

Реализация алгоритмов обработки данных

Сделать обработку данных между классами, моделями и страницами

Students.php:

```
// --- STUDENTS ---
const tableBodyStudents = $('#students-table tbody');

function loadStudents() {
    $.get('/UP-08-Puchnin-Kriulin/controllers/api/student_api.php', function (data) {
        tableBodyStudents.empty();
        data.forEach(student => {
            const row = `
                <tr data-id="${student.student_id}">
                    <td>${student.student_id}</td>
                    <td contenteditable="true" class="edit-last-name">${student.last_name}</td>
                    <td contenteditable="true" class="edit-first-name">${student.first_name}</td>
                    <td contenteditable="true" class="edit-middle-name">${student.middle_name || ''}</td>
                    <td contenteditable="true" class="edit-group-id">${student.group_id || ''}</td>
                    <td contenteditable="true" class="edit-dismissal-date">${student.dismissal_date || ''}</td>
                    <td>
                        <button class="save-btn save-btn-students">Сохранить</button>
                        <button class="delete-btn delete-btn-students">Удалить</button>
                    </td>
                </tr>`;
            tableBodyStudents.append(row);
        });
    }, 'json');
}

if ($('#students-table').length > 0) {
    loadStudents();
}

$('#add-student-form').on('submit', function (e) {
    e.preventDefault();
    const formData = $(this).serializeArray();
    const data = { action: 'create' };
    formData.forEach(field => data[field.name] = field.value);
    $.post('/UP-08-Puchnin-Kriulin/controllers/api/student_api.php', JSON.stringify(data), function () {
        $('#add-student-form')[0].reset();
        loadStudents();
    });
});

$(document).on('click', '.save-btn-students', function () {
    const row = $(this).closest('tr');
    const id = row.data('id');

    const student = {
        action: 'update',
        student_id: id,
        last_name: row.find('.edit-last-name').text(),
        first_name: row.find('.edit-first-name').text(),
        middle_name: row.find('.edit-middle-name').text() || null,
        group_id: row.find('.edit-group-id').text() || null,
        dismissal_date: row.find('.edit-dismissal-date').text() || null
    };

    $.post('/UP-08-Puchnin-Kriulin/controllers/api/student_api.php', JSON.stringify(student), function () {
        loadStudents();
    });
});

$(document).on('click', '.delete-btn-students', function () {
    const id = $(this).closest('tr').data('id');
    const data = { action: 'delete', student_id: id };
    $.post('/UP-08-Puchnin-Kriulin/controllers/api/student_api.php', JSON.stringify(data), function () {
        loadStudents();
    });
});
});
```

Disciplines.php:

```
// --- DISCIPLINES ---
const tableBodyDisciplines = $('#disciplines-table tbody');

function loadDisciplines() {
    $.get('/UP-08-Puchnin-Kriulin/controllers/api/discipline_api.php', function (data) {
        tableBodyDisciplines.empty();
        data.forEach(discipline => {
            const row = `
                <tr data-id="${discipline.discipline_id}">
                    <td>${discipline.discipline_id}</td>
                    <td contenteditable="true" class="edit-discipline-name">${discipline.discipline_name}</td>
                    <td>
                        <button class="save-btn save-btn-disciplines">Сохранить</button>
                        <button class="delete-btn delete-btn-disciplines">Удалить</button>
                    </td>
                </tr>`;
            tableBodyDisciplines.append(row);
        });
    }, 'json');
}

if ($('#disciplines-table').length > 0) {
    loadDisciplines();
}
```

```

$('#add-discipline-form').on('submit', function (e) {
  e.preventDefault();
  const formData = $(this).serializeArray();
  const data = { action: 'create' };
  formData.forEach(field => data[field.name] = field.value);
  $.post('/UP-08-Puchnin-Kriulin/controllers/api/discipline_api.php', JSON.stringify(data), function () {
    $('#add-discipline-form')[0].reset();
    loadDisciplines();
  });
});

$(document).on('click', '.save-btn-disciplines', function () {
  const row = $(this).closest('tr');
  const id = row.data('id');
  const discipline = {
    action: 'update',
    discipline_id: id,
    discipline_name: row.find('.edit-discipline-name').text()
  };
  $.post('/UP-08-Puchnin-Kriulin/controllers/api/discipline_api.php', JSON.stringify(discipline), function () {
    loadDisciplines();
  });
});

$(document).on('click', '.delete-btn-disciplines', function () {
  const id = $(this).closest('tr').data('id');
  const data = { action: 'delete', discipline_id: id };
  $.post('/UP-08-Puchnin-Kriulin/controllers/api/discipline_api.php', JSON.stringify(data), function () {
    loadDisciplines();
  });
});

```

Absences.php:

```

//---ABSENCES---
const tableBodyAbsences = $('#absences-table tbody');

function loadAbsences() {
  $.get('/UP-08-Puchnin-Kriulin/controllers/api/absence_api.php', function (data) {
    tableBodyAbsences.empty();
    data.forEach(absence => {
      const row = `
        <tr data-id="${absence.absence_id}">
          <td>${absence.absence_id}</td>
          <td contenteditable="true" class="edit-lesson-id">${absence.lesson_id}</td>
          <td contenteditable="true" class="edit-student-id">${absence.student_id}</td>
          <td contenteditable="true" class="edit-minutes-missed">${absence.minutes_missed}</td>
          <td contenteditable="true" class="edit-explanatory-note-path">${absence.explanatory_note_path || ''}</td>
          <td>
            <button class="save-btn save-btn-absences">Сохранить</button>
            <button class="delete-btn delete-btn-absences">Удалить</button>
          </td>
        </tr>`;
      tableBodyAbsences.append(row);
    });
  }, 'json');
}

if ($('#absences-table').length > 0) {
  loadAbsences();
}

```

```

$('#add-absence-form').on('submit', function (e) {
  e.preventDefault();
  const formData = $(this).serializeArray();
  const data = { action: 'create' };
  formData.forEach(field => data[field.name] = field.value);
  $.post('/UP-08-Puchnin-Kriulin/controllers/api/absence_api.php', JSON.stringify(data), function () {
    $('#add-absence-form')[0].reset();
    loadAbsences();
  });
});

$(document).on('click', '.save-btn-absences', function () {
  const row = $(this).closest('tr');
  const id = row.data('id');
  const absence = {
    action: 'update',
    absence_id: id,
    lesson_id: row.find('.edit-lesson-id').text(),
    student_id: row.find('.edit-student-id').text(),
    minutes_missed: row.find('.edit-minutes-missed').text(),
    explanatory_note_path: row.find('.edit-explanatory-note-path').text() || null
  };
  $.post('/UP-08-Puchnin-Kriulin/controllers/api/absence_api.php', JSON.stringify(absence), function () {
    loadAbsences();
  });
});

$(document).on('click', '.delete-btn-absences', function () {
  const id = $(this).closest('tr').data('id');
  const data = { action: 'delete', absence_id: id };
  $.post('/UP-08-Puchnin-Kriulin/controllers/api/absence_api.php', JSON.stringify(data), function () {
    loadAbsences();
  });
});

```

StudentGroups.php

```
// --- STUDENT GROUPS ---
const tableBodyGroups = $('#groups-table tbody');

function loadGroups() {
    $.get('/UP-08-Puchnin-Kriulin/controllers/api/group_api.php', function (data) {
        tableBodyGroups.empty();
        data.forEach(group => {
            const row = `
                <tr data-id="${group.group_id}">
                    <td>${group.group_id}</td>
                    <td contenteditable="true" class="edit-group-name">${group.group_name}</td>
                    <td>
                        <button class="save-btn save-btn-groups">Сохранить</button>
                        <button class="delete-btn delete-btn-groups">Удалить</button>
                    </td>
                </tr>`;
            tableBodyGroups.append(row);
        });
    }, 'json');
}

if ($('#groups-table').length > 0) {
    loadGroups();
}

$('#add-group-form').on('submit', function (e) {
    e.preventDefault();
    const formData = $(this).serializeArray();
    const data = { action: 'create' };
    formData.forEach(field => data[field.name] = field.value);
    $.post('/UP-08-Puchnin-Kriulin/controllers/api/group_api.php', JSON.stringify(data), function () {
        $('#add-group-form')[0].reset();
        loadGroups();
    });
});

$(document).on('click', '.save-btn-groups', function () {
    const row = $(this).closest('tr');
    const id = row.data('id');
    const group = {
        action: 'update',
        group_id: id,
        group_name: row.find('.edit-group-name').text()
    };
    $.post('/UP-08-Puchnin-Kriulin/controllers/api/group_api.php', JSON.stringify(group), function () {
        loadGroups();
    });
});

$(document).on('click', '.delete-btn-groups', function () {
    const id = $(this).closest('tr').data('id');
    const data = { action: 'delete', group_id: id };
    $.post('/UP-08-Puchnin-Kriulin/controllers/api/group_api.php', JSON.stringify(data), function () {
        loadGroups();
    });
});
});
```

Discipline_programs.php:

```
// --- DISCIPLINE PROGRAMS ---
const tableBodyPrograms = $('#programs-table tbody');

function loadPrograms() {
    $.get('/UP-08-Puchnin-Kriulin/controllers/api/program_api.php', function (data) {
        tableBodyPrograms.empty();
        data.forEach(program => {
            const row = `
                <tr data-id="${program.program_id}">
                    <td>${program.program_id}</td>
                    <td contenteditable="true" class="edit-discipline-id">${program.discipline_id}</td>
                    <td contenteditable="true" class="edit-topic">${program.topic}</td>
                    <td contenteditable="true" class="edit-lesson-type">${program.lesson_type}</td>
                    <td contenteditable="true" class="edit-hours">${program.hours}</td>
                    <td>
                        <button class="save-btn save-btn-programs">Сохранить</button>
                        <button class="delete-btn delete-btn-programs">Удалить</button>
                    </td>
                </tr>`;
            tableBodyPrograms.append(row);
        });
    }, 'json');
}

if ($('#programs-table').length > 0) {
    loadPrograms();
}
```

```

$('#add-program-form').on('submit', function (e) {
    e.preventDefault();
    const formData = $(this).serializeArray();
    const data = { action: 'create' };
    formData.forEach(field => data[field.name] = field.value);
    $.post('/UP-08-Puchnin-Kriulin/controllers/api/program_api.php', JSON.stringify(data), function () {
        $('#add-program-form')[0].reset();
        loadPrograms();
    });
});

$(document).on('click', '.save-btn-programs', function () {
    const row = $(this).closest('tr');
    const id = row.data('id');
    const program = {
        action: 'update',
        program_id: id,
        discipline_id: row.find('.edit-discipline-id').text(),
        topic: row.find('.edit-topic').text(),
        lesson_type: row.find('.edit-lesson-type').text(),
        hours: row.find('.edit-hours').text()
    };
    $.post('/UP-08-Puchnin-Kriulin/controllers/api/program_api.php', JSON.stringify(program), function () {
        loadPrograms();
    });
});

$(document).on('click', '.delete-btn-programs', function () {
    const id = $(this).closest('tr').data('id');
    const data = { action: 'delete', program_id: id };
    $.post('/UP-08-Puchnin-Kriulin/controllers/api/program_api.php', JSON.stringify(data), function () {
        loadPrograms();
    });
});

```

Teachers.php:

```

// --- TEACHERS ---
const tableBodyTeachers = $('#teachers-table tbody');

function loadTeachers() {
    $.get('/UP-08-Puchnin-Kriulin/controllers/api/teacher_api.php', function (data) {
        tableBodyTeachers.empty();
        data.forEach(teacher => {
            const row = `
                <tr data-id="${teacher.teacher_id}">
                    <td>${teacher.teacher_id}</td>
                    <td contenteditable="true" class="edit-last-name">${teacher.last_name}</td>
                    <td contenteditable="true" class="edit-first-name">${teacher.first_name}</td>
                    <td contenteditable="true" class="edit-middle-name">${teacher.middle_name} || ''</td>
                    <td contenteditable="true" class="edit-login">${teacher.login}</td>
                    <td contenteditable="true" class="edit-password">${teacher.password}</td>
                    <td>
                        <button class="save-btn save-btn-teachers">Сохранить</button>
                        <button class="delete-btn delete-btn-teachers">Удалить</button>
                    </td>
                </tr>
            `;
            tableBodyTeachers.append(row);
        });
    }, 'json');
}

if ($('#teachers-table').length > 0) {
    loadTeachers();
}

```

```

$('#add-teacher-form').on('submit', function (e) {
    e.preventDefault();
    const formData = $(this).serializeArray();
    const data = { action: 'create' };
    formData.forEach(field => data[field.name] = field.value);
    $.post('/UP-08-Puchnin-Kriulin/controllers/api/teacher_api.php', JSON.stringify(data), function () {
        $('#add-teacher-form')[0].reset();
        loadTeachers();
    });
});

$(document).on('click', '.save-btn-teachers', function () {
    const row = $(this).closest('tr');
    const id = row.data('id');
    const teacher = {
        action: 'update',
        teacher_id: id,
        last_name: row.find('.edit-last-name').text(),
        first_name: row.find('.edit-first-name').text(),
        middle_name: row.find('.edit-middle-name').text() || null,
        login: row.find('.edit-login').text(),
        password: row.find('.edit-password').text()
    };
    $.post('/UP-08-Puchnin-Kriulin/controllers/api/teacher_api.php', JSON.stringify(teacher), function () {
        loadTeachers();
    });
});

$(document).on('click', '.delete-btn-teachers', function () {
    const id = $(this).closest('tr').data('id');
    const data = { action: 'delete', teacher_id: id };
    $.post('/UP-08-Puchnin-Kriulin/controllers/api/teacher_api.php', JSON.stringify(data), function () {
        loadTeachers();
    });
});

```

Что было сделано: реализована динамическая загрузка через **JSON API**, добавление, редактирование и удаление записей происходит без перезагрузки страницы. В результате все действия выполняются асинхронно, что обеспечивает плавный пользовательский интерфейс. Реализована корректная работа `app.js` и моделей между собой, используя реализованные контроллеры для каждого класса.