

Frontend Tasks

So... Here we are. Further work is focused on a single project and divided into 5 steps (milestones).

General Requirements

The project should be implemented as a single-page application (SPA) with a client-side routing (via browser's History API). Communication with a remote data source has to be performed via AJAX.

Interface of the web-site must be properly displayed at least on following screens: 1440x900 (desktop) and your mobile device of choice. (You can use Chrome's emulator to test your site).

Styling solution for this project is **plain CSS**. You can also use a css-preprocessor of your choice: Sass, Stylus. **Third-party styling libraries (such as Bootstrap) are not allowed.**

A showcased version of your application (starting from the milestone 3) must be deployed and hosted via remote web-server. You can use one of free hosting services across The Web (ex. Firebase Hosting).

Framework usage

Even though modern front-end development implies the framework's usage, there is no point in learning any of them before you have sufficient knowledge of the basics. That's why front end part of this project has to be implemented in **JavaScript** programming language (all latest language features are allowed and welcomed, TypeScript may be used as a wrapper over JS). **External libraries and frameworks are not allowed by default.**

However, if you still have a desire to learn a particular JavaScript framework or create a portfolio project using a certain library, **this can be discussed individually along with project topic selection.** Personal knowledge check will be adjusted accordingly.

Development Milestones

Milestone 1. Topic Selection

| Yet another startup idea

A small but important milestone as you will work on chosen topic for the rest of the semester. In the attached document you'll find some ready-made ideas with a detailed description of some of them. However, feel free to explore provided topics and come up with your own idea of web-application of similar complexity. But remember, **you can't change the topic after you've passed the second stage.**

Acceptance Criteria:

When you're ready, fill out **the form** to let me know which topic you've chosen.

Milestone 2. UX design & HTML

| Oleg, where's the mock-up?..

To pass the milestone, you must provide images (hand-drawn, digitally created) for most of the pages your web-site will have (for both display sizes). Each block and element should be marked by corresponding HTML-tag.

Example: UX wireframe example (your final result has to be more detailed, than the provided image)

Acceptance Criteria:

- GitHub repository with me invited as a collaborator
- README file with the following information:
 - Full name and group number
 - Project topic and its succinct description
 - Main functions your application will have
 - Wireframe attached in some way (as a link, photo, etc.)

Approximate grading:

[4-6] generic tags knowledge (`div` , `span` , `img` , `input` , `ul` , `p` , etc.)

[7-8] semantically-correct markup (`header` , `footer` , `article` , `nav` , etc.), forms

[9-10] rich semantic (`figure` , `picture` , `aside` , etc.), various input types

Milestone 3. UI design & CSS

The result of this step should be a semi-final set of pages with a production-ready design **for both screen sizes** (the design can be updated on later stages). Pay attention to modern CSS best-practices: float block positioning, !important overrides or <table> layouts are not allowed.

Remember, starting from this point the latest version of you application should be deployed somewhere.

Disclaimer: as we do not study UI design in this course, there are no strict requirements to your project interface appearance. However, a more or less meaningful UI is required to pass this milestone, since it will enforce you to use a good amount of CSS. You will find some helpful links in the attached document.

Acceptance Criteria:

- A set of static pages styled with CSS
- PR into the main branch (don't forget to assign me as a reviewer)
- PR contain a link to the hosted application (the same for further milestones)
- UI doesn't make me cry 😊

Approximate grading:

[4-6] basic CSS (customised inputs, buttons, etc.), `:hover` and `:disabled` states for interactive elements

[7-8] flex box, grid, non-static element positioning, `:focus` state for interactive elements, BEM-like CSS class naming, media-queries, pseudo-classes

[9-10] custom fonts, dark theme support, animations, rich visual style

Milestone 4. Here comes JavaScript

It's time to make your static pages alive and interactive with the help of JavaScript. At this point you must provide a working skeleton of the website: basic navigation, routing, authentication, API integration, etc.

Acceptance Criteria:

- PR into the main branch
- Base principles of clean architecture should be taken into account (please don't place all the functions in a single file)
- As a user, I can log into your application and explore its main features (and it won't break suddenly)

Approximate grading:

[4-6] basic JS: functions, variables (`var` usage is forbidden), DOM manipulations, timeouts and intervals, array methods (like `map`, `reduce`, `forEach`, `filter`, etc.)

[7-8] prototypes/classes, AJAX requests (via Fetch API), promises, arrow functions, JS modules

[9-10] SPA implementation (via browser's History API), `async/await`, local storage, drag&drop technique

Milestone 5. Final Frontier

The final web-site that implements all requirements within selected topic. You should implement missing form validations, errors handling, some nice-to-have features, etc. (we will discuss this in the previous step). CI/CD set-up is also encouraged at this stage.

Acceptance Criteria:

- PR into the main branch
- As a user, I expect all the features mentioned in the README to be implemented, all buttons are clickable, the app doesn't crash even if I submit an empty form, etc.