

# Web Laboratory works

## 1. Simple landing page

**Description:** Create a landing page following the given design (Figma).

**Requirements:**

- Use of semantic tags (<header>, <nav>, <ul>, <footer>, etc.) wherever is necessary
- All fonts, colour palette, elements sizes must strictly **match the design**.
- **Pixel perfect** is also **not required**, BUT your website should visually match the design template (i.e. If the element is horizontally centred on the design, it should look centred on the website)
- All **class names** must not be meaningless (i.e. class="myClass") and should follow the same **naming convention**, preferably BEM, but you could come up with your own (just be consistent)
- **Basic cross-browser support** (*website should look correctly on last versions of modern browsers (Chrome, Safari, Edge)*)
- For now, **responsiveness** is not required.

**P.S** If your semester project has a landing website that **satisfies the requirements**, then you could skip this work by bringing your website for an assessment.

Note: The above goes only for those students. who were **actually responsible for the website development**. The whole team cannot use the right to skip this work!

A website example (from Live coding):

<https://github.com/NazarGorokhivskiy/LandingForLiveCoding>

**LIVE CODING LINK (YouTube video)**

<https://youtu.be/-X-XTeDaCmo>

Variant	Design template link
---------	----------------------

1	<a href="https://www.figma.com/file/CKE7rSZvwWAckhsFpQLfZW/LabWork1-V1">https://www.figma.com/file/CKE7rSZvwWAckhsFpQLfZW/LabWork1-V1</a>
2	<a href="https://www.figma.com/file/XHYmp9yKTxcOQD5zFABKQH/LabWork1-V2">https://www.figma.com/file/XHYmp9yKTxcOQD5zFABKQH/LabWork1-V2</a>
3	<a href="https://www.figma.com/file/vwiwu6GnDphlI25vncvI3F/LabWork1-V3">https://www.figma.com/file/vwiwu6GnDphlI25vncvI3F/LabWork1-V3</a>
4	<a href="https://www.figma.com/file/pRfvOqsPgu7HrIMPgLKkto/LabWork1-V4">https://www.figma.com/file/pRfvOqsPgu7HrIMPgLKkto/LabWork1-V4</a>
5	<a href="https://www.figma.com/file/bW1PzYIXgABwctUobkRAyA/LabWork1-V5">https://www.figma.com/file/bW1PzYIXgABwctUobkRAyA/LabWork1-V5</a>
6	<a href="https://www.figma.com/file/wmMTQTdBymNHxX8ew0mG45/LabWork1-V6">https://www.figma.com/file/wmMTQTdBymNHxX8ew0mG45/LabWork1-V6</a>
7	<a href="https://www.figma.com/file/iJD99BqtY8vzmZdrXbZOCc/LabWork1-V7">https://www.figma.com/file/iJD99BqtY8vzmZdrXbZOCc/LabWork1-V7</a>
8	<a href="https://www.figma.com/file/fJKhikm2WvdbmetEvibfhN/LabWork1-V8">https://www.figma.com/file/fJKhikm2WvdbmetEvibfhN/LabWork1-V8</a>
9	<a href="https://www.figma.com/file/XT2jhcwmHBGxAjIBJsMnH0/LabWork1-V9">https://www.figma.com/file/XT2jhcwmHBGxAjIBJsMnH0/LabWork1-V9</a>
10	<a href="https://www.figma.com/file/ZM6o4UFoZUq9Kup8NjV94p/LabWork1-V10">https://www.figma.com/file/ZM6o4UFoZUq9Kup8NjV94p/LabWork1-V10</a>

## 2. Advanced landing

**Description:** Improve a landing page following the given design (same as previous) by adding full responsiveness and some animation.

### Requirements:

- The website should be **partly responsive**:  
Required: for **320px** (iPhone 5s) **2560px** (4K screens).  
*Hint: for extra-large screen width, you can just fit website content inside a previously set container for the regular screen width and centre it horizontally.*
- The header should collapse to a **hamburger** on small screens
- The website must contain **animations (at least 3)**. Can be made in any possible way, preferably with css properties (*animation/transition*).
- Your project should be **logically structured** (all your CSS can't be in just one-two files)
- All of the **previous work requirements** must be kept.

**P.S** If your semester project has a landing website that **satisfies the requirements**, then you could skip this work by bringing your website for an assessment.  
Note: The above goes only for those students, who were **actually responsible for the website development**. The whole team cannot use the right to skip this work!

A website example (from Live Coding):

<https://github.com/NazarGorokhivskiy/LandingForLiveCoding>

### **LIVE CODING LINK (YouTube video)**

<https://youtu.be/-xBxSv51TAw>

## **CRUD Javascript App (3-5)**

### **3. View Page**

**Description:** In this work, you have to make a simple presentation part of a website - **View Page** (You can find the template of the page by following the [link](#))

For your blocks you must use **data** from your java/python project class.

In case you don't have a **data** from previous year, you should choose any free task from this link:

[https://docs.google.com/document/d/1RW9PpalOIHn-nVli8kbCr71vu\\_XLNsjj4TKpTdBNZ-w/edit?usp=sharing](https://docs.google.com/document/d/1RW9PpalOIHn-nVli8kbCr71vu_XLNsjj4TKpTdBNZ-w/edit?usp=sharing)

Then using JavaScript, you need to implement the following operations on your data (*it is up to you to decide which field should be used for each of the operations*):

- **Sort** of your items option
- **Search** option

- **Count total amount** of some of the field  
(e.g total price of all cars)

### Requirements:

- **Responsiveness** absolutely not required.
- **Styling** is not important at all. Is up to you.

### Our recommendations and tips:

- use JS Array methods: map(), sort(), filter(), reduce()
- use native JS for any DOM operations (querySelector | findById | insertAdjacentHTML | etc)
- a website [example](#) from live coding
- Working with DOM & JS Array methods project  
<https://github.com/bradtraversy/vanillawebprojects/tree/master/dom-array-methods>
- useful projects [examples](#)

## 4. Create/Edit Pages

**Description:** In this work, you need to continue working and add two new parts to your website - **Create & Edit Pages** (You can find the template of these pages by following the [link](#))

**Update/Delete operations are not required for this work!**

Also you must validate the forms using HTML **attributes** (inputs must be configured for your data format).

If incorrect data is entered in the inputs, you must use JavaScript to inform the user with **modal windows** or just plain alert() function.

**Bonus** points are provided for this work, if you implement a styled modal window that will work with JavaScript.

**Requirements:**

- **Responsiveness** absolutely not required.
- **Styling** is not important at all. Is up to you.

**Our recommendations and tips:**

- a website [example](#) from live coding

## 5. Backend

**Description:** In the last part of working on the website you have to implement all Create/Read/Update/Delete operations which must be made via the corresponding HTTP methods in your REST API.

**Important.** You don't have to make a backend from the beginning - connect an existing one that you worked on in the first year

**Our recommendations and tips:**

- [Working with REST API with fetch\(\) link](#) (POST method example)
- a website [example](#) from live coding

## 6-11. Single Page App with React.js

**6th work** - Landing (Home) page with React.js

**7th work** - Items list (Catalog) page. React Routing

**8th work** - Item page. Adding interactivity to a website.

**9th work** - Making everything work. (connecting to REST API)

**10th work** - Cart page (Items list). First look to Redux

**11th work** - Cart page (Shipping form) with validation (? Formik)

**Description:** Create a SPA E-commerce platform with the following pages: Home, Catalog, Item, Cart, using React.js library and your REST API.

*Note:* You probably have to extend your existing backend app with a bunch of new features, so choose backend tech stack *wisely!*

**Requirements:**

- All of the pages have to be made with React.js and all interaction should work properly (without many visible issues)
- Crossbrowserness isn't required, but at least latest versions of Chrome should be supported (so we can check your works on our systems)
- **UX:** All page should stick to the following wireframes. Design and layout of some elements is totally up to you, but if the element is presented on the wireframe, you should implement it as well  
<https://wireframepro.mockflow.com/view/lviv-iot-react-app>
- **UI (Design, css techniques etc.):**  
Is **totally** up to you ;) But choose at least one.

A design example from Live Coding, only for inspiration:)

<https://www.figma.com/file/JYE1Y7MD3dxmRU9HgS1mxv/e-commerce-ui-kit-explorer-s-hop>

A website example (from Live Coding):

**TODO...**