**INTERNET PROGRAMMING**Laboratory work 1-4  
Variant – 7

Prepared by:

student of *Ba-121-22-4-SE*,

*Anna Kuts`*

Accepted by:

*Makliuk O.O.*

**PART 1: FORMS   
Task**

1. Get acquainted with the concepts of form, control, rules in HTML.

2. Study the declaration and instruction syntax of forms, control elements.

3. Build an HTML page that displays the task corresponding to the

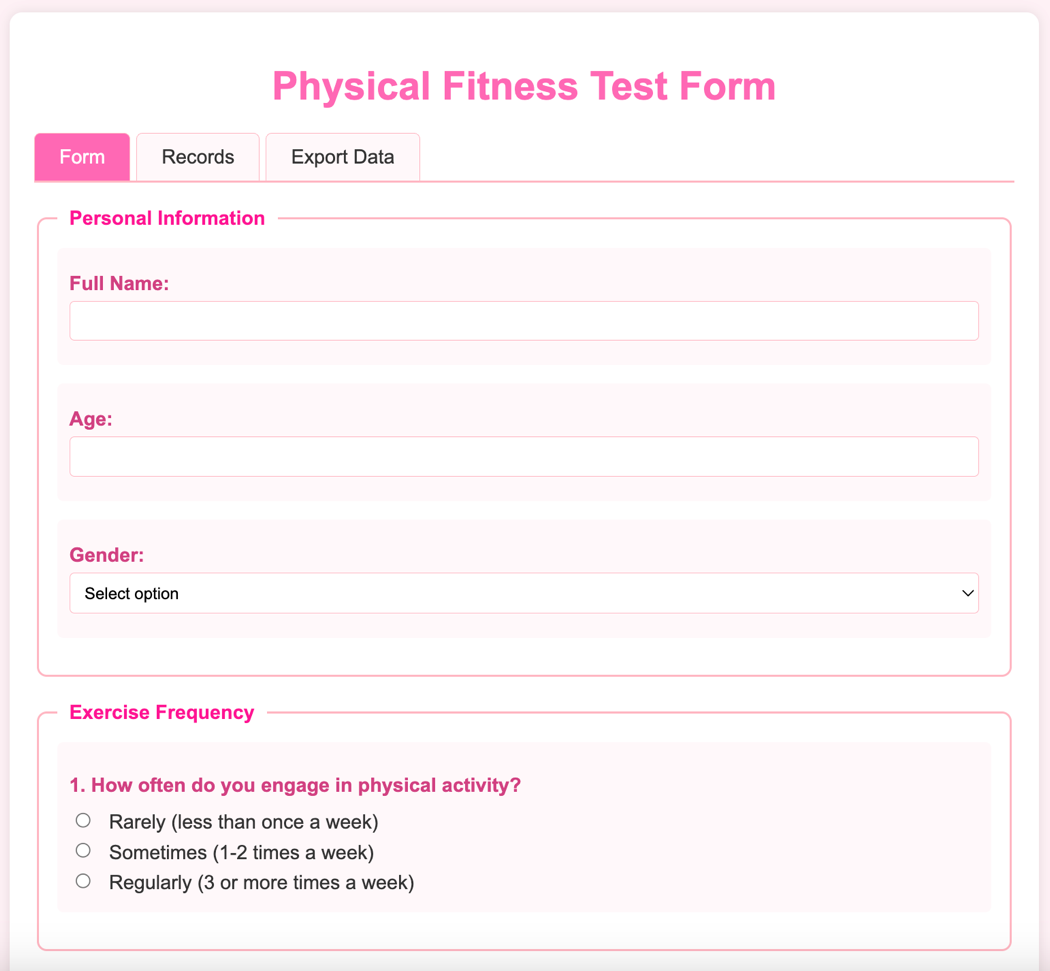
variant.

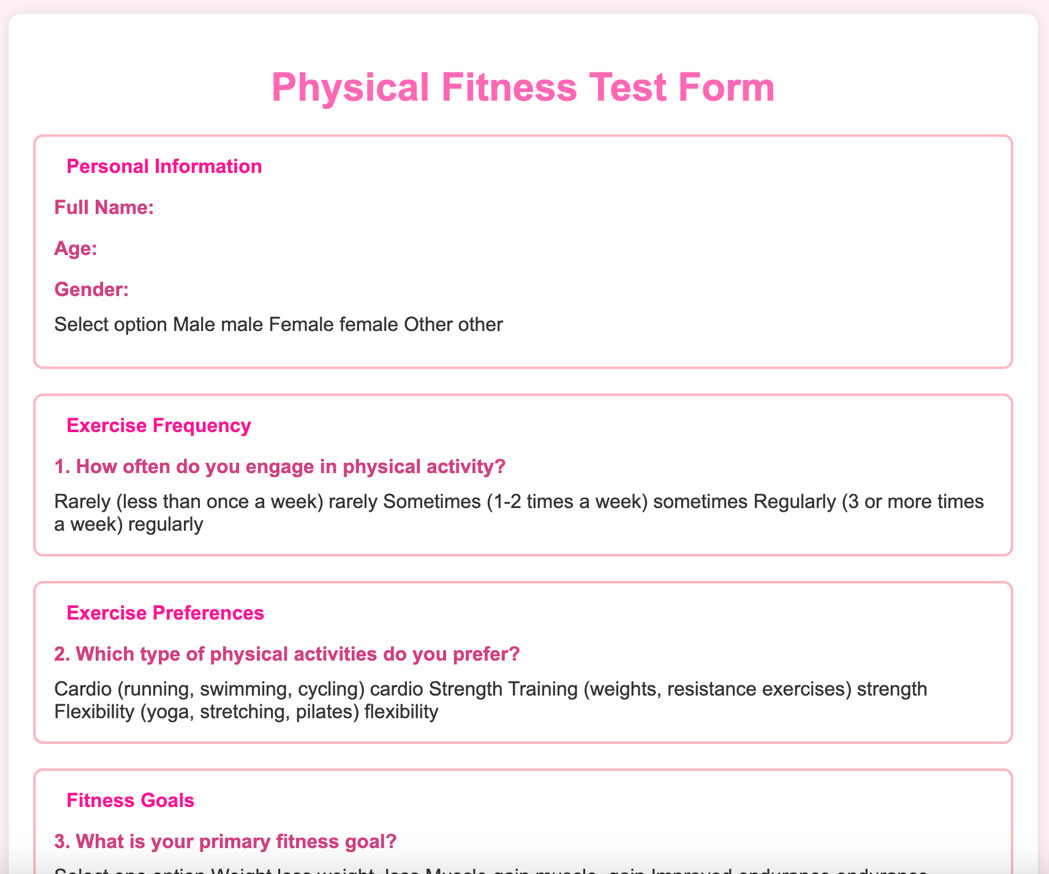
4. Rewrite your form according to XForm

Create a form that represents a physical test. Each question should have three answer options. You should use different types of controls throughout the task.

**Solution**

HTML:



XML:  
  
<https://github.com/Jasokaa/IP-Olenin>

**Conclusion**

In this work, I learnt the declaration and use of different types of form elements in HTML.

1. Define a form, label, and control.

* Form: An HTML element (<form>) used to collect user input and send it to a server. It can include various input elements like text fields, buttons, checkboxes, etc.
* Label: An HTML element (<label>) that defines a caption for a form control. It improves accessibility by associating a text description with the corresponding input field.
* Control: Any HTML form element that allows user interaction/input, such as <input>, <select>, <textarea>, etc.

2. Define the types of main controls.  
Main types of form controls include:

* Text controls: <input type="text">, <textarea>
* Button controls: <button>, <input type="submit">, <input type="reset">
* Selection controls: <select>, <option>, <checkbox>, <radio>
* File controls: <input type="file">
* Hidden controls: <input type="hidden">

3. Features and usage examples of the <select> element.

* Allows users to choose one or multiple options from a dropdown list.
* Can be used with <option> and <optgroup> elements.
* Supports single and multiple selections.

Example:

<select name="country">

<option value="ua">Ukraine</option>

<option value="pl">Poland</option>

</select>

4. Possible uses of the <optgroup> element.

* Groups related options within a <select> element to improve readability and user experience.
* Adds a label to each group of options.

Example:

<select>

<optgroup label="Europe">

<option>Ukraine</option>

<option>Germany</option>

</optgroup>

<optgroup label="Asia">

<option>Japan</option>

<option>China</option>

</optgroup>

</select>

5. What are <legend> and <fieldset> elements used for?

* <fieldset>: Groups related form elements together visually and semantically.
* <legend>: Provides a caption or title for the <fieldset> group.

Example:

<fieldset>

<legend>Personal Info</legend>

<label>Name: <input type="text" name="name"></label>

</fieldset>

6. Provide an example of using the <label> element.  
Example:

<label for="email">Email:</label>

<input type="email" id="email" name="email">

This connects the label to the input, improving usability and accessibility.

7. What are the main elements of XForms?  
XForms is an XML-based language for building web forms. Main elements include:

* <xforms:model> – Defines the data model and constraints.
* <xforms:instance> – Contains the initial data.
* <xforms:bind> – Binds data to form controls and defines validation rules.
* <xforms:input>, <xforms:select>, etc. – UI controls for data input.
* <xforms:submit> – Submits the form data.

**PART 2: FORMS AND JAVASCRIPT**

**Task**

1. Get to know JavaScript tools for processing form data.

2. Creating form data (see option in Table 1)

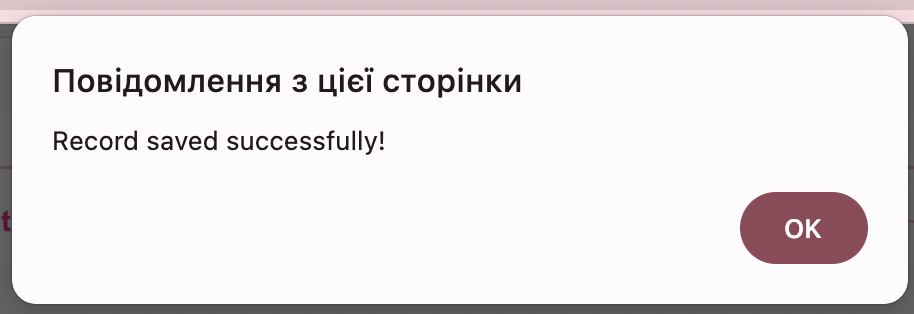
3. Updating data.

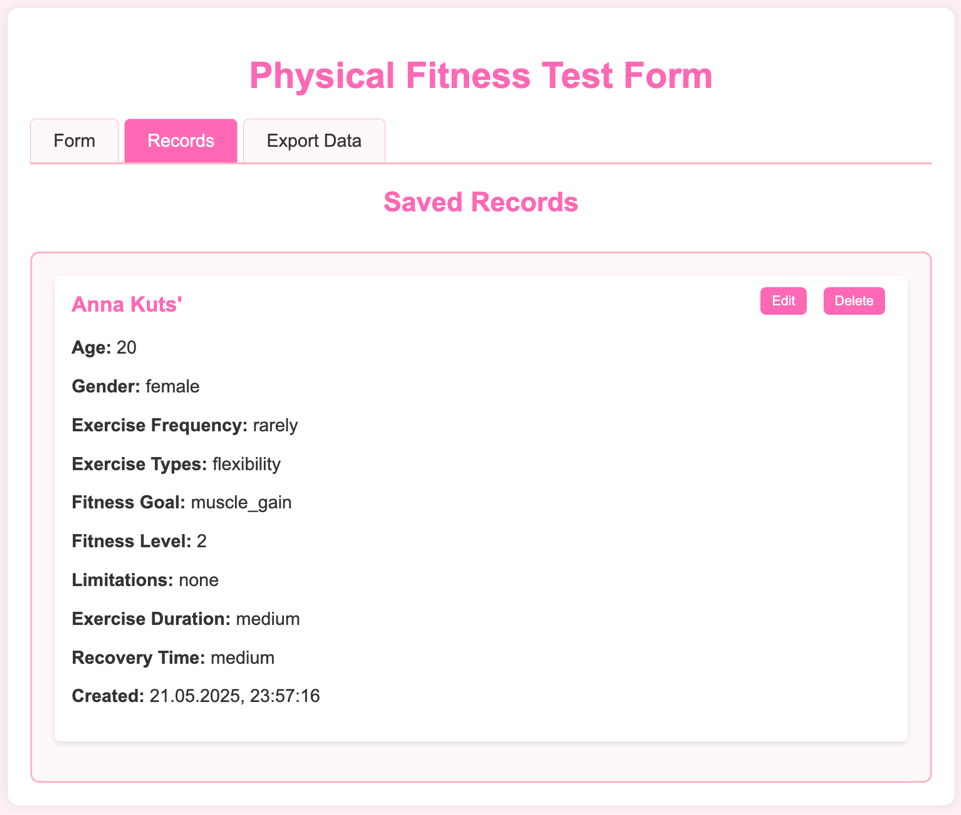
4. Saving form data to files (.xml, .json).

5. Reading records.

6. Deleting records.

**Solution**

****

****

<https://github.com/Jasokaa/IP-Olenin>

**Conclusion**

In this work, I filled form and form data processing using JavaScript.

**1. Which attribute of the** <input> **element specifies the URL of the file that will process the input data when the form is submitted?**  
**This is not an attribute of the** <input> **element**, but rather of the <form> element.

* **Attribute**: action
* **Example**:

<form action="submit.php" method="post">

<input type="text" name="username">

</form>

Here, submit.php is the URL that will process the form data.

**2. Which attribute of the** <input> **element specifies how the form data should be encoded when submitted (only for forms with** method="post"**)?**  
This is **not an** <input> **attribute**, but an attribute of the <form> element.

* **Attribute**: enctype
* **Common values**:
  + application/x-www-form-urlencoded (default)
  + multipart/form-data (for file uploads)
  + text/plain

**Example:**

<form action="upload.php" method="post" enctype="multipart/form-data">

<input type="file" name="image">

</form>

**3. How are form data stored in an XML file?**  
To store form data in XML, you would convert the data into XML format manually or programmatically (e.g., with JavaScript or server-side code).

**Example XML:**

<formData>

<name>John</name>

<email>john@example.com</email>

</formData>

You typically collect the data using JavaScript or a backend language, then format it into XML before saving or sending it.

**4. How is form data saved in a JSON structure?**  
Form data can be collected and converted to JSON using JavaScript. This is useful for APIs or storing structured data.

**Example JSON:**

{

"name": "John",

"email": "john@example.com"

}

**Example with JavaScript:**

const form = document.querySelector('form');

const formData = new FormData(form);

const jsonData = Object.fromEntries(formData.entries());

console.log(JSON.stringify(jsonData));