

**Learn Something Every Day**

Requirements

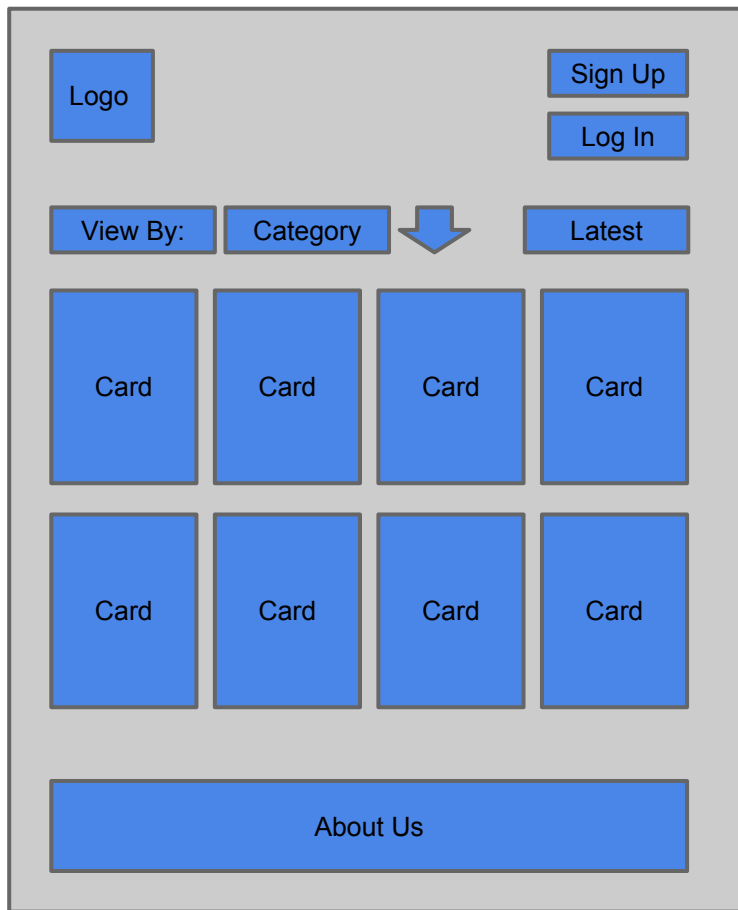
# What is it?

Learn Something Every Day is a product that will make school more fun.

Too many students (elementary to tertiary) spend too much time learning things they are not interested in. Hence, when they come home from school, parents ask “What did you learn?” and the answer is “Nothing” or “Not much”.

We need to put the success back into learning. Students should learn something every day. It doesn't matter if it's simple or hard, the success is important.

We will assume the users range from elementary kids to adults. With this in mind, the processes need to be easy enough for a 6 yo without an email address to achieve.



## Landing Page

The Landing Page is what you see if you hit [www.lsed.com](http://www.lsed.com) (or whatever the URL is) and you have not logged in. It is a generic home page for everyone that is not the same as the individual home page for each user.

Selina will complete the design.

The Category button needs to show a list of categories to choose from. (That's what the arrow will have when the design is finished). Once selected, the most popular cards for that category need to be shown.

The Latest button shows the latest new cards, not the cards with the latest edits.

The number of cards to show needs to be defined by an administrator. The default selection for the cards needs to be defined by an administrator.

The About Us section will link to a page where the team can be show-cased.

The wireframe shows a landing page layout with a central sign-up modal. The background page has a 'Logo' button in the top left, a 'Sign Up' button in the top right, a 'View By' dropdown, three 'Card' placeholders, and an 'About Us' button at the bottom. The modal is a pink rectangle in the center with the following text and input fields:

Enter a user name: \_\_\_\_\_  
Enter a password: \_\_\_\_\_  
Re-enter password: \_\_\_\_\_  
  
Your school: <select> or \_\_\_\_\_  
Your age now in years \_\_\_\_\_  
Your state you live in now : \_\_\_\_\_  
Your country you live in now: \_\_\_\_\_  
  
*Email:* \_\_\_\_\_  
*Re-enter Email:* \_\_\_\_\_  
  
**OK | Cancel**  
  
**Success! Welcome to LSED!**

## Sign Up Page

The user clicks Sign Up.

The Sign Up functionality somehow loads on the Landing Page, either over the top of the existing page, or by clearing a space.

They are asked to enter a user name. This can contain spaces. Rational: a school student could enter their name with a space because they don't have an email address. If the user name is not unique, the user needs to be prompted for an alternative one. That check needs to happen within the flow.

They are asked to enter a password, and re-enter it. These need to be checked against each other. There are no minimum password rules, except not blank. The password needs to be encrypted in some way when stored.

They are asked to enter their school. This acts as information we can use if they need to identify them to, for example, reset their account. They can pick from a list or enter the name. The name should not need to be case sensitive. The list comes from here: <http://www.ed.gov/developers>  
Grady, you can download the data and upload it into the database. Or you can use their API to query the data.

Continued -->

The wireframe shows a page layout with a central registration form. The background is light gray. On the left, there is a vertical column of blue rectangular buttons labeled 'Logo', 'View By', 'Card', 'Card', and 'About Us'. On the right, there is a red 'Sign Up' button. The central registration form is a light red rectangle with the following text and input fields:

Enter a user name: \_\_\_\_\_  
Enter a password: \_\_\_\_\_  
Re-enter password: \_\_\_\_\_  
Your school: <select> or \_\_\_\_\_  
Your age now in years \_\_\_\_\_  
Your state you live in now : \_\_\_\_\_  
Your country you live in now: \_\_\_\_\_  
*Email:* \_\_\_\_\_  
*Re-enter Email:* \_\_\_\_\_  
**OK | Cancel**  
**Success! Welcome to LSED!**

## Sign Up Page 2

They enter their age now, as a single value representing years. We don't need to know their date of birth.

They enter their state, the one they are in now. That should come from a list, or can be filled in, assuming the student is not in the USA.

They enter the country they live in now. That should come from a list or be filled in.

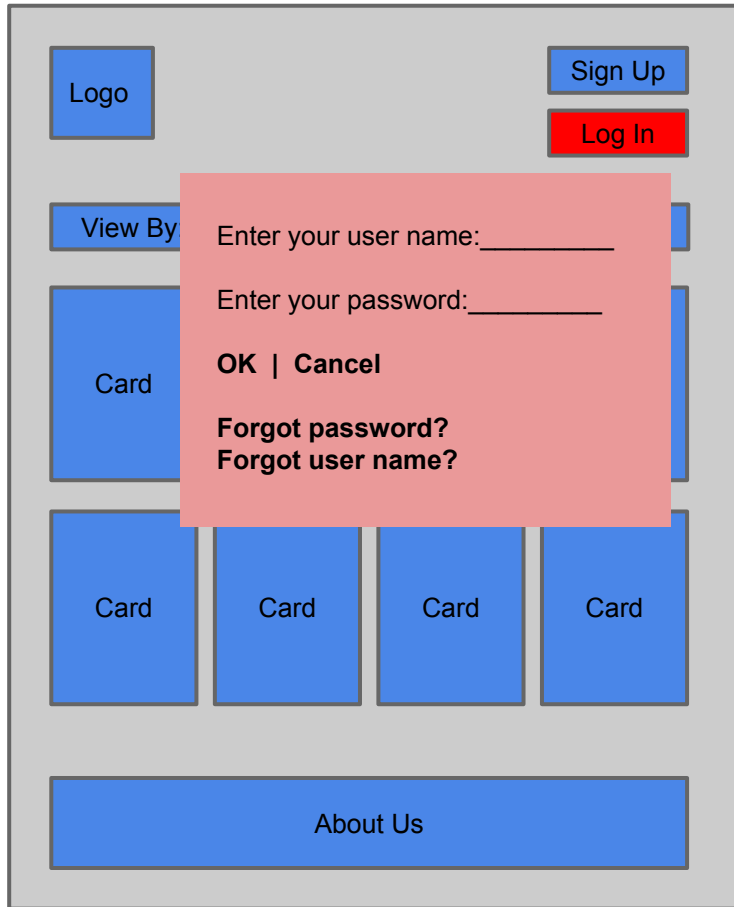
They enter their email address and confirm it, but this is not mandatory. I've shown it in italics, but the design could change that.

When they click OK, that is the signal to do the database transactions as all the other checks should have happened on the page.

We need to store the date of enrollment so we can use it for:

- 1) calculating their birth year so we can make content available appropriately
- 2) in case of any dispute about account.

We need to display a Success or Failure message on the screen so they know they are registered.



## Login Page

The user clicks on Log In. A pop up is shown.

They enter their user name and password, and if they match a user in the database, they are taken to the Home Page for that user (discussed later).

They can try as many times as they like to guess their password, there is no limit to the number of tries.

If they click on Forgot password, or Forgot user name they are shown the questions they need to answer.

Continued -->

The image shows a wireframe of a login page. The background is a light gray rectangle. In the top-left corner is a blue box labeled "Logo". In the top-right corner are two buttons: a blue "Sign Up" button and a red "Log In" button. Below the "Logo" is a blue box labeled "View By". Below that is a blue box labeled "Card". Below that is another blue box labeled "Card". At the bottom is a wide blue box labeled "About Us". A red modal form is centered over the page. It contains the following text and input fields: "User name: \_\_\_\_\_", "Your name: \_\_\_\_\_", "Your email: \_\_\_\_\_", "Your school: <select> or \_\_\_\_\_", "Your age that you entered: \_\_\_\_\_", "Your state you entered : \_\_\_\_\_", "Your country you entered: \_\_\_\_\_", followed by the word "OK" in bold. Below "OK" are three more input fields: "User Name: \_\_\_\_\_", "Enter a password: \_\_\_\_\_", and "Re-enter password: \_\_\_\_\_".

## Login Page - Forgot details

The user clicks on Log In. A pop up is shown.

They enter their user name and password, and if they match a user in the database, they are taken to the Home Page for that user (discussed later).

They can try as many times as they like to guess their password, there is no limit to the number of tries.

If they click on Forgot password, or Forgot user name they are shown the questions they need to answer.

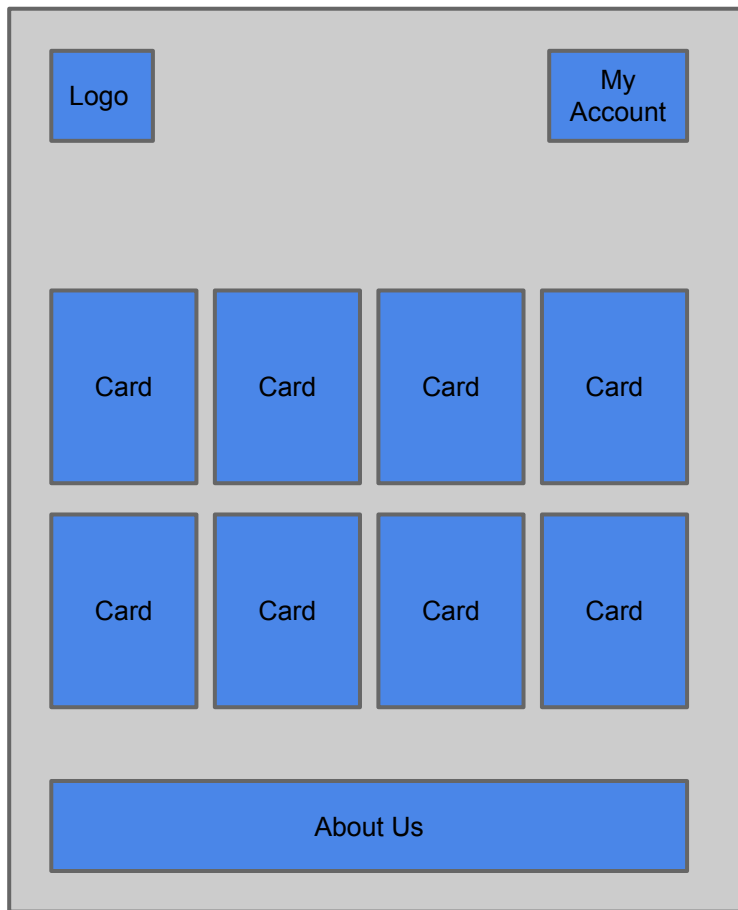
They enter as much information as they have and click OK.

If they have forgotten their password, and if the new password matches in both fields, change their password and log them in. They forgot their password if they entered their User name.

If they forgot their User name, read their “real” name and see if it matches. If so, check the other information matches as well.

If the “real” name doesn’t match, see if there is an email address and check that.

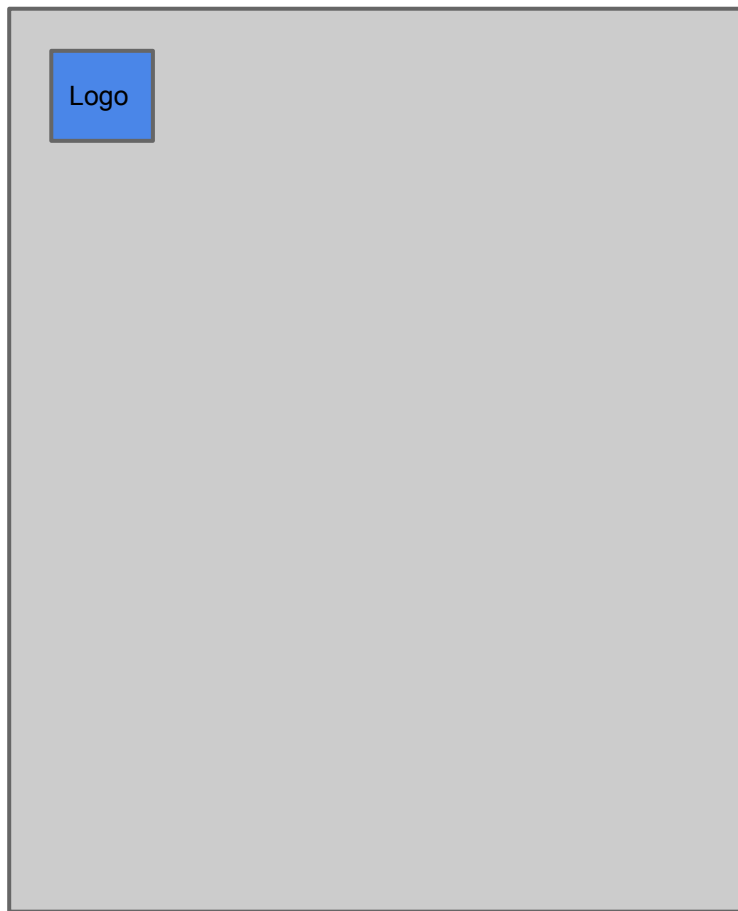
If nothing matches, take them back to the Sign Up screen and get them to start again.



Home Page

Not yet finished





## Administrator Functions

Delete an account

Add bulk accounts

Change content on pages, like About Us

Reset a password

Reset password questions

**Not yet finished**