

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

**Інститут КНІТ  
Кафедра ПЗ**

**ЗВІТ**

До лабораторної роботи № 2

**З дисципліни:** *“Програмування в Інтернет”*

**На тему:** *“Форма як головний елемент динаміки та зворотнього зв’язку з сервером”*

**Лектор:**

викл. каф. ПЗ  
Степанов Д. С.

**Виконав:**

ст. гр. ПЗ-22  
Чаус Олег

**Прийняла:**

ст. викл. каф. ПЗ  
Грицай О. Д.

« \_\_\_\_ » \_\_\_\_\_ 2023 р.

Σ= \_\_\_\_ .

Львів – 2023

**Тема роботи:** Форма як головний елемент динаміки та зворотнього зв'язку з сервером.

**Мета роботи:** оволодіти засобами керування формами, внесення інформації для сервера на динамічній web-сторінці, ознайомитись з функціями JavaScript та бібліотеки jQuery.

## ЗАВДАННЯ

Розробити web-сторінку згідно макета (wireframe)

<https://cacao.com/diagrams/ZvVhYS3UpG5PdbBy/EDE3A>

Обираємо для розробки шаблон Students Add/Edit

Використовуємо Gitlab (<https://gitlab.com>)

Сторінка має містити:

- модальне вікно для внесення даних як на макеті з прихованим полем id
- для редагування даних використовується те саме вікно
- кнопки у модальному вікні та реакції на ці кнопки функціями JavaScript/jQuery.
- функція додавання/збереження має сформулювати рядок даних, що буде передаватись з даної форми на сервер.
- створити стандартний опис з будь яким кешуванням PWA

## ХІД ВИКОНАННЯ

1. Створив макет форми, використовуючи HTML.

```
<form class="d-none" id="edit-form">
  <div class="row my-3 g-0">
    <label class="form-label col-xl-3 col-form-label"
for="group-select">Group</label>
    <select class="form-select col-xl" id="group-select" required
aria-label="Select Group">
      <option value="" selected disabled>Select Group</option>
      <option>KN-21</option>
      <option>PZ-21</option>
      <option>PZ-22</option>
    </select>
  </div>
  <div class="row mb-3 g-0">
    <label class="form-label col-xl-3 col-form-label"
for="name-input">Name</label>
    <input type="text" id="name-input" class="form-control col-xl"
required>
  </div>
  <div class="row mb-3 g-0">
```

```

        <label class="form-label col-xl-3 col-form-label"
for="surname-input">Surname</label>
        <input type="text" id="surname-input" class="form-control col-xl"
required>
    </div>
    <div class="row mb-3 g-0">
        <label class="form-label col-xl-3 col-form-label"
for="group-select">Gender</label>
        <select class="form-select col-xl" id="gender-select" required
aria-label="Select Gender">
            <option value="" selected disabled>Select Gender</option>
            <option>Male</option>
            <option>Female</option>
            <option>Non-Binary</option>
        </select>
    </div>
    <div class="row mb-3 g-0">
        <label class="form-label col-xl-3 col-form-label"
for="date-select">Birthday</label>
        <input type="date" id="date-select" class="form-control col-xl"
required>
    </div>
    <div class="d-flex justify-content-end">
        <button type="submit" class="btn btn-outline-success me-2
button-submit">Submit</button>
        <button type="button" class="btn btn-outline-danger
button-cancel">Cancel</button>
    </div>
</form>

```

## 2. Створив код для створення/редагування студентів.

```

function isAssigned(id) {
    return studentIDs.has(id);
}

function generateUniqueID() {
    let id = Math.floor(Math.random() * 100000).toString().padStart(5, "0");
    while (isAssigned(id)) {
        id = Math.floor(Math.random() * 100000).toString().padStart(5, "0");
    }
}

```

```

    }
    return id;
}

function populateForm(form, student) {
    form.find("#group-select").val(student.group);
    form.find("#name-input").val(student.name);
    form.find("#surname-input").val(student.surname);
    form.find("#gender-select").val(student.gender);
    form.find("#date-select").val(student.birthday);
}

function createRow(student) {
    const newRow = $("<tr>");
    newRow.append($("<td>").append($("<input>").attr("type", "checkbox")));
    for (let i = 0; i < 4; i++) {
        newRow.append($("<td>"));
    }
    newRow.append($("<td>").append($("<div>").addClass("green-dot")));

    newRow.append($("<td>").append($("#button-wrapper").clone().removeClass("d-none")));
    return newRow;
}

function populateRow(row, student) {
    row.find("td:nth-child(2)").text(student.group);
    row.find("td:nth-child(3)").text(`${student.name} ${student.surname}`);
    row.find("td:nth-child(4)").text(student.gender);
    row.find("td:nth-child(5)").text(student.birthday);
}

function logAction(action, student) {
    let thisAction;
    if (action === "delete") {
        thisAction = {
            action: action,
            id: student.id

```

```

    };
  } else {
    thisAction = {
      action: action,
      student: student
    };
  }
  console.log(JSON.stringify(thisAction));
}

function createForm(name, id = -1) {
  id = Number(id);
  const notificationWindow = createPopup(name, "", []);
  $(notificationWindow).children("p, .buttons").remove();
  const newEditForm = $("#edit-form").clone();
  newEditForm.removeAttr("class");
  $(newEditForm).find(".button-cancel").click(closePopup);
  $(notificationWindow).children("h2").after(newEditForm);

  let action;
  let selectedStudent;
  if (id >= 0) {
    action = "edit";
    const selectedStudent = students.find((student) => Number(student.id)
=== Number(id));

    populateForm(newEditForm, selectedStudent);

    newEditForm.submit((e) => {
      e.preventDefault();
      selectedStudent.updateFromForm(newEditForm);
      const thisRow = $('tr[data-id=${selectedStudent.id}]');
      populateRow(thisRow, selectedStudent);
      logAction(action, selectedStudent);
      closePopup();
    });
  }
  else {

```

```

    action = "add";
    newEditForm.submit((e) => {
        e.preventDefault();
        selectedStudent = new Student();
        selectedStudent.updateFromForm(newEditForm);
        selectedStudent.id = generateUniqueID();

        const newRow = createRow();
        populateRow(newRow, selectedStudent);

        students.push(selectedStudent);
        studentIDs.add(selectedStudent.id);

        $(newRow).attr("data-id", selectedStudent.id);
        $("#students-table > tbody").append(newRow);

        $(newRow).find(".edit-button").click(function () {
            createForm("Edit student",
$(this).parent().parent().parent().data("id"));
        })
        $(newRow).find(".delete-button").click(function () {
            deleteStudent($(this).parent().parent().parent().data("id"));
        });
        logAction(action, selectedStudent);
        closePopup();
    });
}

$("#add-button").on("keydown", function (event) {
    if (event.keyCode === 13) {
        event.preventDefault();
    }
});

$("#add-button").click(function () {
    createForm("Add student");

```

```
});
```

3. Створив стрічки, що будуть передаватися на сервер при додаванні, редагуванні та видаленні.

```
{ "action": "add", "student": { "group": "KN-21", "name": "Oleh", "surname": "Chaus", "gender": "Male", "birthday": "2023-03-14", "id": "65258" } }
{ "action": "edit", "student": { "group": "PZ-22", "name": "Oleh", "surname": "Chaus", "gender": "Male", "birthday": "2004-04-01", "id": "65258" } }
{ "action": "delete", "id": "65258" }
```

4. Додав кешування PWA.

manifest.json

```
{
  "name": "Students Central",
  "short_name": "Students Central",
  "start_url": "/",
  "display": "standalone",
  "background_color": "#ffffff",
  "theme_color": "#212529",
  "icons": [
    {
      "src": "/images/icons/icon-48-48.png",
      "type": "image/png",
      "sizes": "48x48",
      "purpose": "any"
    },
    {
      "src": "/images/icons/icon-72-72.png",
      "type": "image/png",
      "sizes": "72x72",
      "purpose": "any"
    },
    {
      "src": "/images/icons/icon-96-96.png",
      "type": "image/png",
      "sizes": "96x96",
      "purpose": "any"
    },
    {
      "src": "/images/icons/icon-144-144.png",
      "type": "image/png",
      "sizes": "144x144",
      "purpose": "any"
    }
  ]
}
```

```

    },
    {
      "src": "/images/icons/icon-192-192.png",
      "type": "image/png",
      "sizes": "192x192",
      "purpose": "any"
    },
    {
      "src": "/images/icons/icon-512-512.png",
      "type": "image/png",
      "sizes": "512x512",
      "purpose": "any"
    }
  ]
}

```

sw.js

```

// Install service worker
self.addEventListener('install', (event) => {
  event.waitUntil(
    caches.open('my-pwa-cache')
      .then((cache) => {
        return cache.addAll([
          '/',
          '/index.html',
          '/app.css',
          '/app.js',
          '/images/logo.png'
        ]);
      })
  );
});

// Fetch resources with cache fallback
self.addEventListener('fetch', (event) => {
  event.respondWith(
    caches.match(event.request)
      .then((response) => {
        if (response) {

```



```

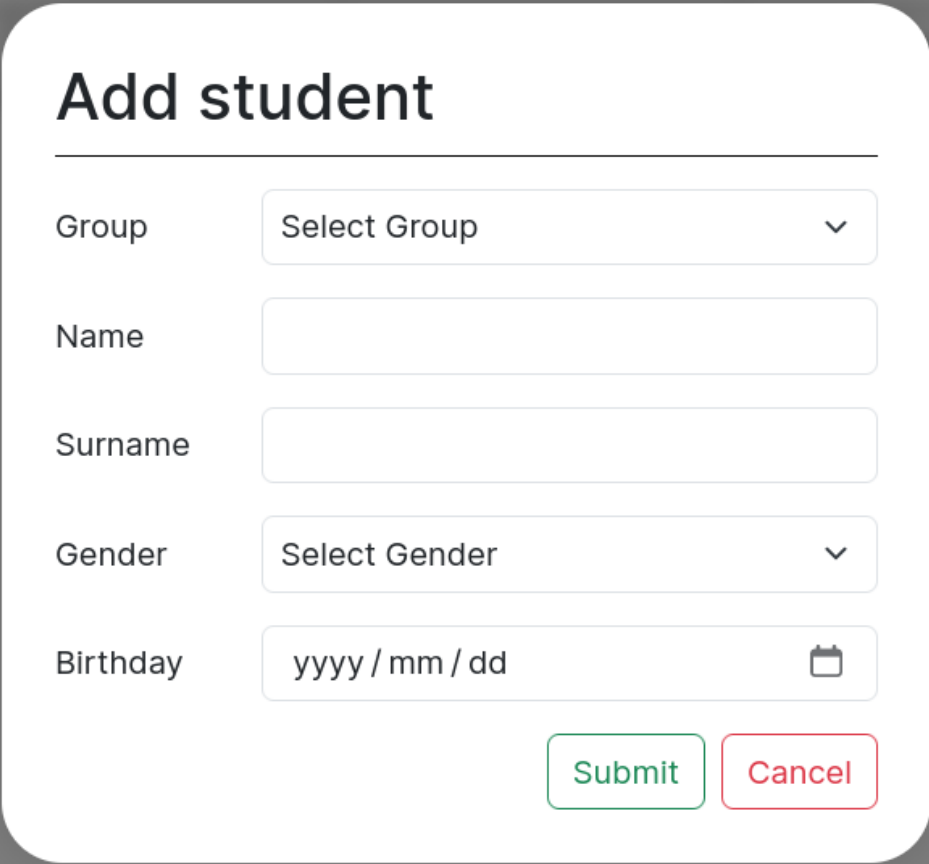
        return response;
    }
    return fetch(event.request)
        .then((response) => {
            return caches.open('my-pwa-cache')
                .then((cache) => {
                    cache.put(event.request,
response.clone());

                    return response;
                });
        });
    });
});

// Update cache when new service worker is activated
self.addEventListener('activate', (event) => {
    event.waitUntil(
        caches.keys().then((cacheNames) => {
            return Promise.all(
                cacheNames.filter((cacheName) => {
                    return cacheName.startsWith('my-pwa-cache-') &&
                        cacheName !== 'my-pwa-cache-v1';
                }).map((cacheName) => {
                    return caches.delete(cacheName);
                })
            );
        });
    );
});
});

```

## РЕЗУЛЬТАТИ



The image shows a 'Add student' form with a white background and rounded corners, set against a dark gray background. The form contains five input fields: a dropdown for 'Group', and text inputs for 'Name', 'Surname', and 'Gender'. The 'Birthday' field is a date picker showing 'yyyy / mm / dd'. At the bottom right are 'Submit' and 'Cancel' buttons.

## Add student

Group

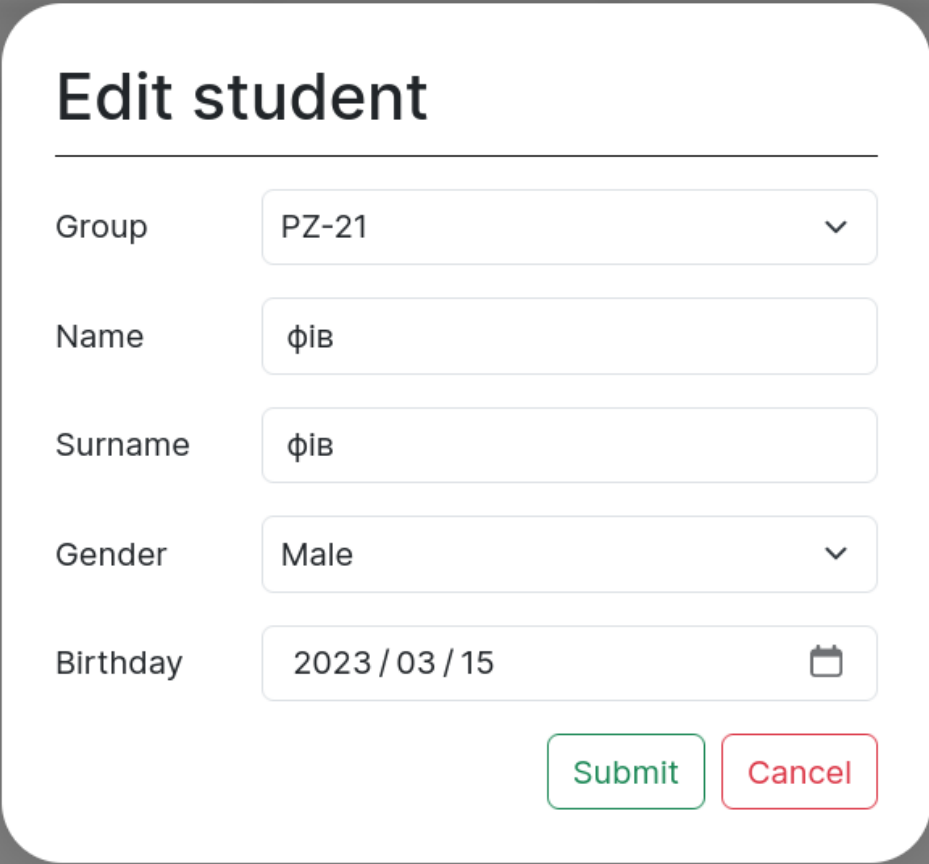
Name

Surname

Gender

Birthday

*Рис. 1. Форма додавання*



## Edit student

Group

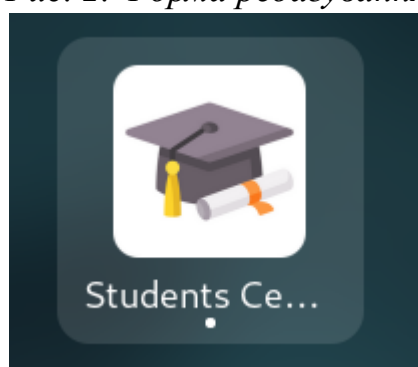
Name

Surname

Gender

Birthday

*Рис. 2. Форма редагування*



*Рис. 3. Додаток PWA*

[Посилання на репозиторій в GitLab](#)

## ВИСНОВКИ

Під час виконання лабораторної роботи я вивчив основні концепції та навички роботи з формами, функціями JavaScript та бібліотекою jQuery, що дозволило мені

створити інтерактивні елементи та забезпечити взаємодію між користувачем та веб-сторінкою.