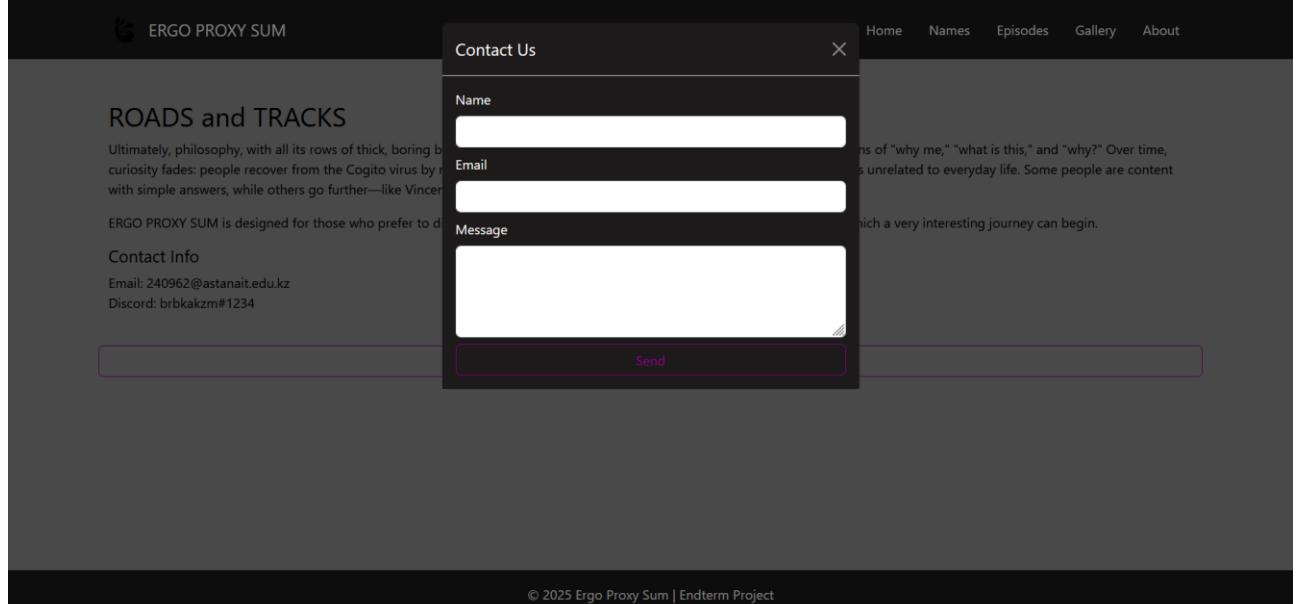


Final Project

Alexander Zholmukhametov
MT-2406

Form Validation



The screenshot shows a browser window with the title "ERGO PROXY SUM". Below the title, there's a navigation bar with links for Home, Names, Episodes, Gallery, and About. A modal dialog box is open, titled "Contact Us". It contains three input fields: "Name", "Email", and "Message". Below the "Message" field is a "Send" button. The background of the page shows some text and images related to the project.

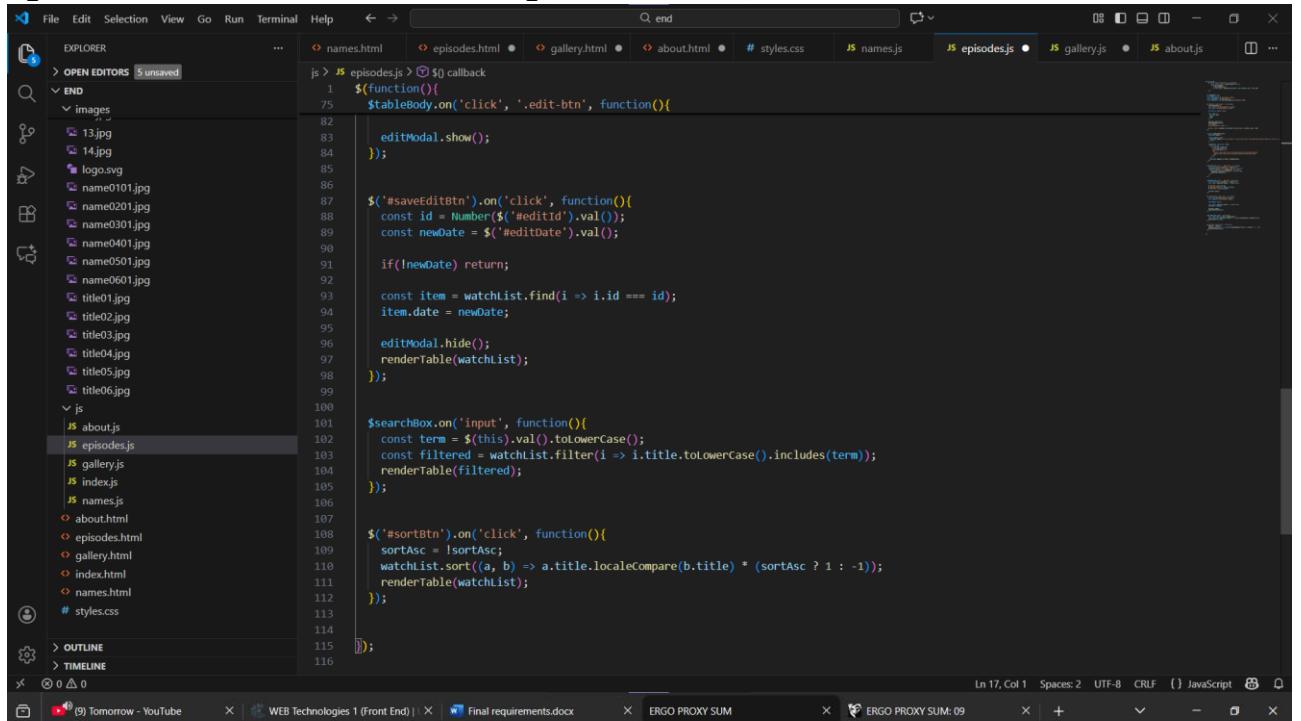
```
File Edit Selection View Go Run Terminal Help
EXPLORER OPEN EDITORS 5 unsaved
END images 13.jpg 14.jpg logo.svg name01.jpg name02.jpg name03.jpg name04.jpg name05.jpg name06.jpg title01.jpg title02.jpg title03.jpg title04.jpg title05.jpg title06.jpg
js about.js episodes.js gallery.js index.js names.js
about.html episodes.html gallery.html index.html names.html
# styles.css
OUTLINE TIMELINE
file:///C:/Users/superkorovamertva/Desktop/end/about.html
Ln 51, Col 4 Spaces: 4 UTF-8 CRLF JavaScript
(0) Comatose - YouTube WEB Technologies 1 (Front End) Final requirements.docx ERGO PROXY SUM ERGO PROXY SUM: 09
Home Names Episodes Gallery About
Contact Us
Name
Email
Message
Send
© 2025 Ergo Proxy Sum | Endterm Project
```

js > JS about.js ...
1 \$(function(){
2 function validateEmail(email){
3 return /^[^@]+@[^\s@]+\.[^\s@]+\\$/ .test(email);
4 }
5
6 \$('#contactForm').on('submit', function(e){
7 e.preventDefault();
8 const name = \$('#name').val().trim();
9 const email = \$('#email').val().trim();
10 const message = \$('#message').val().trim();
11
12 let ok = true;
13
14 if(name.length < 2){
15 ok = false;
16 \$('#name').addClass('is-invalid');
17 \$('#nameError').text('Please enter at least 2 characters.');//
18 } else {
19 \$('#name').removeClass('is-invalid');
20 \$('#nameError').text('');//
21 }
22
23 if(!validateEmail(email)){
24 ok = false;
25 \$('#email').addClass('is-invalid');
26 \$('#emailError').text('Enter a valid email address.');//
27 } else {
28 \$('#email').removeClass('is-invalid');
29 \$('#emailError').text('');//
30 }
31
32 if(message.length < 10){
33 ok = false;
34 \$('#message').addClass('is-invalid');
35 \$('#msgError').text('Message must be at least 10 characters.');//
36 } else {
37 // Additional validation logic here if needed

I made a contact form with custom validation using jquery and bootstrap including field checks for name, email and message. Email format verification uses Regular Expression. Form provides real-time

feedback and prevents submission if validation fails with corresponding alerts.

Dynamic Search Functionality



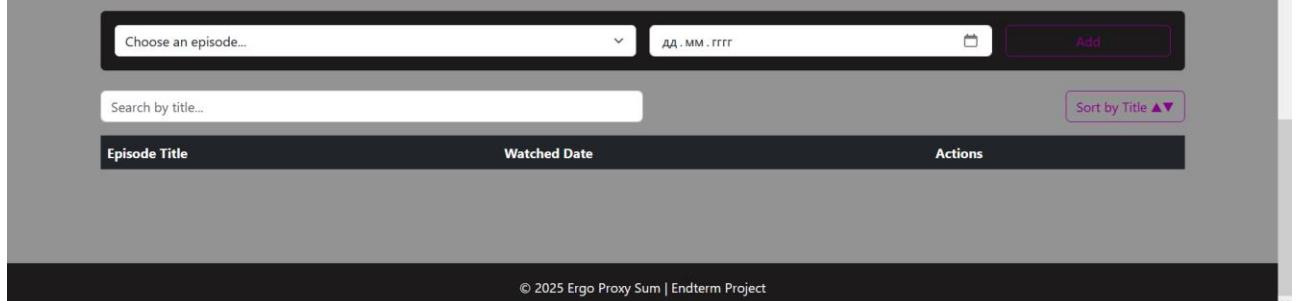
```

File Edit Selection View Go Run Terminal Help < - > end
EXPLORER > OPEN EDITORS 5 unsaved
END
images
13.jpg
14.jpg
logo.svg
name01.jpg
name02.jpg
name03.jpg
name04.jpg
name05.jpg
name06.jpg
title01.jpg
title02.jpg
title03.jpg
title04.jpg
title05.jpg
title06.jpg
js
about.js
episodes.js
gallery.js
index.js
names.js
about.html
episodes.html
gallery.html
index.html
names.html
# styles.css
names.html
episodes.html
gallery.html
about.html
# styles.css
names.js
episodes.js
gallery.js
index.js
names.html
# styles.css
OUTLINE
TIMELINE
Ln 17, Col 1 Spaces: 2 UTF-8 CRLF () JavaScript
file:///C:/Users/superkorovamerta/Desktop/end/episodes.html

```

Returning Home >>> Returning for the sake of salvation – for Reel, who might die; >>>
 Returning in someone else's place – for Houde (as Vincent); >>> Returning Home – for
 Houde himself and his wooden soldier; >>> Returning by Order – when the Director General
 of the Security Bureau returned the military drones to the airlock; >>> The Beginning of the
 Journey Home – for Vincent on the Usagi, although at that moment he doesn't yet know
 where he's headed; >>> Death, accomplished and impending – for the residents of the
 Community, because they had already "died" when they left the city: they have nowhere
 and no reason to strive; >>> the death of the "queen," who lost her "son" and saw the
 "princess" who would take her place; >>> the liberation that came with the Messenger of
 Death—the return of the dead to the dead.

My Watch List



Episode Title	Watched Date	Actions
Mediator VI	2023-10-01	Edit
Democracy	2023-10-02	Edit

© 2025 Ergo Proxy SUM | Endterm Project

I also implemented a real-time search bar that dynamically filters episode titles in watch list table using jquery.

Modal or Popup Interaction

```

<html lang="en">
  <body>
    <main class="container my-5">
      <div class="row">
        <button type="button" class="btn btn-outline-secondary mt-4" data-bs-toggle="modal" data-bs-target="#contactModal">
          Contact Us
        </button>
      </div>
      <div class="modal fade" id="contactModal" tabindex="-1" aria-hidden="true">
        <div class="modal-dialog">
          <div class="modal-content text-white">
            <div class="modal-header">
              <h5 class="modal-title">Contact UsNameEmailMessageMessage sent</div>
              </form>
            </div>
          </div>
        </div>
      </div>
    </main>
  </body>
</html>

```

The screenshot shows a code editor with the 'about.html' file open. The code uses Bootstrap's modal system to create a contact form. It includes fields for Name, Email, and Message, each with validation feedback. A 'Send' button triggers a success message. The code editor interface shows other files like 'names.html', 'episodes.html', 'gallery.html', and 'index.html' in the sidebar.

ERGO PROXY SUM

Contact Us

Name

Email

Message

Send

© 2025 Ergo Proxy Sum | Endterm Project

I used Bootstrap modals to display contact form and table editing. It can be closed with ESC key or cross button.

Dynamic Table with CRUD Operations

```

File Edit Selection View Go Run Terminal Help < > names.html episodes.html gallery.html about.html # styles.css JS names.js JS episodes.js JS gallery.js JS about.js ...
OPEN EDITORS 5 unsaved
EXPLORER END
images
13.jpg
14.jpg
logo.svg
name0101.jpg
name0201.jpg
name0301.jpg
name0401.jpg
name0501.jpg
name0601.jpg
title01.jpg
title02.jpg
title03.jpg
title04.jpg
title05.jpg
title06.jpg
js
about.js
episodes.js
gallery.js
index.js
names.js
about.html
episodes.html
gallery.html
index.html
names.html
# styles.css
OUTLINE
TIMELINE
Ln 17, Col 1 Spaces: 2 UTF-8 CRLF ( ) JavaScript
(9) Suffering - YouTube | WEB Technologies 1 (Front End) | Final requirements.docx | ERGO PROXY SUM | file:///C:/Users/superkorovamrvta/Desktop/end/episodes.html
Community, because they had already "died" when they left the city: they have nowhere and no reason to strive; >>> the death of the "queen," who lost her "son" and saw the "princess" who would take her place; >>> the liberation that came with the Messenger of Death—the return of the dead to the dead.

```

My Watch List

Episode Title	Watched Date	Actions
meditatio II - confession - 良き市民の告白	2025-11-05	<button>Edit</button> <button>Delete</button>
meditatio III - mazecity - 無への跳舞	2025-11-26	<button>Edit</button> <button>Delete</button>
meditatio V - TASOGARE - 召還	2025-11-05	<button>Edit</button> <button>Delete</button>

I created a Watch List table with CRUD operations where user can add an episode using dropdown, edit and delete inputs and also search particular entries. All by using jquery.

Image Gallery with Filtering and Lightbox

The screenshot shows a code editor with several files open in the sidebar: names.html, episodes.html, gallery.html, about.html, styles.css, names.js, episodes.js, gallery.js, and about.js. The gallery.js file is the active tab, displaying the following code:

```

js > JS gallery.js > JS callback > JS renderGallery > JS each() callback > JS item
1  $(function(){
2    const images = [
3      {src: 'images/1.jpg', category: 'character', title: '1'},
4      {src: 'images/2.jpg', category: 'fanart', title: '2'},
5      {src: 'Images/3.png', category: 'fanart', title: '3'},
6      {src: 'images/4.jpg', category: 'character', title: '4'},
7      {src: 'images/5.jpg', category: 'screenshot', title: '5'},
8      {src: 'images/6.jpg', category: 'fanart', title: '6'},
9      {src: 'images/7.jpg', category: 'character', title: '7'},
10     {src: 'images/8.jpg', category: 'screenshot', title: '8'},
11     {src: 'images/9.jpg', category: 'character', title: '9'},
12     {src: 'Images/10.jpg', category: 'screenshot', title: '10'},
13     {src: 'Images/11.jpg', category: 'fanart', title: '11'},
14     {src: 'images/12.jpg', category: 'fanart', title: '12'},
15     {src: 'images/13.jpg', category: 'character', title: '13'},
16     {src: 'images/14.jpg', category: 'fanart', title: '14'}
17   ];
18
19   const $gallery = $('#gallery-container');
20   const $lightbox = $('.lightbox');
21   const $lightboxImg = $('#lightbox-img');
22   let currentIndex = 0;
23
24
25   function renderGallery(items){
26     $gallery.empty();
27     $.each(items, function(i, img){
28       const item = $('');
29       item.hide().appendTo($gallery).fadeIn(400);
30     });
31   }
32
33   renderGallery(images);
34
35 }
36
37 
```

The browser window below shows the 'ERGO PROXY SUM' gallery page at <file:///C:/Users/superkorovamertya/Desktop/end/gallery.html>. The page has a header with links for Home, Names, Episodes, Gallery, and About. Below the header is a section titled 'GALLERY' with the subtext 'Wallpapers and arts made by community'. There is a navigation bar with tabs: All (selected), Screenshots, Fanarts, and Characters. Six images are displayed in a grid:

- A woman holding a gun.
- A scene from Ergo Proxy.
- A man and a woman.
- A woman's face.
- A couple.
- A man's face.

I upgraded my image gallery with filtering by category and lightbox effect to watch a single image. Also implemented scrolling with arrow keys and added fade animations.

Animation and Visual Feedback

The screenshot shows the Visual Studio Code interface. The left sidebar displays a file tree with several image files (e.g., 13.jpg, 14.jpg, logo.svg, name0101.jpg, etc.) and several JavaScript files (e.g., about.js, episodes.js, gallery.js, index.js, names.js). The right pane shows the content of the 'names.js' file:

```
js > names.js > $() callback
1  $(function(){
2    $('a[href^="#"]').on('click', function(e){
3      const target = $(this.getAttribute('href'));
4      if(target.length){
5        e.preventDefault();
6        $('html,body').animate({scrollTop: target.offset().top - 70}, 400);
7      }
8    });
9  );
10
11
12
```

Implemented scrolling animations for Names and Episodes pages and also image animations for characters.

Conclusion

In this final project I tried to combine all my knowledge gained throughout the course such as Flex/Grid layout, CSS styling, Bootstrap elements, JS and jQuery scripts and animations.