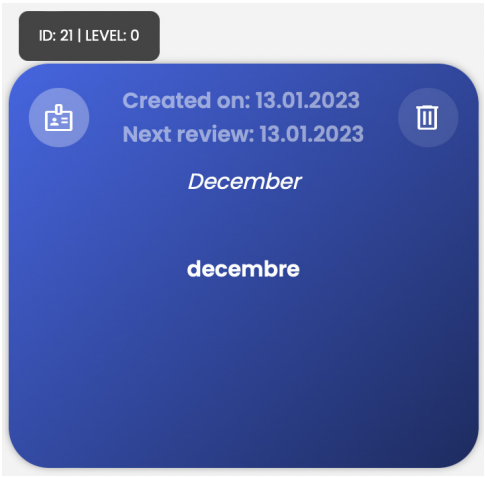


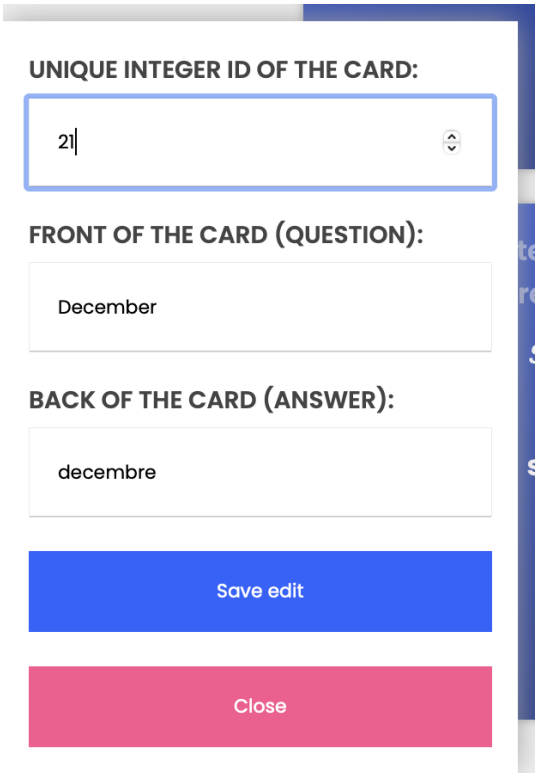
Each time you get a card right in on the STUDY page, the level of your card increases, and so does the interval until the next review. For example, if your card is at level 5. It means it is scheduled for review 5*5=25 days after the last review. However, if you get a card wrong, the level will be decreased to 0, meaning you review that card again the same day (to make sure you learn it), and it will be increased to level 1 (1*1=1 day), meaning you will see that card the next day.

STEP 5: Update a card

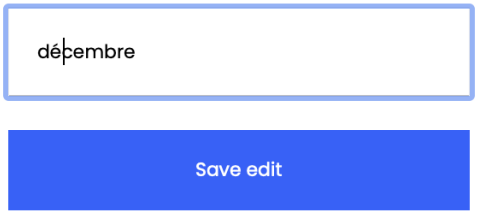
If you have a typo in your card, do the following steps to update your card:

1. Check the unique CARD ID number by hovering over the verification badge 
2. Click the pencil button 
3. The edit form is now displayed. Enter the card ID, and if the ID is valid, the front and back will be already filled in for you.





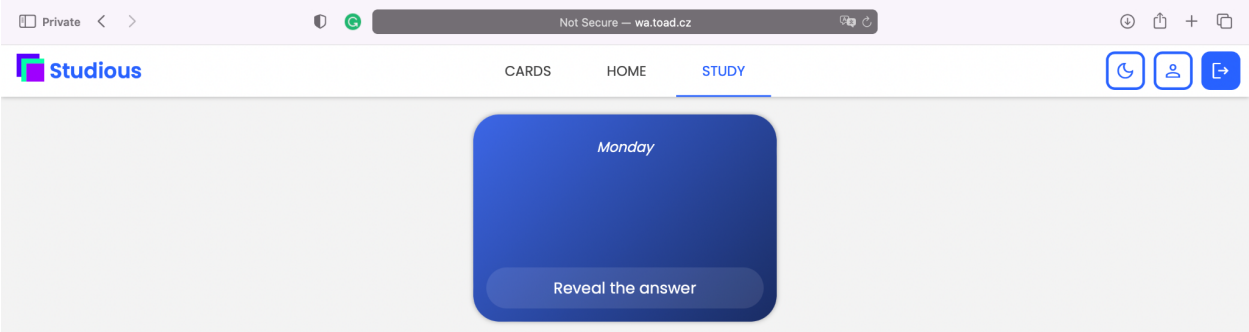
4. Now, all that is left to do is to make the update. In this case, correct the typo, which should use the french accent é, and click SAVE EDIT.



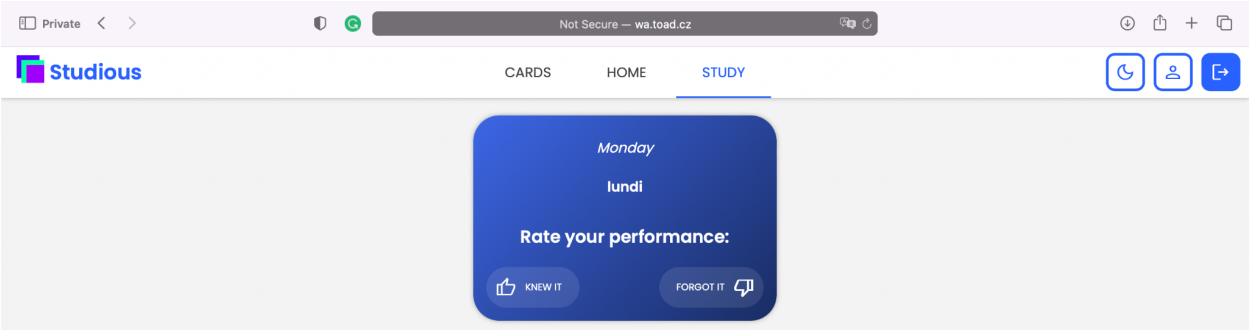
STEP 6: Delete a card

Click the bin button  on the CARDS page to delete the selected card.

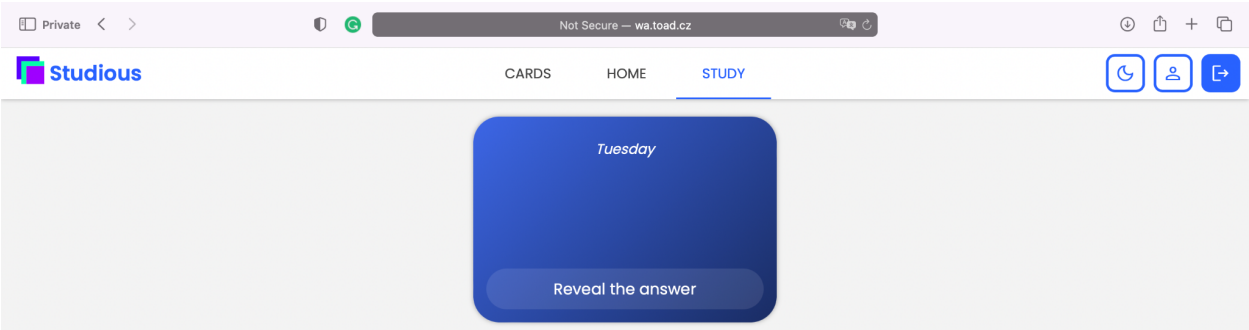
STEP 7: Study your cards



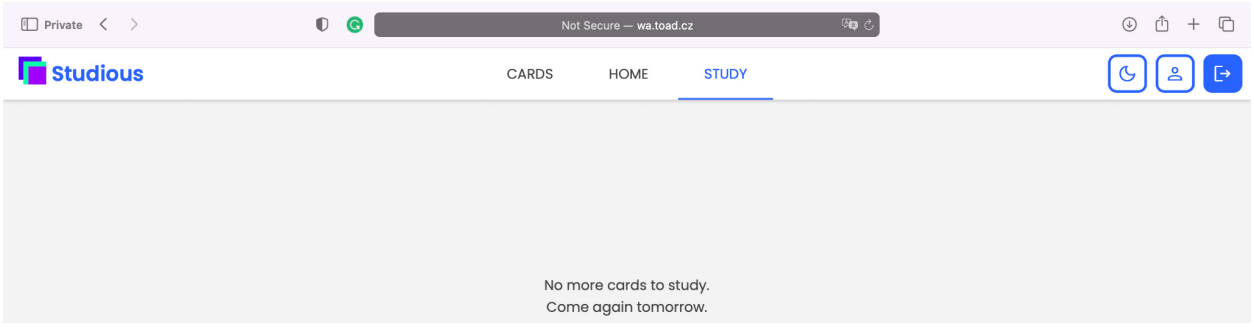
On the STUDY page, one of the cards scheduled to review for today will be displayed. For now, you can only see the front of the card, testing how to say Monday in French. If you think you know the answer or give up, click REVEAL THE ANSWER.



You can see the correct answer displayed, and now is the time to rate your memory response. Click KNEW IT if you were close enough. This will increase the card's level and schedule it for review further away in the future. Another card scheduled for review will be displayed.

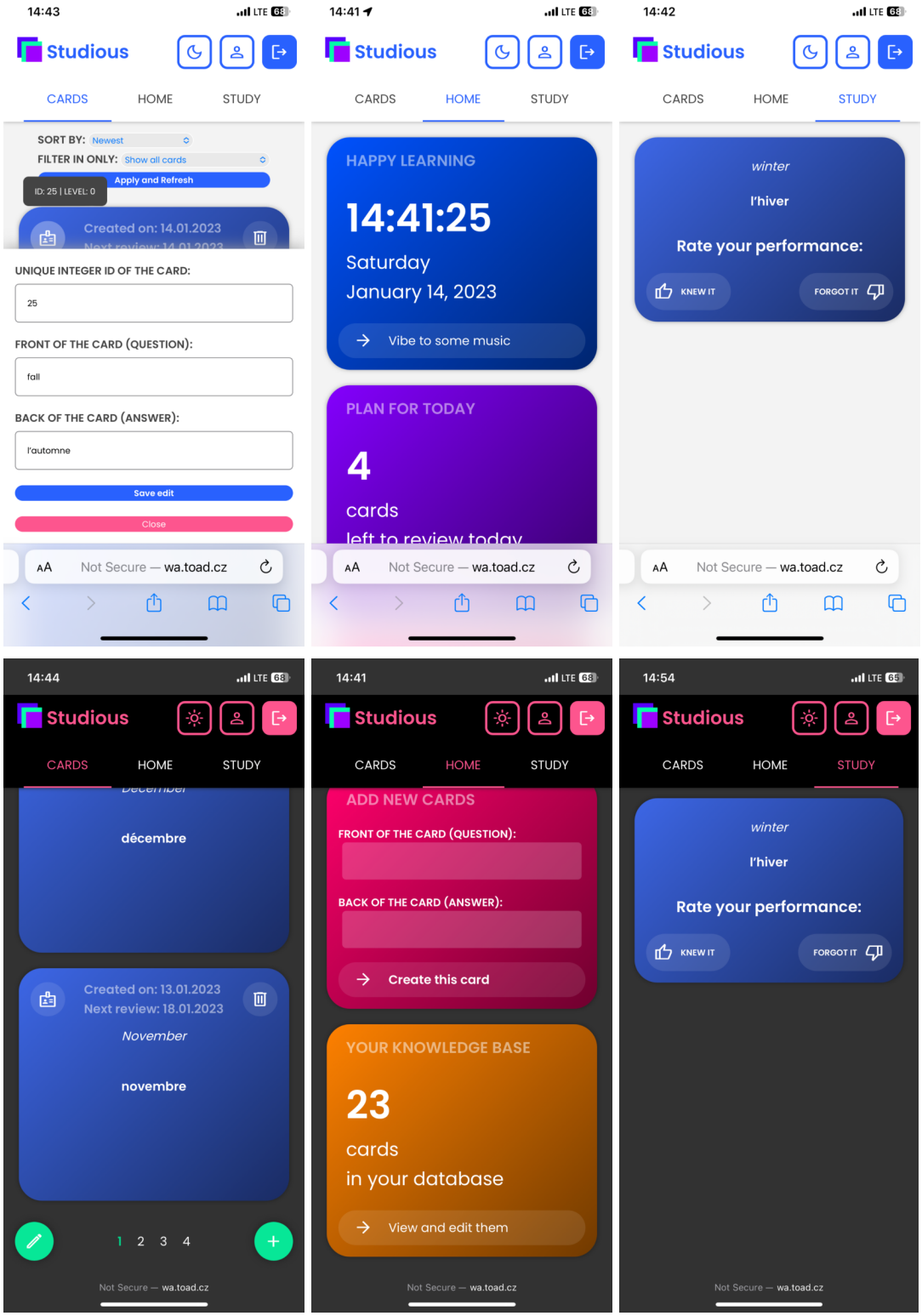


If you click FORGOT IT, the card's level will be back at 0 (default for new cards) and displayed again until you get it right. You will keep getting new cards until there are no left for today.



STEP 8: Go mobile

Studious is optimized for mobile so that you can achieve your goals wherever you go.



Studious | Technical Implementation

The web application is written purely using:

- HTML5
- SCSS3
- JavaScript (ES7+)
- PHP7
- MySQL

No frameworks or special libraries are used in accordance with course requirements.
This choice of stack is given by server-supported tech and course requirements.

Files Overview

Key components of the website that hold HTML code are:

1. index.php
2. home.php
3. cards.php
4. study.php

And their corresponding JS scripts:

1. index-script.js
2. home-script.js
3. cards-script.js
4. study-script.js

Additionally, all HTML-holding files use script.js.
Website structure is styled using style.css plus by appropriate JS scripts.
Server-side logic is handled by the following files:

1. config.php - database access credentials
2. tasks.php - DB connection functions, majority of functions handling POST
3. availability-check.php - script to handle AJAX for username and email availability
4. card-for-edit.php - script to handle AJAX XMLHttpRequest for prefilling ediCardForm
5. delete.php - script to delete a specific card from a database

Additional files include:

1. intro-background.jpg - [0,689 MB] used for index background
2. logo-icon.png - [0.002 MB] used for logo and favicon
3. database-logs.txt - if database connection fails, logs are to be found here

Images are compressed and used resourcefully to optimize website performance.

User Input

- The app entails several complex forms, and all user input is thoroughly validated through HTML, JavaScript, and PHP.
- HTML validation: input attributes (required, type, maxlength, pattern, …)
- JavaScript validation: form.addEventListener('submit',validate), preventDefault, heavy input analysis through regular expressions and other measures
- PHP validation: csrf token, isset, another round of thorough analysis
- Forms are accessible with the use of label for all input elements.

Website Structure with HTML

- All HTML code has been validated to conform to modern standards and avoid deprecated expressions.
- The HTML effectively uses semantic elements like *header, nav,* and *main*.
- The HTML describes purely structure, and no inline CSS is present to improve code readability. The same goes for JavaScript.
- Google Font Poppins is used throughout the document.

Document Presentation with CSS

- All CSS rules are placed in a dedicated file fully separated from HTML code and linked appropriately using <link>.
- Flexbox is being heavily used to achieve the desired presentation of elements.
- Some rules use advanced selectors as well as CSS combinators and pseudo-classes.
- Media queries are present to achieve a compact layout on mobile devices and select only relevant information for print.
- Advanced CSS, including animations, keyframes, and variables is present in order to enhance user experience.

Client-Side Scripting with JavaScript

- JavaScript works with CSS variables as well as LocalStorage to provide users with the option to choose dark/light mode of the website.
- JavaScript is also used to display/hide some of the forms by modifying applicable CSS
- Script for each page uses window.onload function to hookup event listeners (click, submit, keyup) to relevant objects on the page.
- The application makes use of asynchronous JavaScript XMLHttpRequest to contact the server and retrieve relevant information on the go as a user types some input.
- Website uses JavaScript mainly to facilitate visual styles, input validation and provide helpful feedback.

Server-Side Scripting with PHP/MySQL, Security

- Application offers its functionality only to registered users and all others are redirected to an introductory page with sign-up upon an attempt to access secure part of the app.
- Another security measures include:
  - XSS prevention with the use of htmlspecialchars function
  - CSRF prevention with the use of generating a complex token for secure actions
  - Data breach prevention by storing hashed/salted passwords (password_hash())
  - Double submission prevention by effectively rerouting the user
  - SQL Injection prevention with the use of bind_param
- Most errors are effectively displayed to the user in applicable situations while sensitive database-related errors are passed on to a secure file on the server.
- User-input is not lost in case of an error, so that user can correct themself without refilling all the data.
- All database communication is facilitated by MySQL with the use of a secure and effective pipeline $MySQLi->prepare->bind_parram->execute->get_result
- MySQL also powers pagination, sorting, and filtering thanks to LIMIT, ORDER BY, and WHERE keywords

List of tasks.php functions:

connect	Function to establish a connection with our MySQL database
registerUser	Function to sign up a user with thorough validation and prevention of double-submission
loginUser	Function to sign in the user and initiate a session in case of correct credentials
logoutUser	Function to sign out a user in case of the correct csrf token
getNumberOfCards	Function to get number of all cards the user has in their database, used for pagination calculation and HOME page statistic
getNumberOfTodayCards	Function to get number of cards ready for review
getTodayOne	Function to retrieve the next card available for review
feedbackCard	Function to update card level and review date based on the strength of user's memory response
addCard	Function to add a card into signed-in user's personal table
editCard	Function to edit a card in a signed-in user's table
createQueryForTotal	Function to create a MySQL query based on currently used sorting and filtering settings

createQueryFor Page	Function to create a query for current page in pagination
getCards	Function to retrieve cards based on sorting and filtering settings
getCardsForPage	Function to retrieve cards based on sorting and filtering settings just for the current page