**User requirements**

Site should be a single page contained within a single file (including JS and CSS). Page should be able to detect it’s own cookies and login user automatically. Cookies should be removed automatically when browser is closed (can be achieved by setting cookie expiration date to the past).

All input fields should accept only letters from English alphabet, numbers 0-9 as well as these symbols: ~!@#$%^\*\_-+=

Input field shouldn’t allow empty spaces unless specified otherwise.

Page should be able to display when not logged in:

**Login**

Login page should contain two fields: login and password. There is should be only one button: login. There shouldn’t be option to register since this portal is for students and staff only.

Page should be able to display **ONLY** when logged in:

It is recommended to have horizontal navigation bar on top of the page with these options: profile, search, users (if admin).

**Profile**

Profile should display login, username and role. If it user’s own profile, it should be indicated. Page should give ability to:

1. Edit username and password if it’s user’s own user profile.
2. Edit username and password, delete profile of anybody except other admins if the user is admin.
3. Edit username and password and role of anybody if the user is superadmin.

**Search**

It’s own “page” that displays list of articles. It should have

1. Searchbar where keywords are entered

2. Sidebar (preferably on the left) with the list of filters, each option is a toggle.

Filters separated on groups and just like search results are loaded dynamically from server. Filters do not affect search results until user hits “apply filters” button. Sidebar can be hidden if user wants to, giving more space for search results.

3. List of search results. Request (and display) all results if no filters selected. It is recommended to not display results all at once, but to display limited amount of results at a time, with ability to switch between portions of the list. In this case it should display number of the “page” and how much “pages” left. Switching between “pages” should not reload an actual page to not affect search results or filters.

4. Each item in the list should display name, category, type and first 15-30 words of description (define suitable amount yourself). If user is admin or superadmin, button to delete article (with confirmation) chould be present. Name should be clickable and leading to a “page” of full article.

5. A button to create article (if tutor, admin or superadmin)

**Article**

Category (expect text up to 30 symbols) (always present)

Type (expect text up to 30 symbols) (always present)

Name (expect text up to 30 symbols) (always present)

Born (expect text up to 30 symbols) (if present)

Died (expect text up to 30 symbols) (if present)

Nationality (expect text up to 30 symbols) (if present)

Known for (expect text up to 100 symbols) (if present)

Notable work (expect text up to 100 symbols) (if present)

About (expect text up to 10000 symbols) (always present)

Year (expect text up to 100 symbols) (if present)

Medium (expect text up to 100 symbols) (if present)

Dimensions (expect text up to 100 symbols) (if present)

Location (expect text up to 100 symbols) (if present)

Designed by developer (expect text up to 100 symbols) (if present)

a button to edit each option and submit changes (if tutor, admin or superadmin), and a button to delete article with confirmation (if admin or superadmin)

**Users** (if admin or superadmin)

Similar to Search, but instead results is a list of users, and filter contains as options only roles (student, tutor, admin, superadmin) and types of search (login or username). Each item should display same fields as profile page, rules of editing and deleting profiles same as in a profile page. “Login” should be clickable and lead to a profile of the said user.