CONTENTS

INTRODUCTION ***2***

DESCRIPTION OF PROJECT AND TASKS ***2***

Login screen 2

Project selection screen 2

Project screen 2

API **5**

Authentication 5

Getting a list of projects 6

Getting a project 6

Create a new project 6

Retrieving Items 6

Project update 7

INSTRUCTIONS FOR THE COMPETITOR ***7***

MARKING SCHEME ***8***

INTRODUCTION

You need to develop a "Skill construction" client web application for site development.

The customer wants the application to be simple and not overloaded, so that it is convenient and pleasant to use, so try to add animation and micro-animation to your design.

Also, the customer asks to pay special attention to the simplicity of working with the application - everything should be clear even for a new user who has never used such editors.

DESCRIPTION OF PROJECT AND TASKS

Login screen

On this screen, you need to display a login form containing a login and pin code input field. After entering a successful pair of login and pin code, the user should see the project selection screen.

Project selection screen

On the screen with a choice of projects, you need to display all available projects based on data from the API. Projects only need to display the title.

Clicking on a project should display the project screen. The transition to the project screen should be animated.

On this screen, you need to place a button to exit the application, when pressed, the user must leave the application and see the login screen. The logged-in user should not see the projects screen or the project screen in any way.

Also, on this screen there should be a button for creating a new project. Clicking on which should open a screen with the created project.

Project screen

The project screen should have the structure as shown in the layout provided (Figure 1).

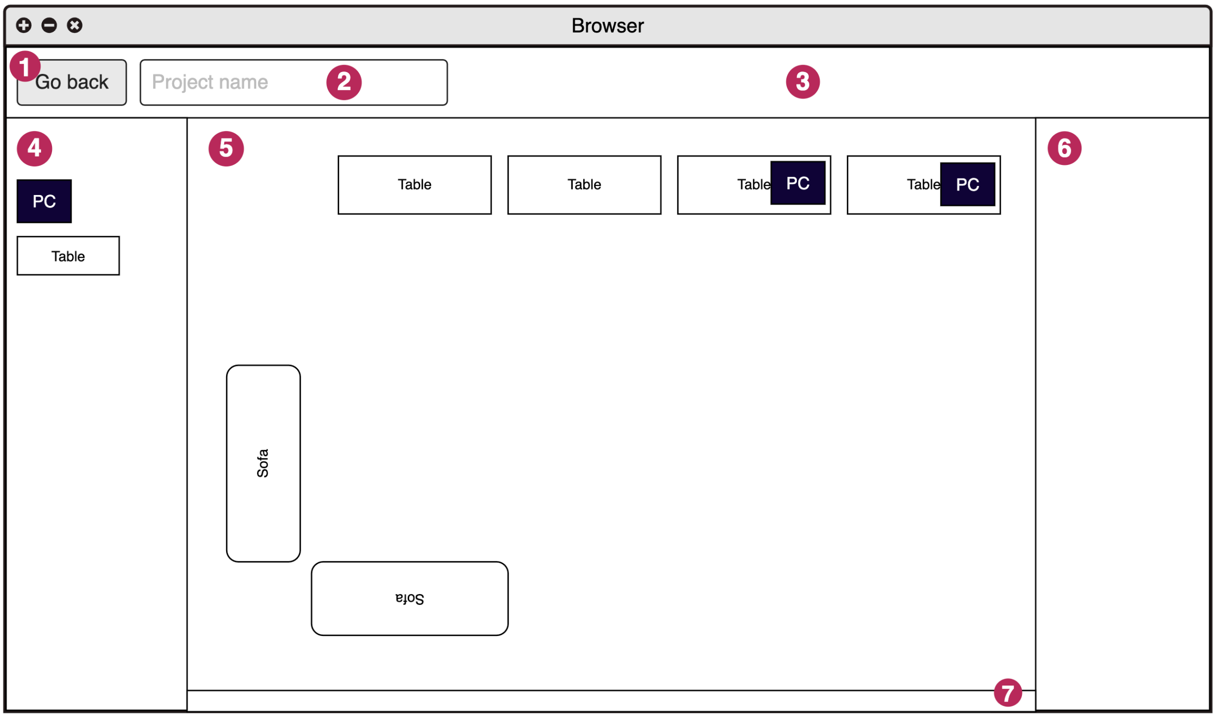


Figure 1 - Project Screen Layout

The image above shows the following elements and areas of the application:

1. Button to exit the project
2. Name of the project
3. Toolbar
4. Elements panel
5. Workspace
6. Project structure
7. Line with information

Button to exit the project - when you click on the button, the previous project with all projects should be opened.

Project name - this text field should display the actual project name obtained from the API. The user should be able to rename the project by changing the text content. The new project name should be automatically saved by requesting the API.

On the toolbar, you can arrange some functional buttons if you want.

The elements panel should display all available elements for placement on the workspace. Information about elements must be obtained from the public API.

In response from the API, you will receive the following information about the elements:

- ID
- Name
- Width - indicated in centimeters, use 1:1 scale
- Height - indicated in centimeters, use 1:1 scale
- Element code in SVG format

The work area is intended for arranging elements from the elements panel. The user should be able to drag the element he needs from the elements panel to the workspace and release it. After that, the element should be displayed in the workspace (Figure 2).

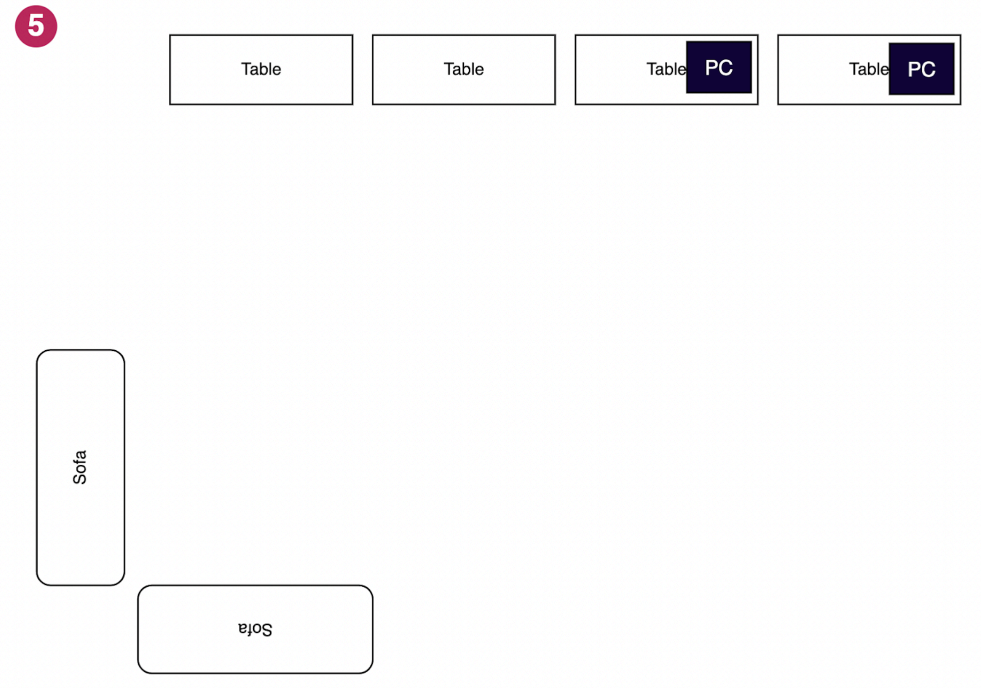


Figure 2 - An example of elements placed on the application workspace

The user should be able to drag and drop elements around the workspace, rotate and delete them. The active element should visually stand out from the rest.

Each element has an **is_basis** property with a value of **true** or **false**. If the property value is **true**, then such an element allows other elements to be placed on top of itself when moving (Figure 3).

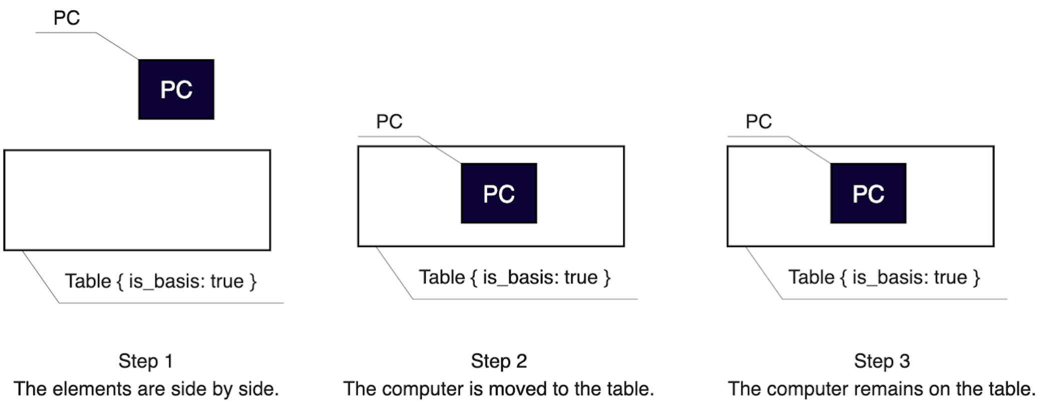


Figure 3 - An example of moving the "PC" element to the "Table" element, which allows you to place other elements on itself

When trying to move an element to an element with **is_basis** is **false**, the element being moved should go back (Figure 4).

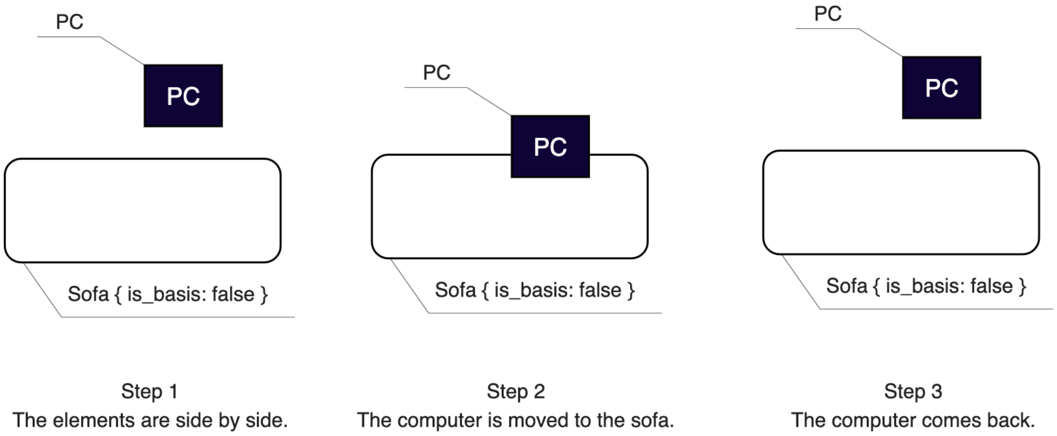


Figure 4 - An example of moving the "PC" element to the "Table" element, which prohibits placing other elements on itself

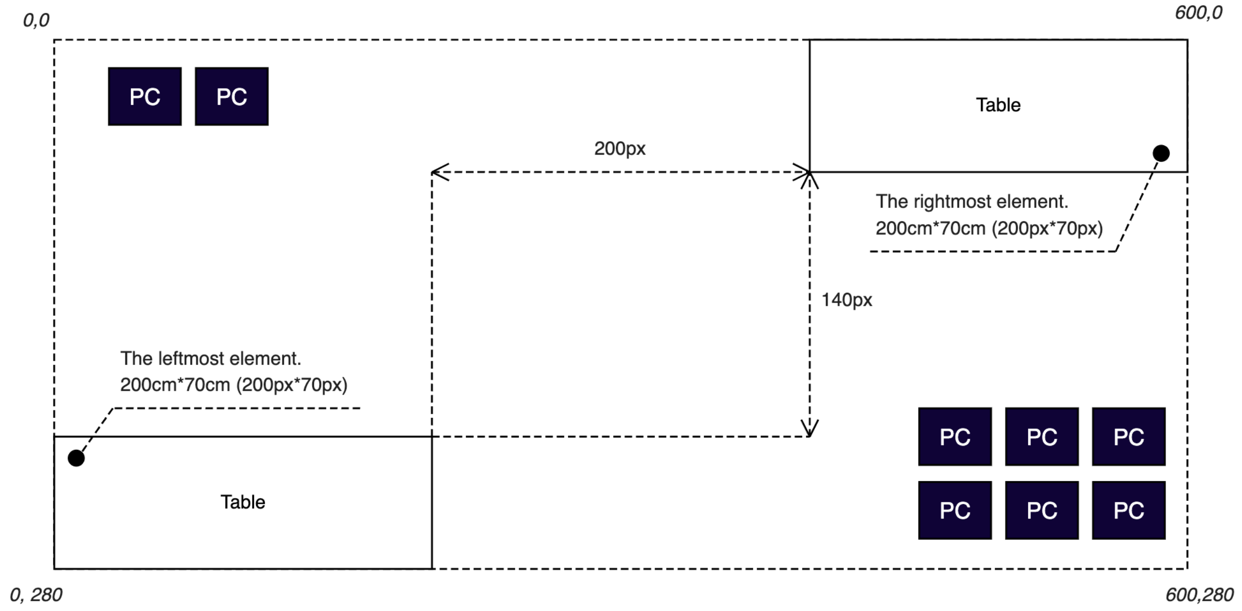
Deleting the active element should be possible by pressing the Del key on the keyboard.

All elements located in the workspace should also be displayed in the project structure in the form of a list for easier navigation through the project. When you click on an element in the project structure, it should be highlighted on the workspace too.

Pressing the ESC key should reset the active elements, i.e. no element should be active.

The user should be able to move the entire workspace in space using the spacebar and the left mouse button, as well as be able to zoom in and out of the workspace.

The line with information should display the area of the current development. The area to be calculated must be specified in square meters and calculated by the extreme elements in the workspace. An example of calculation is shown in Figure 5.



$$S = (200+200+200) * (70 + 140 + 70) / 10000 = 16{,}8 \text{ meters}$$

Figure 5 - An example of calculating the area

API

You are provided with a ready-made REST API for storing information about your projects.

Use http://nvafmzc-m1.wsr.ru as the {host} variable in request URLs.

Authentication

With this request, you can get an authentication token for your user. If you don't have a user yet, it will be created automatically based on your data.

```
URL: {host}/api/login
Method: POST
Request Headers:
    •   Accept: application/json
    •   Content-Type: application/json
Request body:
{
    “login”: <your login, required string>,
    “pin”: <your pin-code, required integer from 1000 to 9999>,
}
```

If there is a user with the given PIN code, then you will receive a token, otherwise such a user will be created.

As a result, you will receive the following response:

```
Response status: 200
Response body: {
   “data”: {
       “token”: <token string>
   }
}
```

If the user exists, but the PIN does not match, then you will receive the following response:

```
Response status: 422
Response body: {
   “errors”: {
       “pin”: “Incorrect pin-code”
   }
}
```

Getting a list of projects

With this query you can get a list of your projects.

```
URL: {host}/api/projects
Method: GET
Headers:
    •   Accept: application/json
    •   Authorization: Bearer <token>
Response body: {
   “data”: [
       {
          “id”: 1,
          “name”: “project name”
       }
   ]
}
```

Getting a project

With this request you can get all the information about a particular project.

```
URL: {host}/api/projects/<id>
Method: GET
Headers:
    •   Accept: application/json
    •   Authorization: Bearer <token>
Response body: {
   “data”: {
        “id”: 1,
       “name”: “project name”,
       “content”: <your project json>
   }
}
```

Create a new project

With this request, you can create a new project.

```
URL: {host}/api/projects
Method: POST
Headers:
    •   Accept: application/json
    •   Content-Type: application/json
    •   Authorization: Bearer <token>
Response code: 201
Response body: {
   “data”: {
       “id”: 1,
       “name”: “New project”,
       “content”: []
   }
}
```

Retrieving Items

With this query, you can get a list of all available elements with detailed information about them.

```
URL: {host}/api/elements
Method: GET
Headers:
    •   Accept: application/json
Response body: {
    “data”: [
       {
           “id”: <uuid>,
           “name”: “Table”,
           “width”: 200,
           “height”: 70,
           “is_basis”: <true or false>,
           “svg”: <svg code>
       }
    ]
}
```

Project update

With this request, you can update information about your project. The "name" and "content" fields are optional and can be sent together or separately.

```
URL: {host}/api/projects/<id>
Method: PATCH
Headers:
    •   Accept: application/json
    •   Content-Type: application/json
    •   Authorization: Bearer <token>
Request body: {
   “name”: “New name”,
   “content”: {}
}
Response code: 200
Response body: {
   “data”: {
      “id”: 1,
      “name”: “New name”,
      “content”: {}
   }
}
```

INSTRUCTIONS FOR THE COMPETITOR

Save your work on the server in the first module folder.

If you are using the project builder and your project code will be minified in the final build, then also save the project source code, except for external packages (node_modules, vendor, etc.) in the src folder.

The web application you developed should open at http://xxx-m1.wsr.ru, where xxx is your username.

Jobs not saved to the server, or jobs that were saved in error or deployed incorrectly, will not be reviewed. So make sure your service is up and running.

The web interface will be evaluated in the Google Chrome browser.

MARKING SCHEME

CRITERION	MEAS. MARKS	JUDG. MARKS	TOTAL
Layout	6,00	0,00	6,00
General functional	3,30	0,00	3,30
Project screen	13,70	0,00	13,70
Design and convenience	0,00	6,00	6,00
General	2,00	1,00	3,00
Code quality	0,00	3,00	3,00
TOTAL	**25,00**	**10,00**	**35,00**