**Нов български университет**

**Информатика**

**Денис Велиев F104630**

**Виктор Мореев F105336**

**Ангел Пенчев F105024**

**Боян Антонов F104525**

**Боян Атанасов F104164**

**Документация за Front-End частта на групов проект по:**

**Практика по програмиране и по реализация на бази данни**

**CSCB634**

**„Уеб базиран електронен дневник“**

**Преподавател:**

**гл. ас. д-р Христина Костадинова**

**Описание**

Grade Center WebApp е уеб базиран електронен дневник, предназначен за въвеждане, редактиране, изтриване и визуализиране на информация относно резултати на ученици от 1-ви до 12-ти клас. Приложението е разработено с React и използва React Redux и Redux Toolkit за управление на състоянието. Фронт-енд частта е написана изцяло на JavaScript с ReactJS.

**Технологии**

1. **React:** Основната библиотека за изграждане на потребителски интерфейс.
2. **React Redux:** За управление на състоянието на приложението.
3. **Redux Toolkit:** За улесняване на работата с Redux.
4. **React Router:** За маршрутизиране в приложението.
5. **Material-UI:** За създаване на красиви и функционални потребителски интерфейси.
6. **Axios:** За HTTP заявки към бекенда.
7. **Vite:** За разработка и билдване на приложението.

**Инсталация**

Отваря се терминал и се въвеждат следните команди, една след друга:

**Клониране на repository**

git clone https://github.com/hideOnDenis/Grade-Center-WebApp-ReactJS

cd grade-center-webapp

**Инсталиране на зависимости**

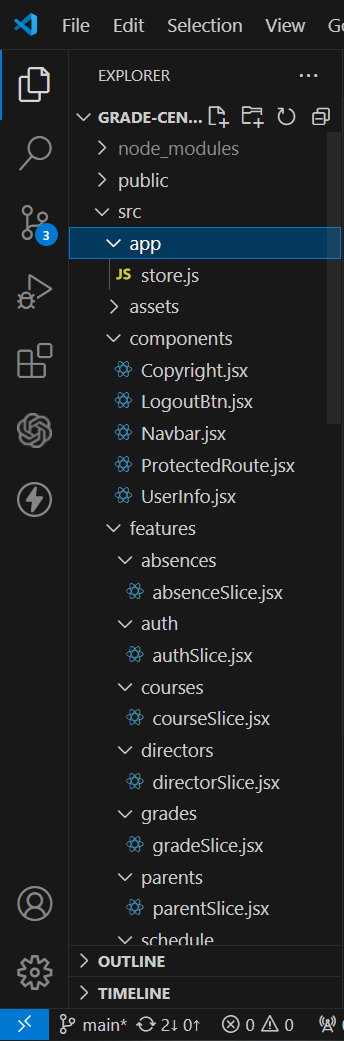
npm install

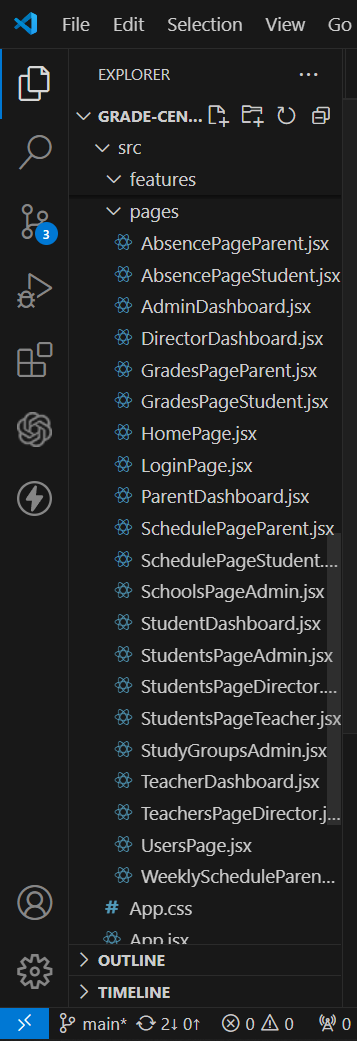
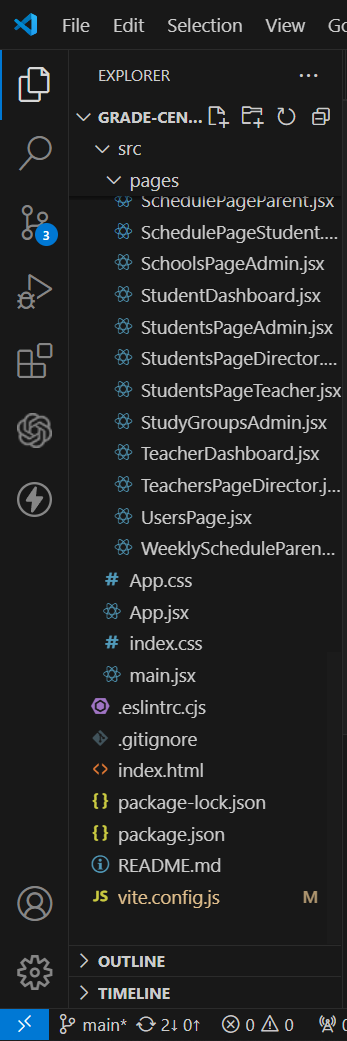
**Скриптове**

Стартиране на приложението в режим на разработка

npm run dev

Това ще стартира Vite development server и ще можете да достъпите приложението на [**http://localhost:5173**](http://localhost:5173). (Трябва да бъде стартиран на този порт, иначе няма да се върже с back-end като хората)

**Структура на проекта**



**Основни функционалности**

1. **Задаване на роли на потребителите** - система за роли като администратор, директор, учител, родител и ученик.
2. **Регистрация и вход** - функционалност за регистриране на нови потребители и вход в системата.
3. **Персонализиране на изгледа** - възможност за персонализиране на интерфейса според ролята на потребителя.
4. **Данни за училище** - въвеждане на информация като име и адрес на училището.
5. **Записване на ученици** - възможност за записване и отписване на ученици в определен клас в училище.
6. **Управление на данни за директор** - въвеждане, редактиране и изтриване на лични данни за директор на училище.
7. **Управление на данни за преподаватели** - въвеждане, редактиране и изтриване на лични данни за преподаватели и предметите, които преподават.
8. **Управление на данни за ученици** - въвеждане, редактиране и изтриване на лични данни за ученици и класа, в който са записани.
9. **Управление на данни за родители** - въвеждане, редактиране и изтриване на лични данни за родители и свързаните с тях ученици.
10. **Оценки и отсъствия** - учителите могат да въвеждат, редактират и изтриват оценки и отсъствия за учениците, на които преподават.
11. **Достъп за родители** - родителите могат да виждат само оценките и отсъствията на собствените си деца.
12. **Достъп за директор** - директорът вижда цялата информация за предмети, учители, ученици и родители, както и статистика за разпределението на оценките и отсъствията.
13. **Администраторски функции** - администраторът управлява данните за потребителите и техните роли, редактира цялата информация в системата и има достъп до статистическа информация за оценките и отсъствията във всички училища и предмети.

**Маршрутизиране**

### Основен контейнер за маршрутизация

<Router>

<Routes>

...

</Routes>

</Router>

Това създава контейнер за всички дефинирани маршрути в приложението, използвайки react-router-dom.

### Публични маршрути

<Route path="/" element={<HomePage />} />

<Route path="/login" element={<LoginPage />} />

* "/" - Зарежда началната страница (HomePage).
* "/login" - Зарежда страницата за вход (LoginPage).

### Защитени маршрути

#### **Администраторски маршрути**

<Route element={<ProtectedRoute allowedRoles={["admin"]} />}>

<Route path="/admin/users" element={<UsersPage />} />

<Route path="/admin/dashboard" element={<AdminDashboard />} />

<Route path="/admin/schools" element={<SchoolsPageAdmin />} />

<Route path="/admin/students" element={<StudentsPageAdmin />} />

<Route path="/admin/studygroups" element={<StudyGroupsAdmin />} />

</Route>

Тези маршрути са достъпни само за потребители с роля "admin". ProtectedRoute компонентът проверява ролята на потребителя преди да позволи достъп.

* "/admin/users" - Зарежда страницата за управление на потребители (UsersPage).
* "/admin/dashboard" - Зарежда административното табло (AdminDashboard).
* "/admin/schools" - Зарежда страницата за управление на училища (SchoolsPageAdmin).
* "/admin/students" - Зарежда страницата за управление на ученици (StudentsPageAdmin).
* "/admin/studygroups" - Зарежда страницата за управление на учебни групи (StudyGroupsAdmin).

#### **Директорски маршрути**

<Route element={<ProtectedRoute allowedRoles={["director"]} />}>

<Route path="/director/dashboard" element={<DirectorDashboard />} />

<Route path="/director/teachers" element={<TeachersPage />} />

<Route path="/director/students" element={<StudentsPageDirector />} />

</Route>

Тези маршрути са достъпни само за потребители с роля "director".

* "/director/dashboard" - Зарежда таблото на директора (DirectorDashboard).
* "/director/teachers" - Зарежда страницата за управление на учители (TeachersPage).
* "/director/students" - Зарежда страницата за управление на ученици (StudentsPageDirector).

#### **Учителски маршрути**

<Route element={<ProtectedRoute allowedRoles={["teacher"]} />}>

<Route path="/teacher/dashboard" element={<TeacherDashboard />} />

<Route path="/teacher/students" element={<StudentsPageTeacher />} />

</Route>

Тези маршрути са достъпни само за потребители с роля "teacher".

* "/teacher/dashboard" - Зарежда таблото на учителя (TeacherDashboard).
* "/teacher/students" - Зарежда страницата за управление на ученици (StudentsPageTeacher).

#### **Родителски маршрути**

<Route element={<ProtectedRoute allowedRoles={["parent"]} />}>

<Route path="/parent/dashboard" element={<ParentDashboard />} />

<Route path="/parent/schedule" element={<SchedulePageParent />} />

<Route path="/parent/:studentId/absences" element={<AbsencePageParent />} />

<Route path="/parent/:studentId/grades" element={<GradesPageParent />} />

<Route path="/parent/:studentId/schedule" element={<WeeklyScheduleParent />} />

</Route>

Тези маршрути са достъпни само за потребители с роля "parent".

* "/parent/dashboard" - Зарежда таблото на родителя (ParentDashboard).
* "/parent/schedule" - Зарежда страницата с разписание (SchedulePageParent).
* "/parent/:studentId/absences" - Зарежда страницата с отсъствия на конкретен ученик (AbsencePageParent).
* "/parent/:studentId/grades" - Зарежда страницата с оценки на конкретен ученик (GradesPageParent).
* "/parent/:studentId/schedule" - Зарежда седмичното разписание на конкретен ученик (WeeklyScheduleParent).

#### **Ученически маршрути**

<Route element={<ProtectedRoute allowedRoles={["student"]} />}>

<Route path="/student/dashboard" element={<StudentDashboard />} />

<Route path="/student/schedule" element={<SchedulePageStudent />} />

<Route path="/student/absences" element={<AbsencePageStudent />} />

<Route path="/student/grades" element={<GradesPageStudent />} />

</Route>

Тези маршрути са достъпни само за потребители с роля "student".

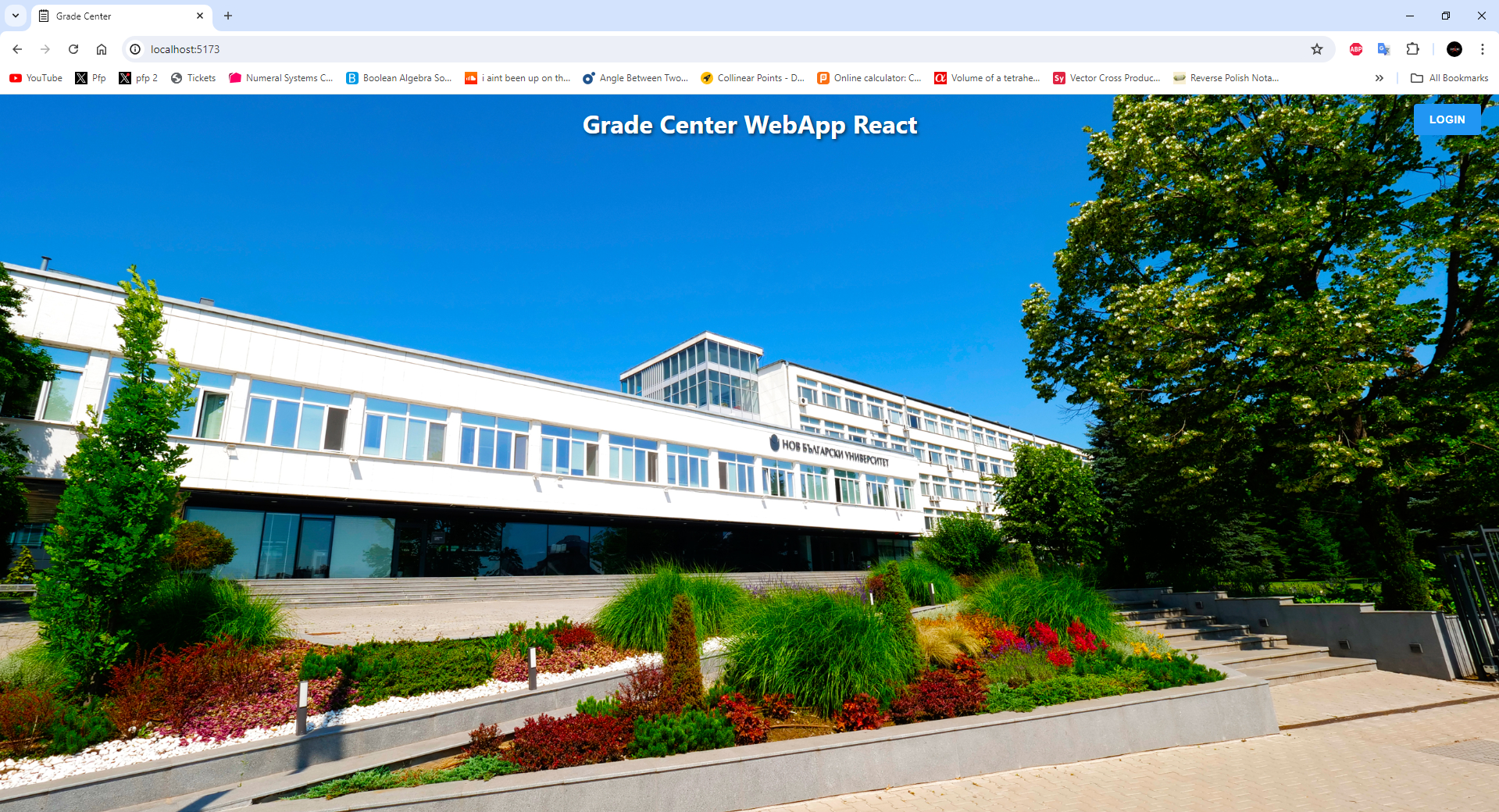
* "/student/dashboard" - Зарежда таблото на ученика (StudentDashboard).
* "/student/schedule" - Зарежда страницата с разписание (SchedulePageStudent).
* "/student/absences" - Зарежда страницата с отсъствия (AbsencePageStudent).
* "/student/grades" - Зарежда страницата с оценки (GradesPageStudent).

### Обобщение

Маршрутизирането в приложението е организирано на база роли на потребителите, като осигурява достъп до специфични страници само за оторизирани потребители. Това се осъществява чрез компонента ProtectedRoute, който проверява дали потребителят има необходимата роля, преди да предостави достъп до съответния маршрут.

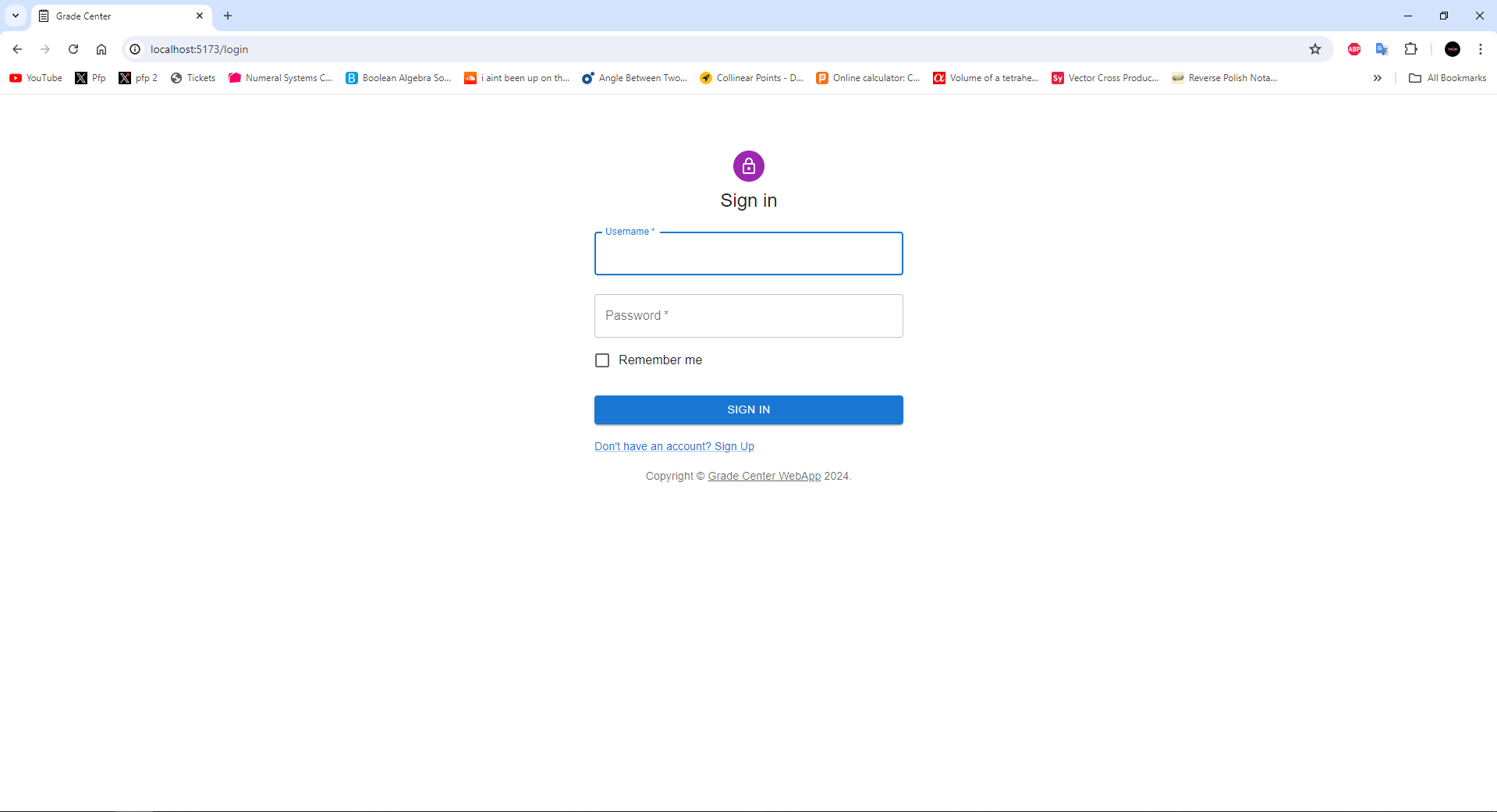
**Екранни снимки (Скрийншоти)**

**Главна страница**

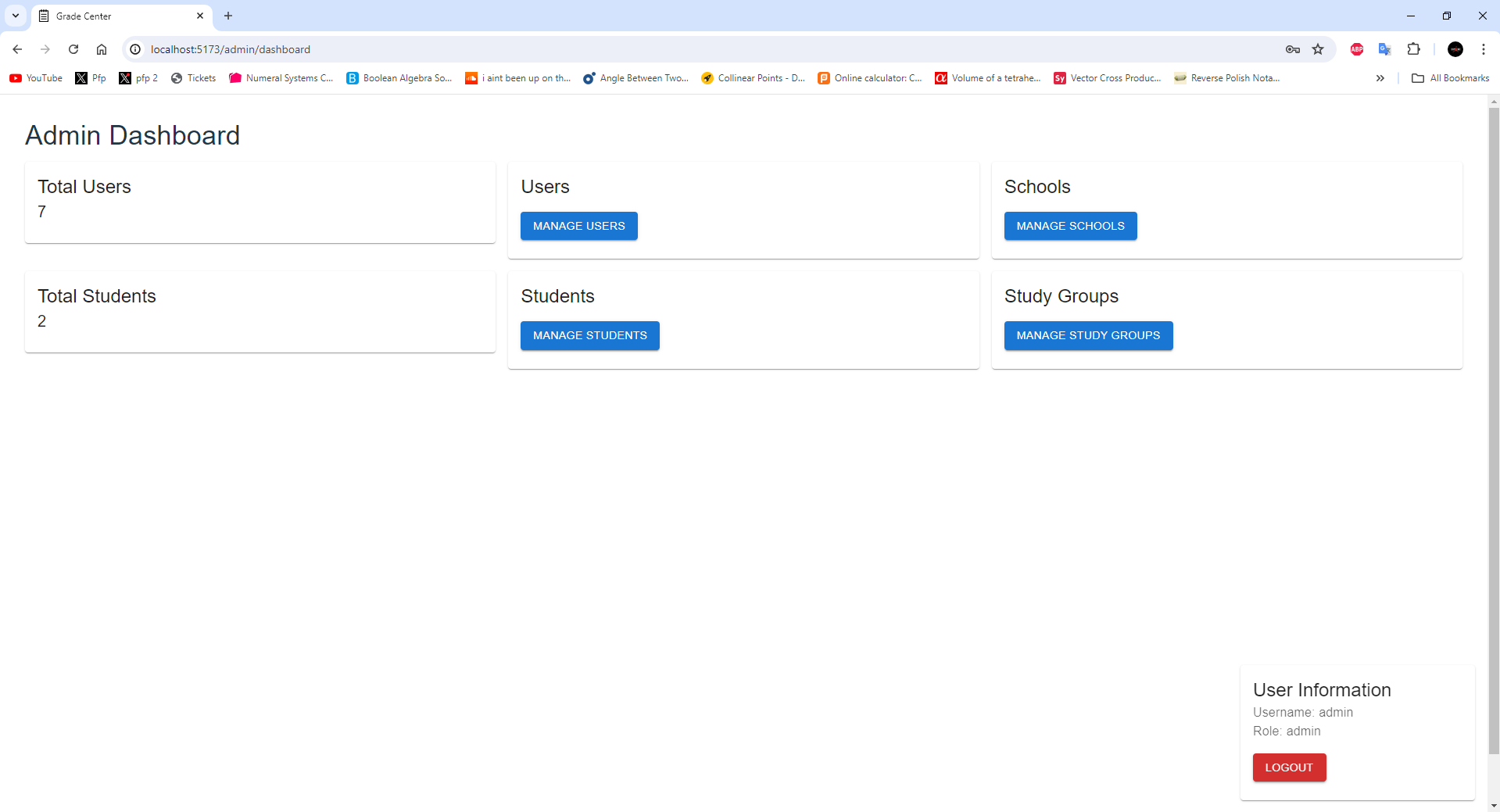


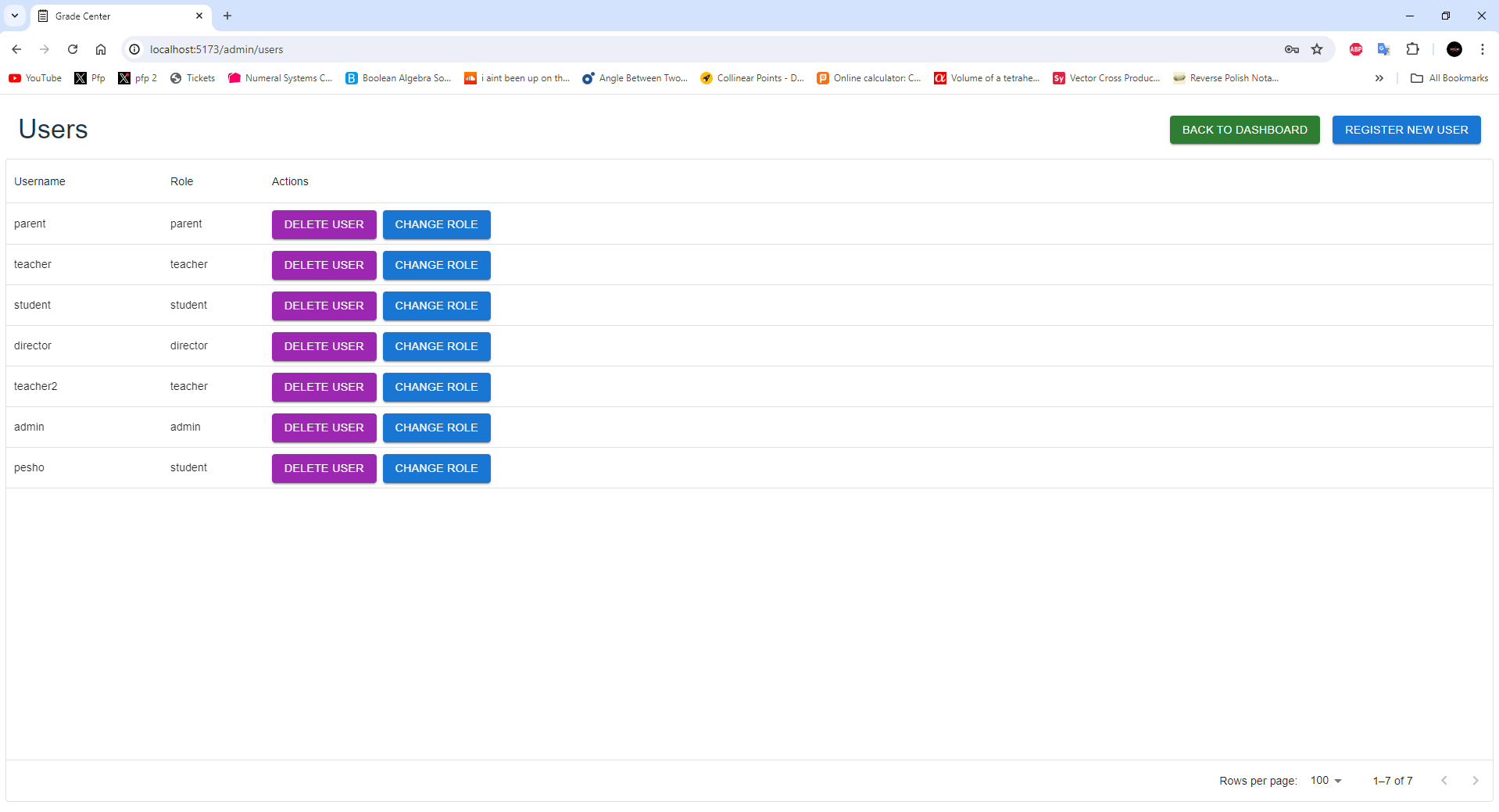
**Страница за вход (Login Page)**

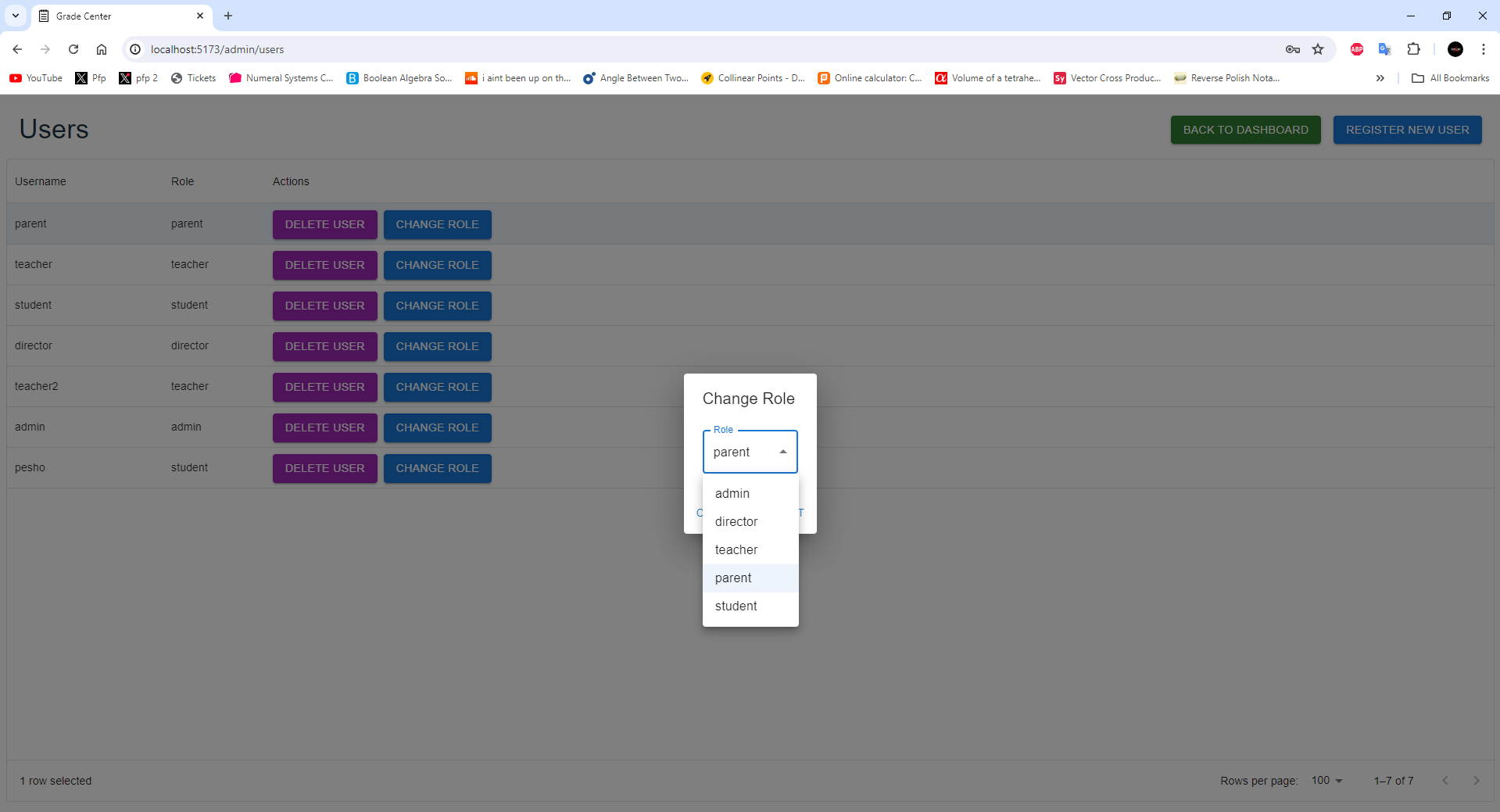
От тук, потребителят влиза в системата, като въведе име и парола.

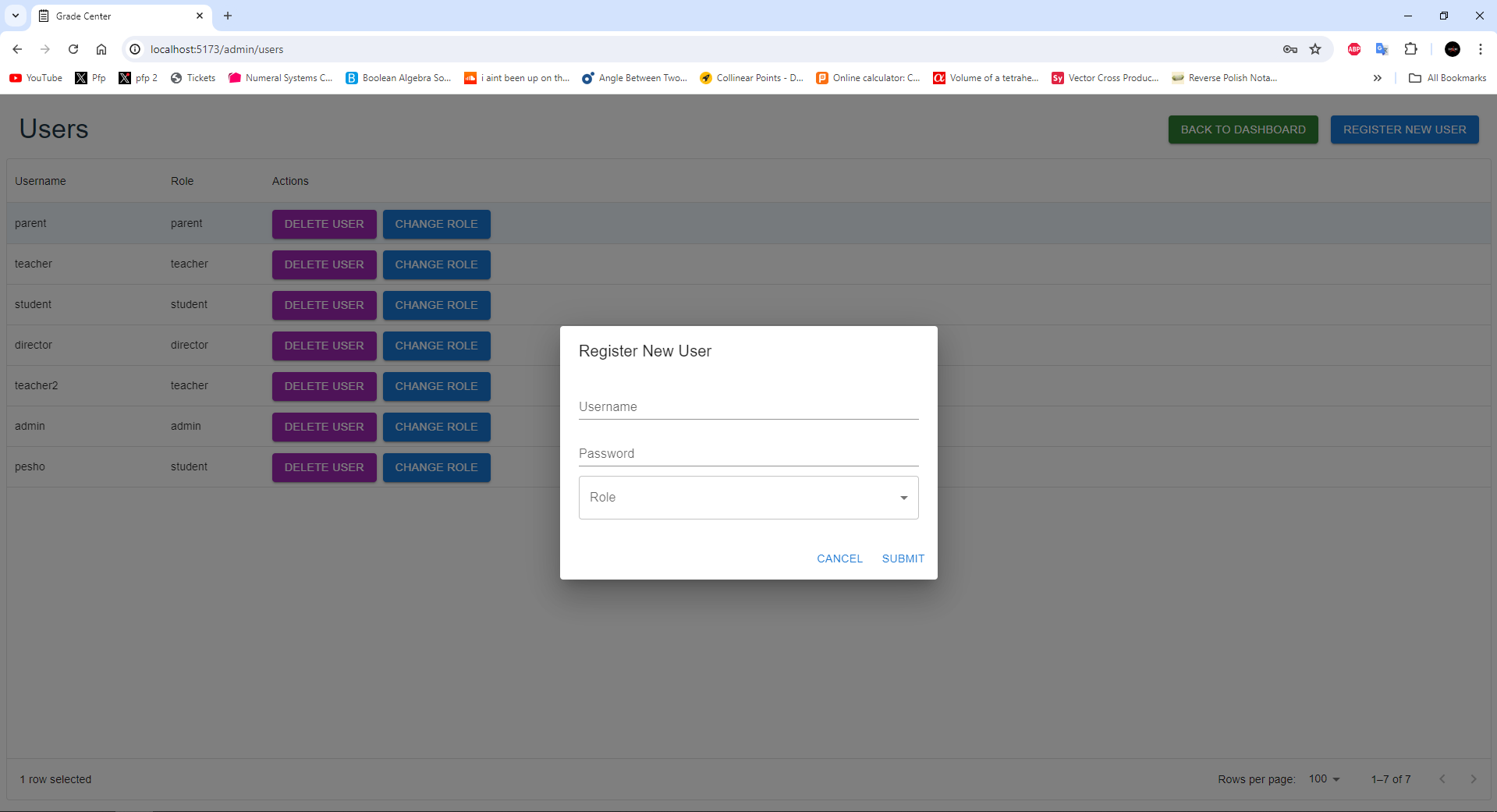


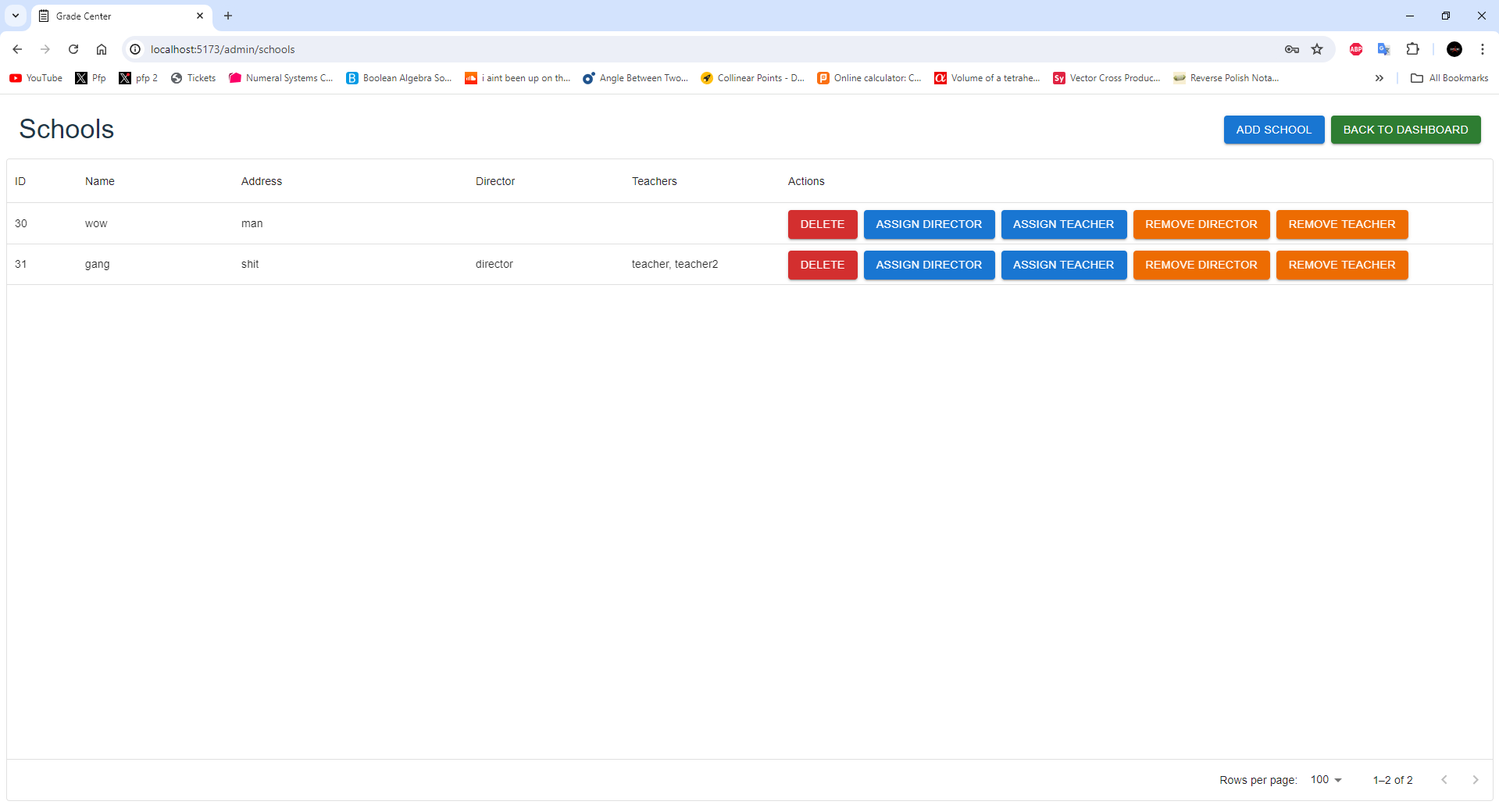
**Всички следващи страници са интерфейс за роля хора с роля администратор:**

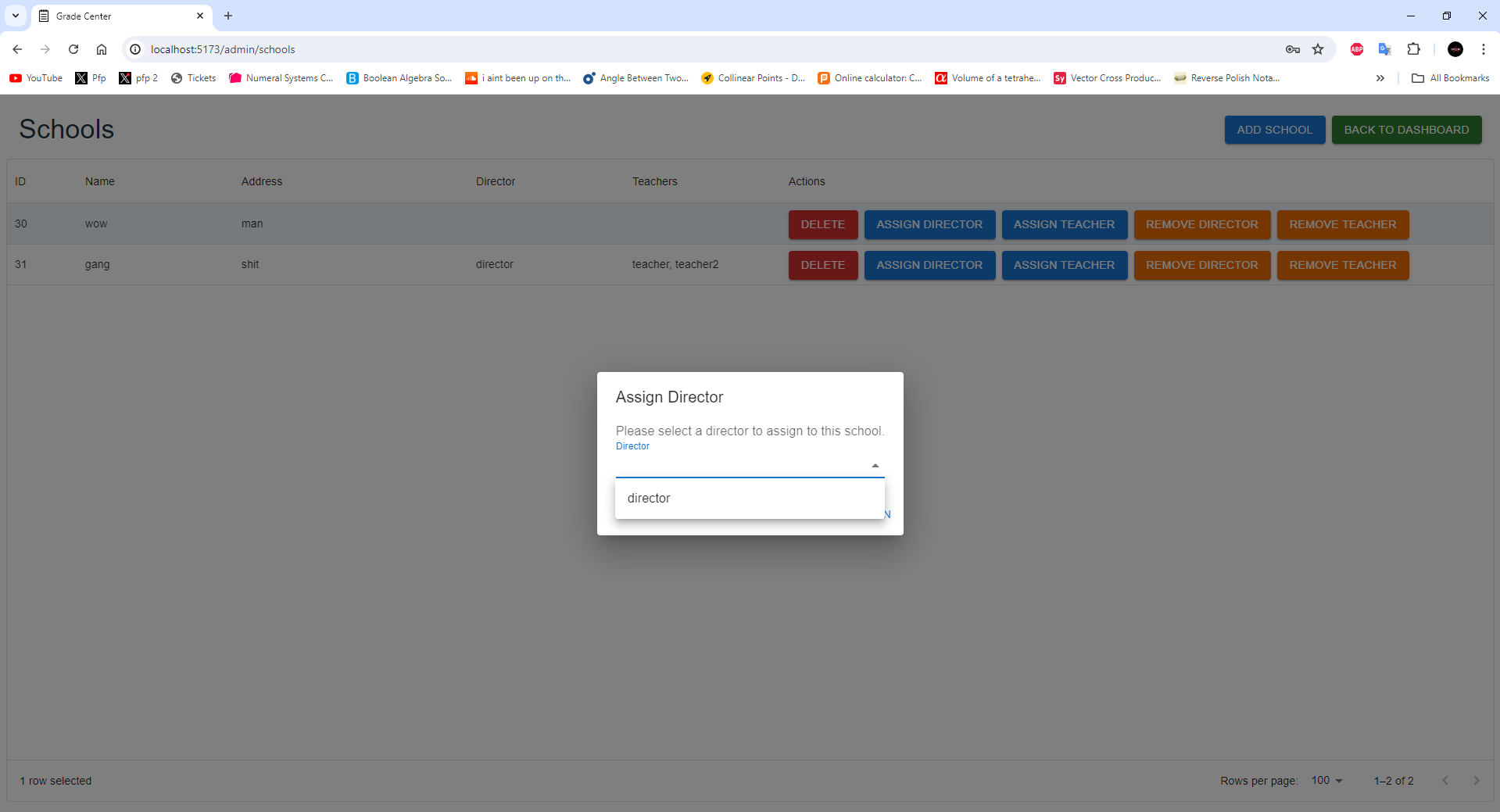


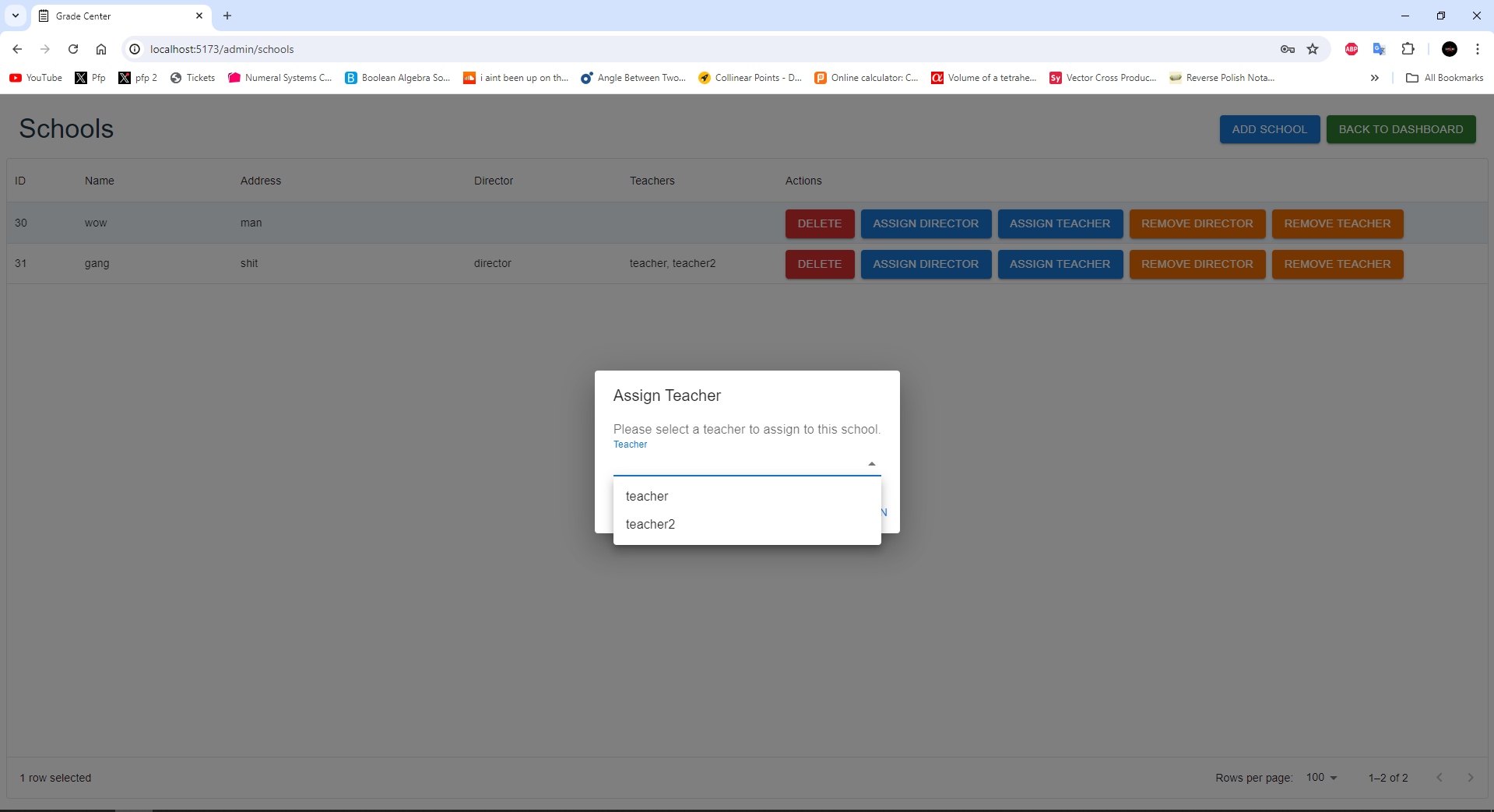


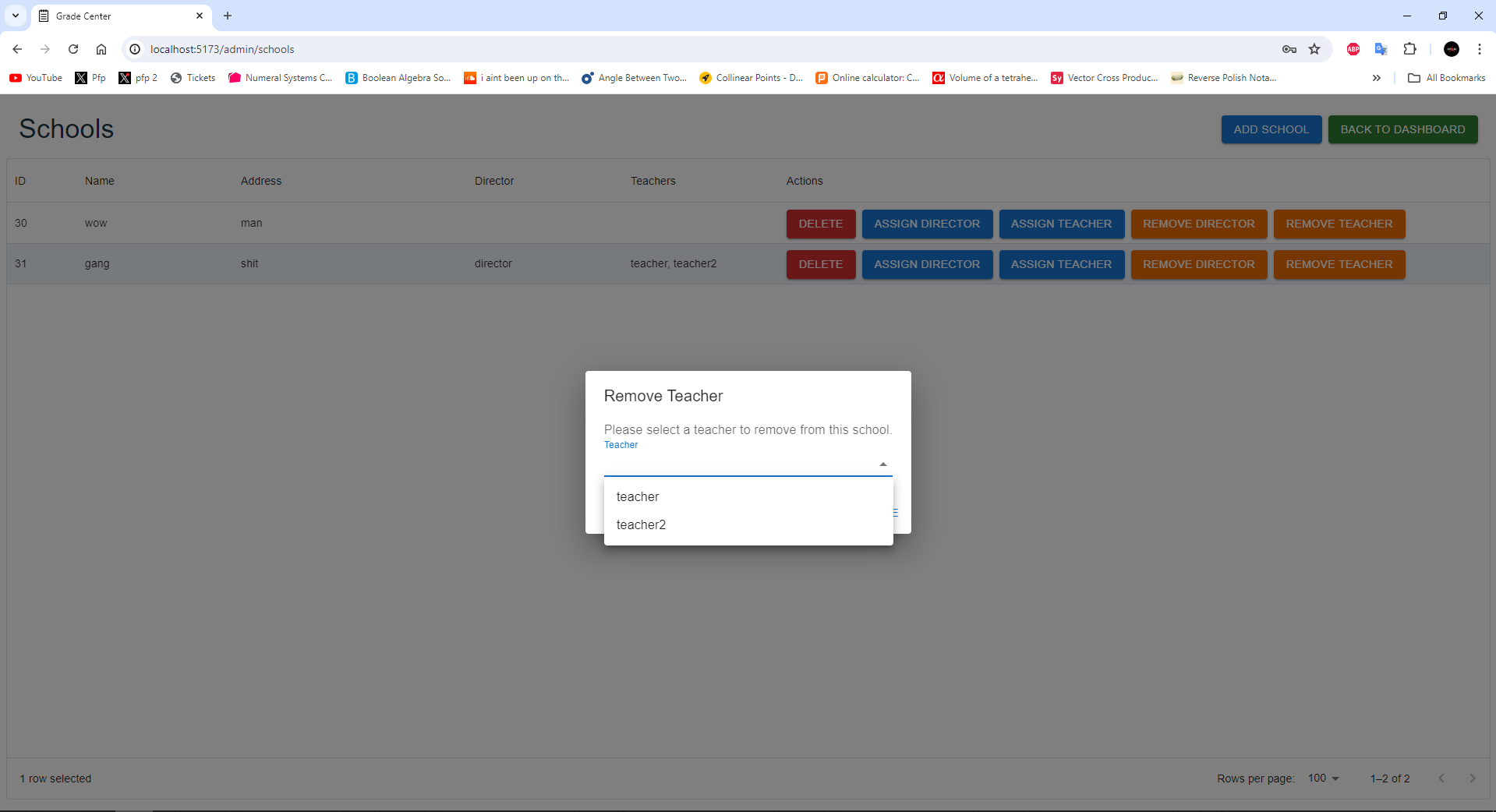


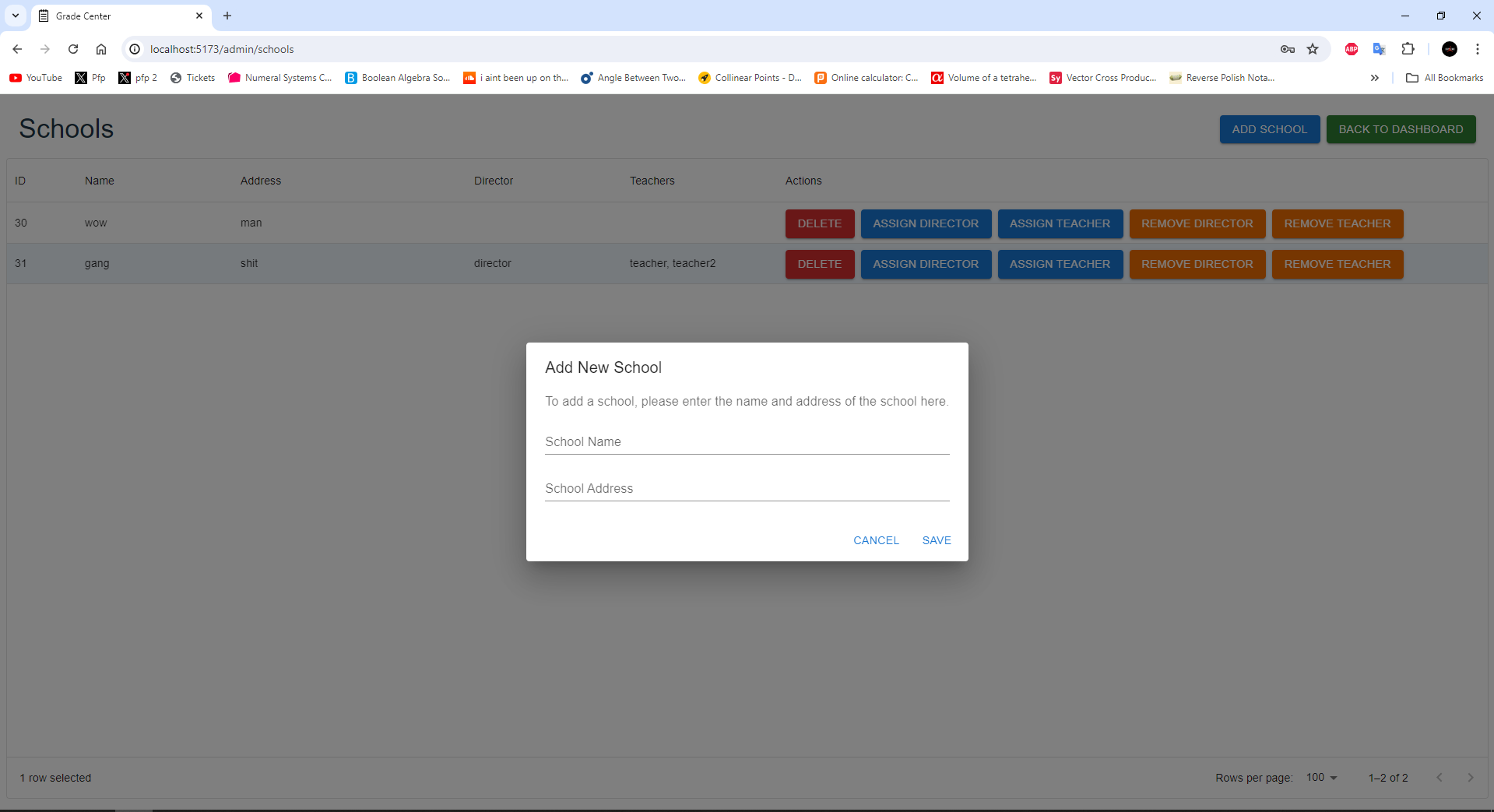




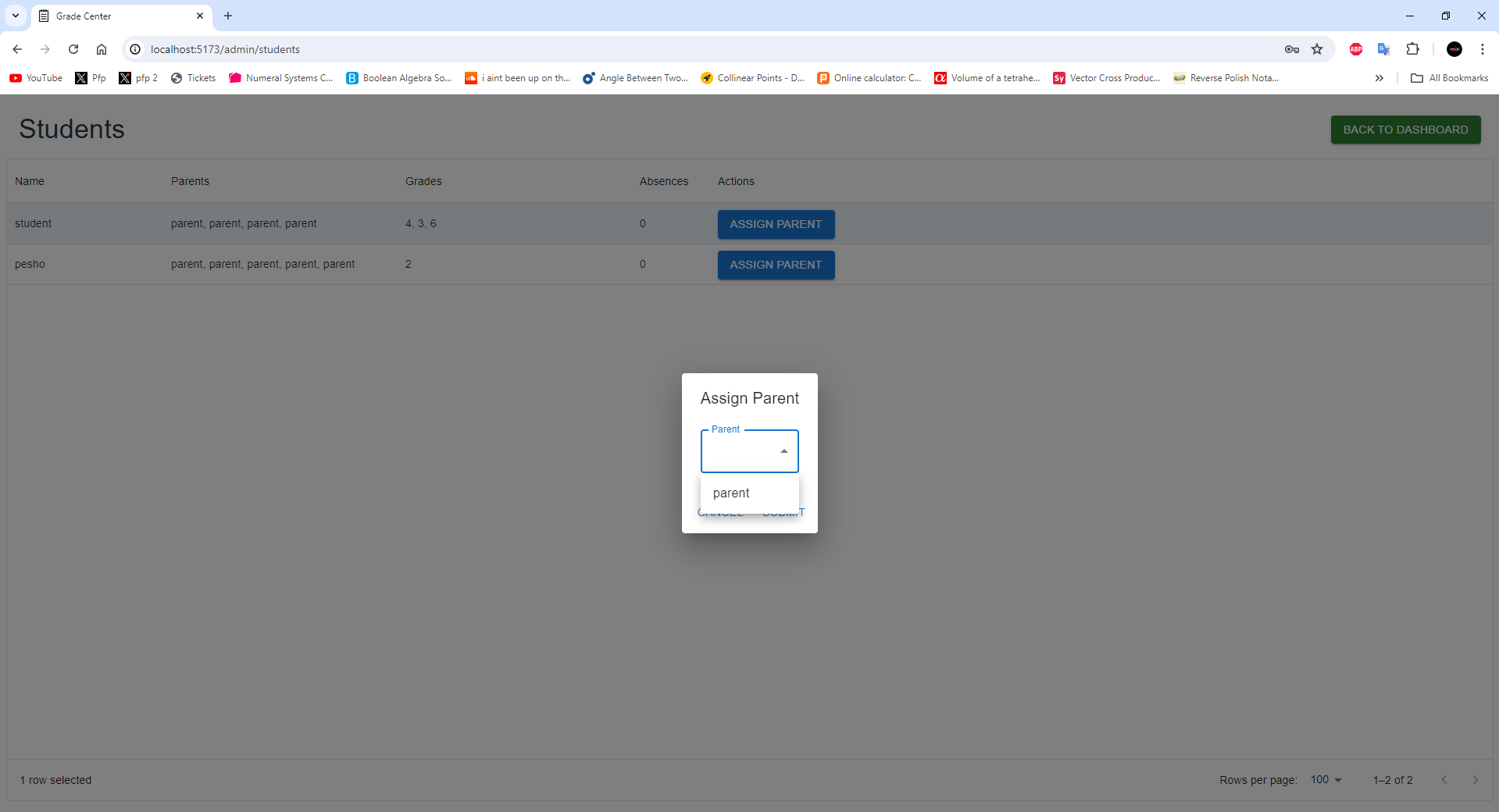


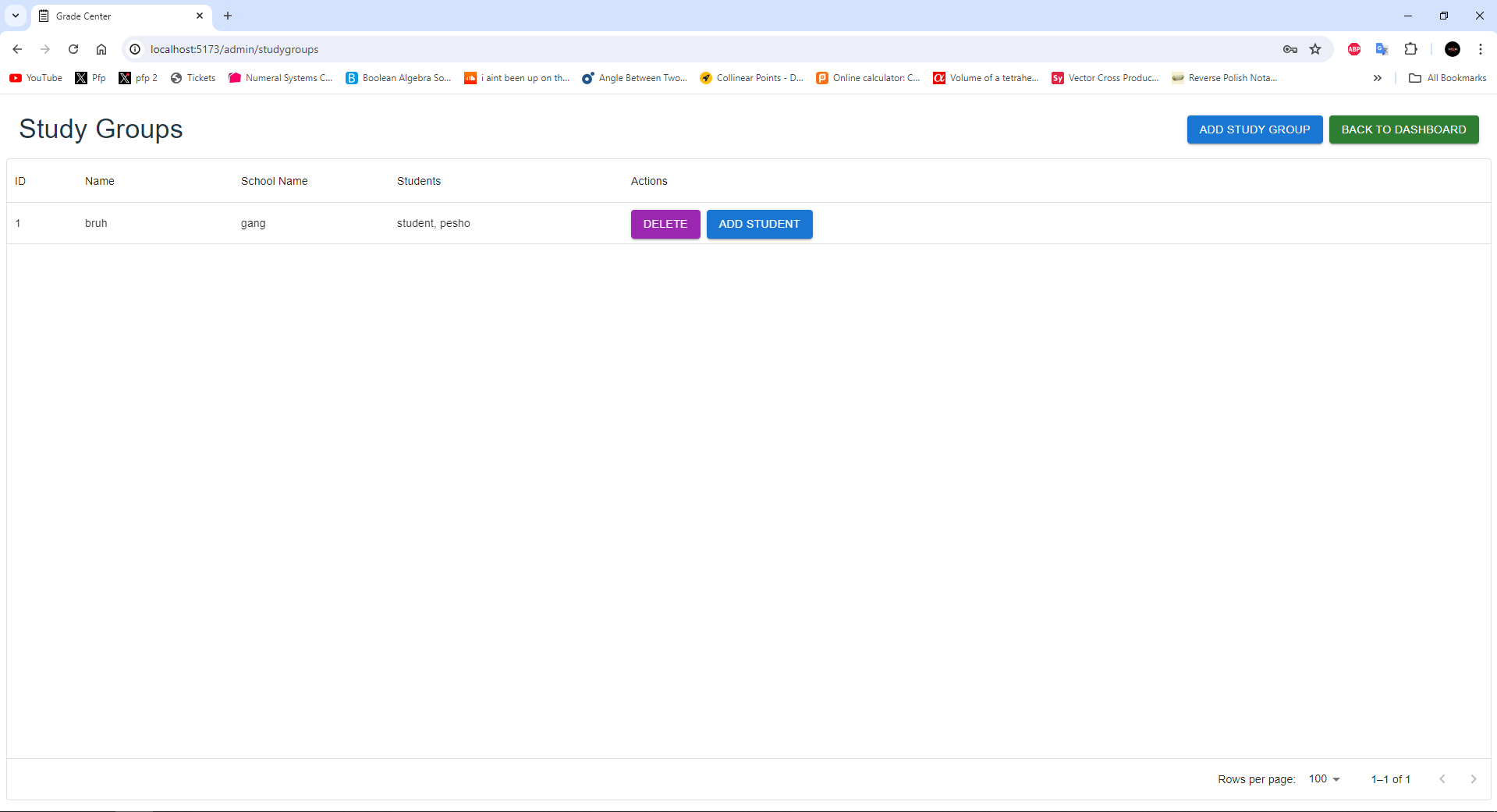




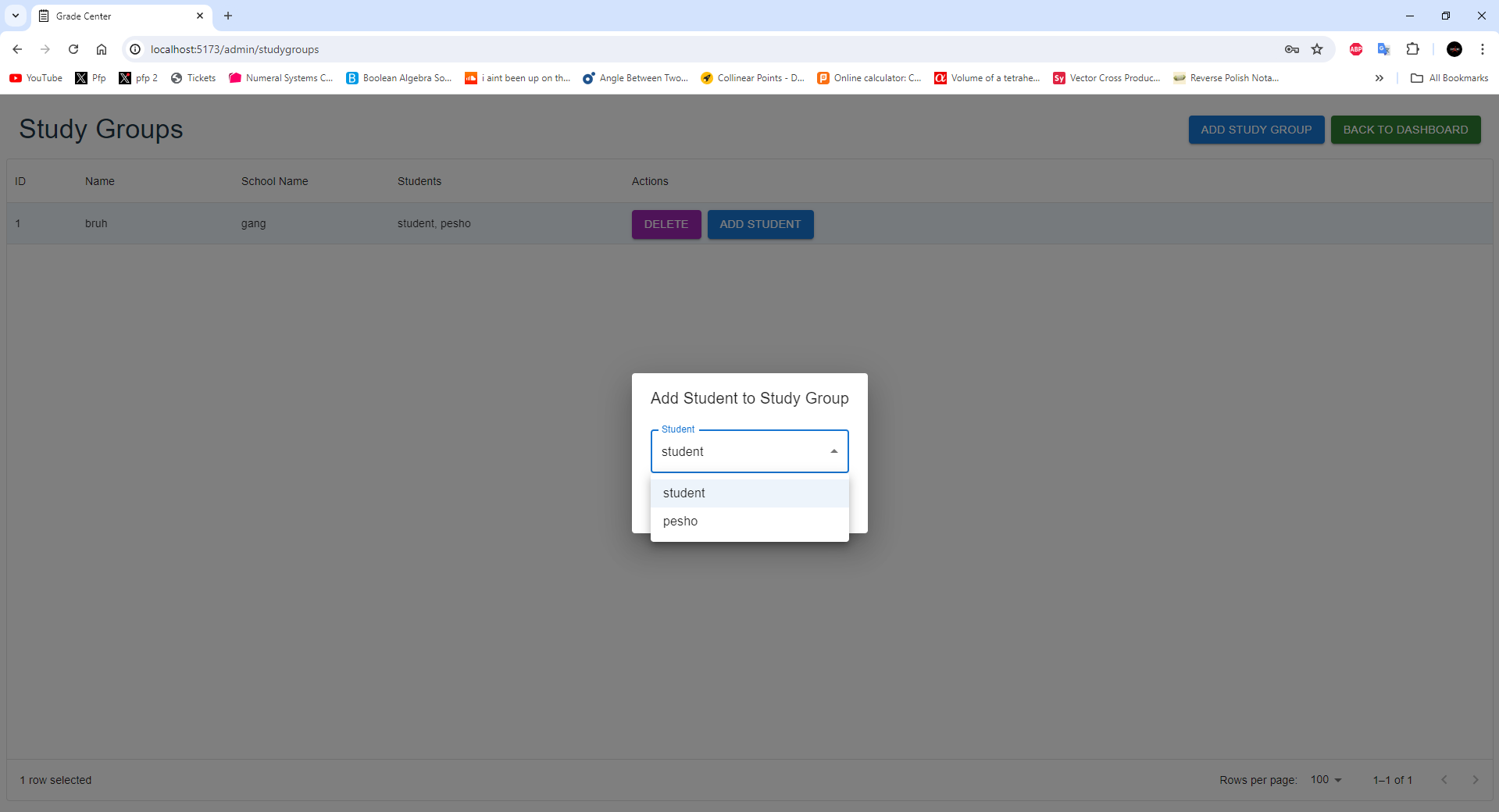


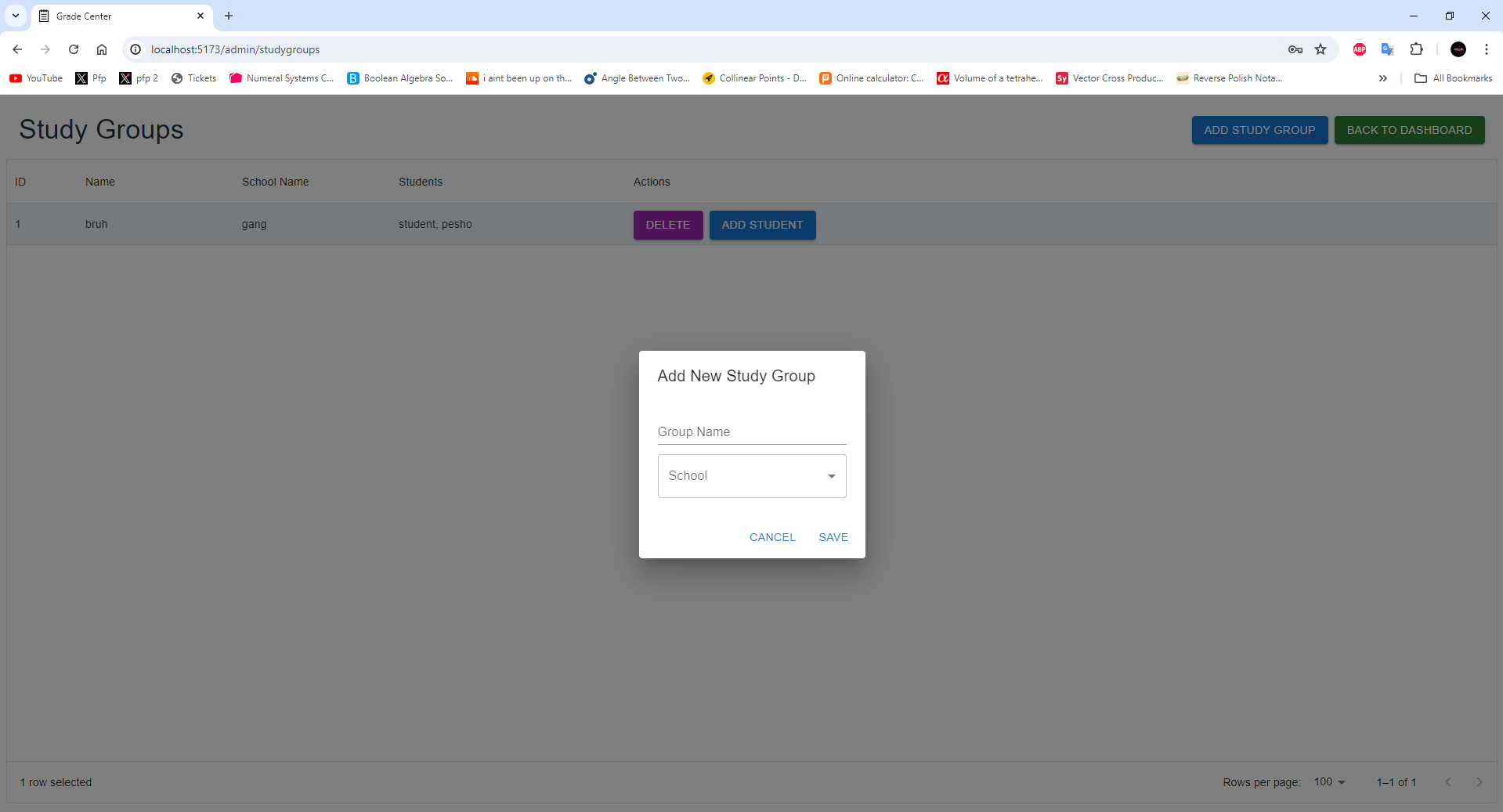




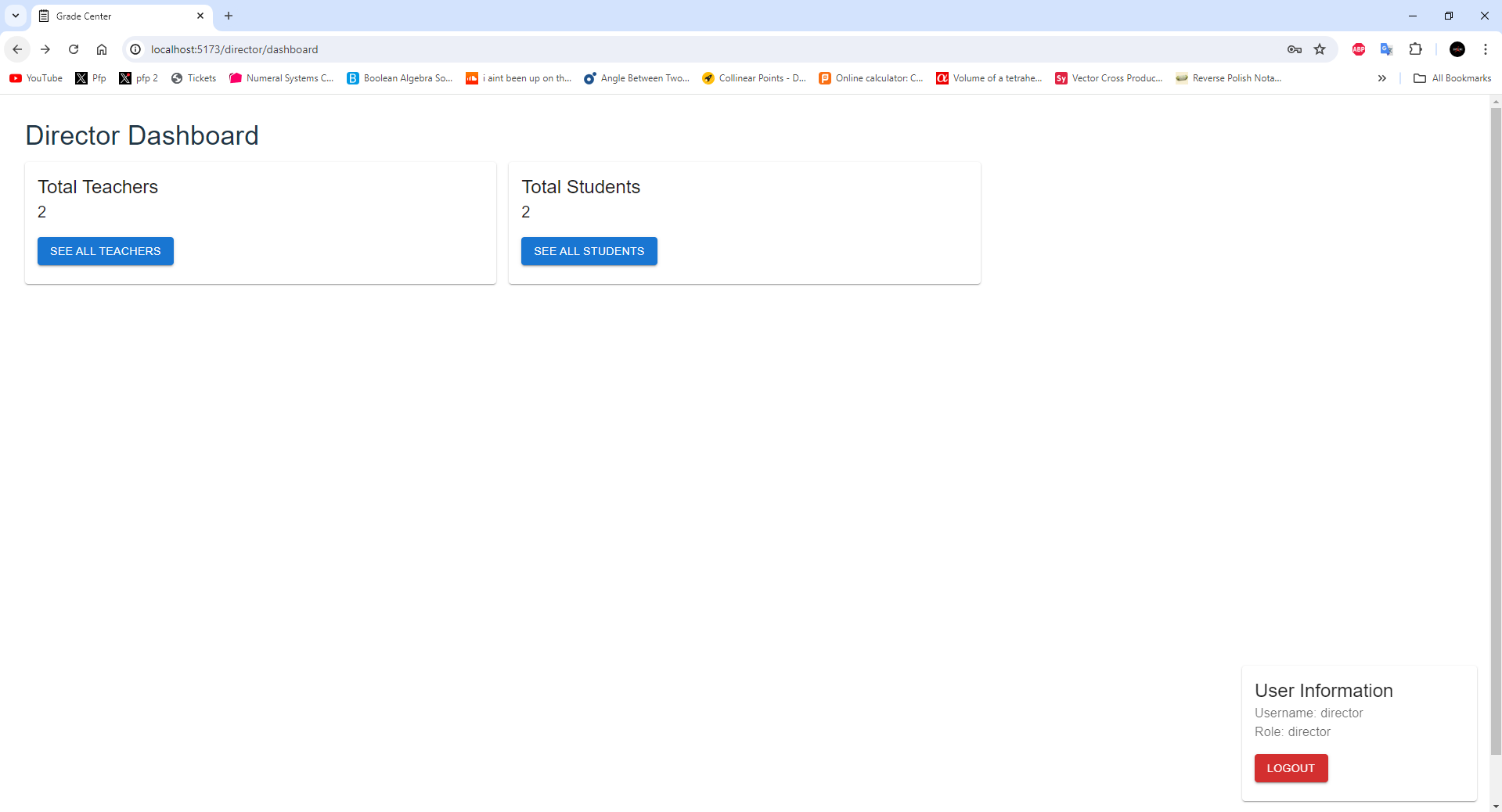


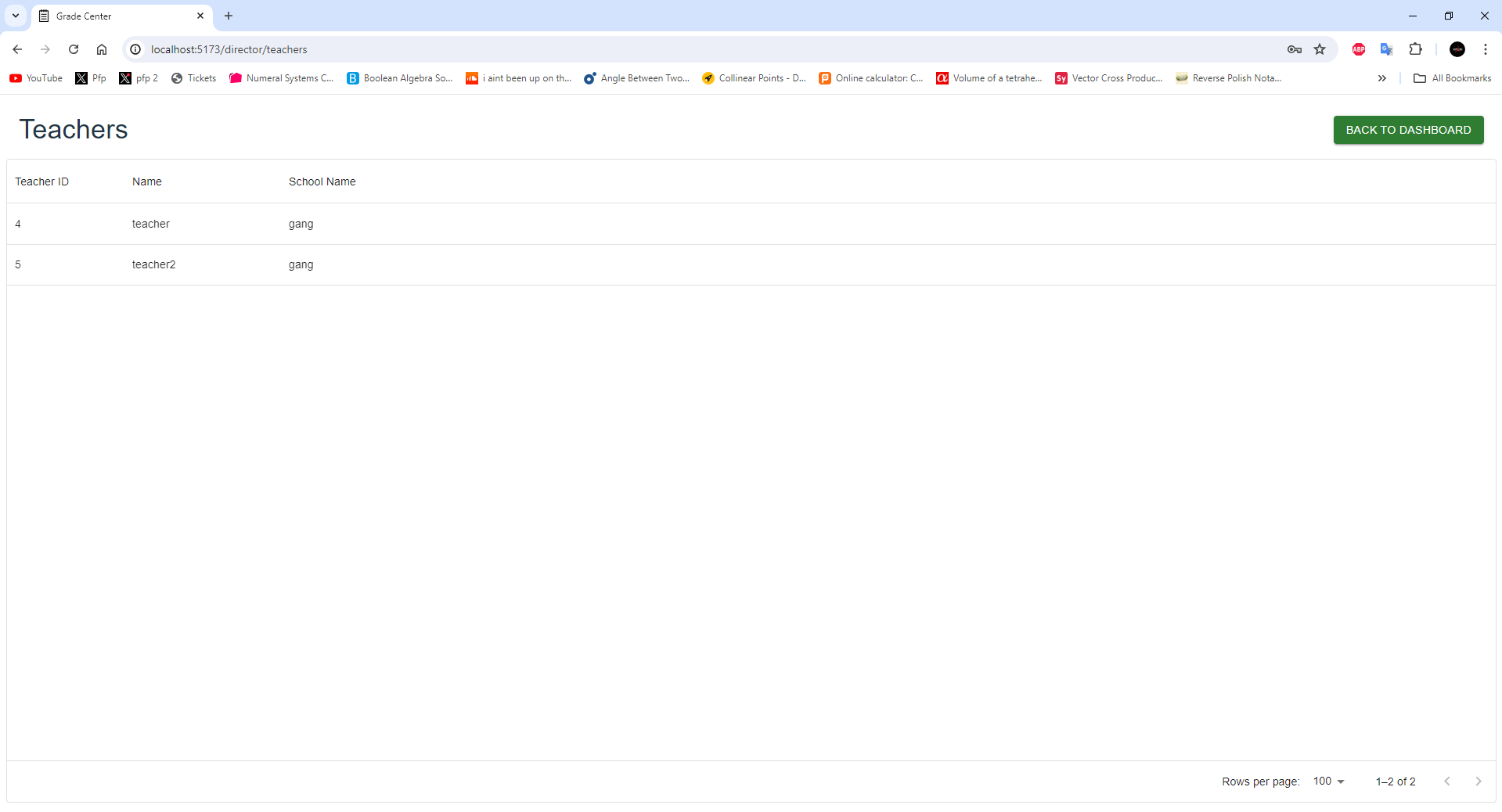
От тук могат да се добавят вече съществуващите ученици от базата към study group:

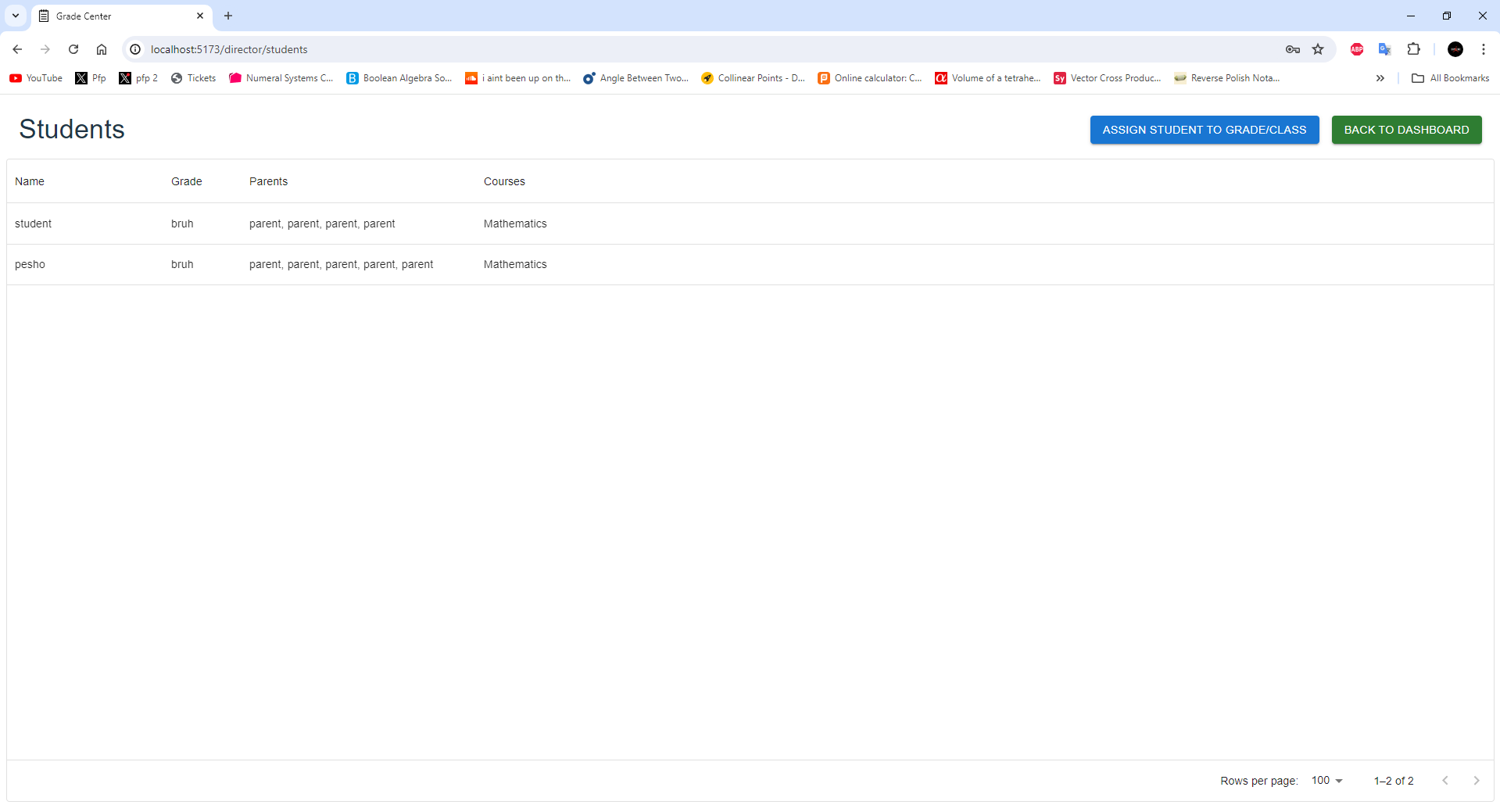




**Всички следващи страници са интерфейс за потребители с роля директор:**

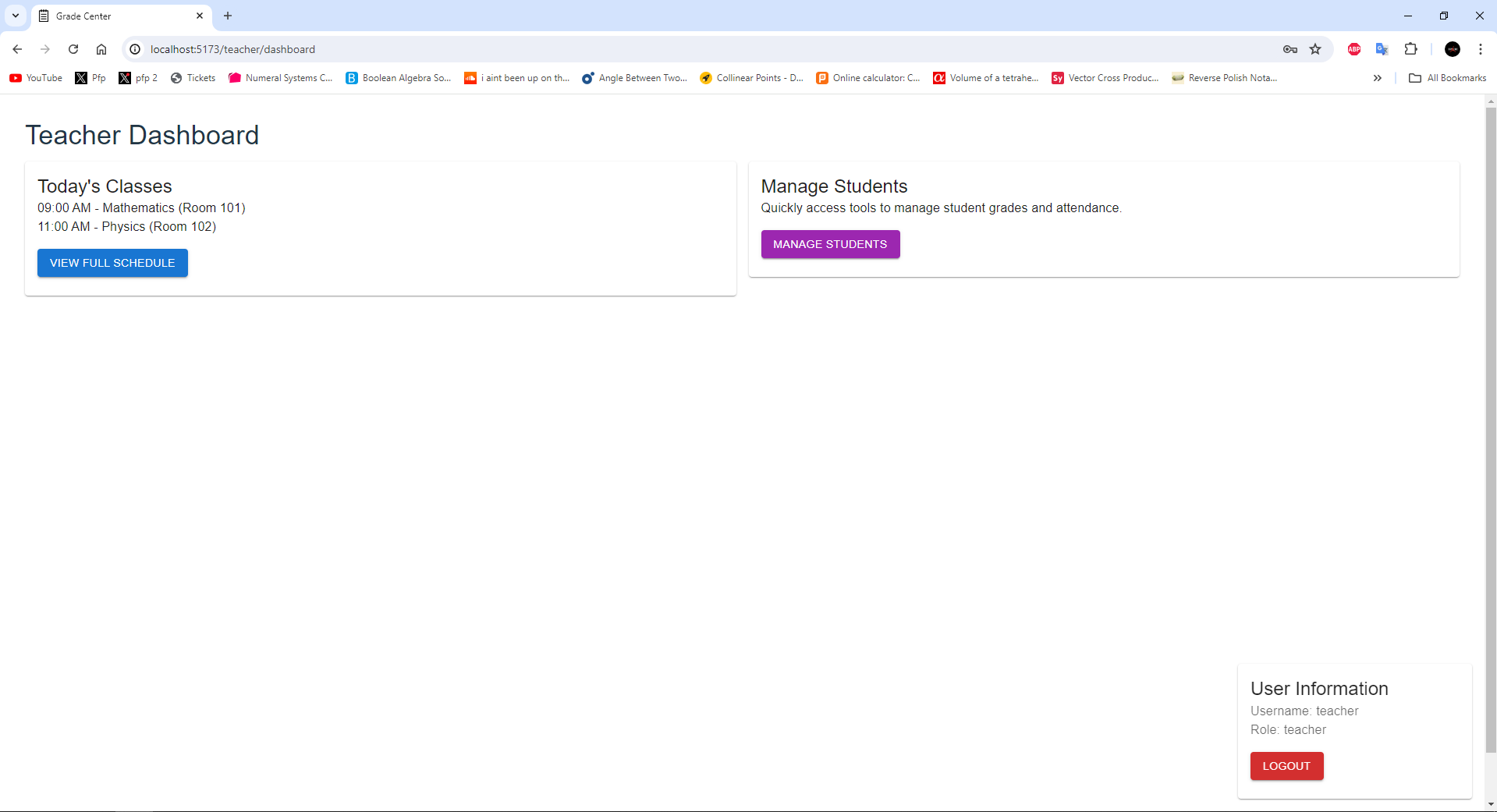


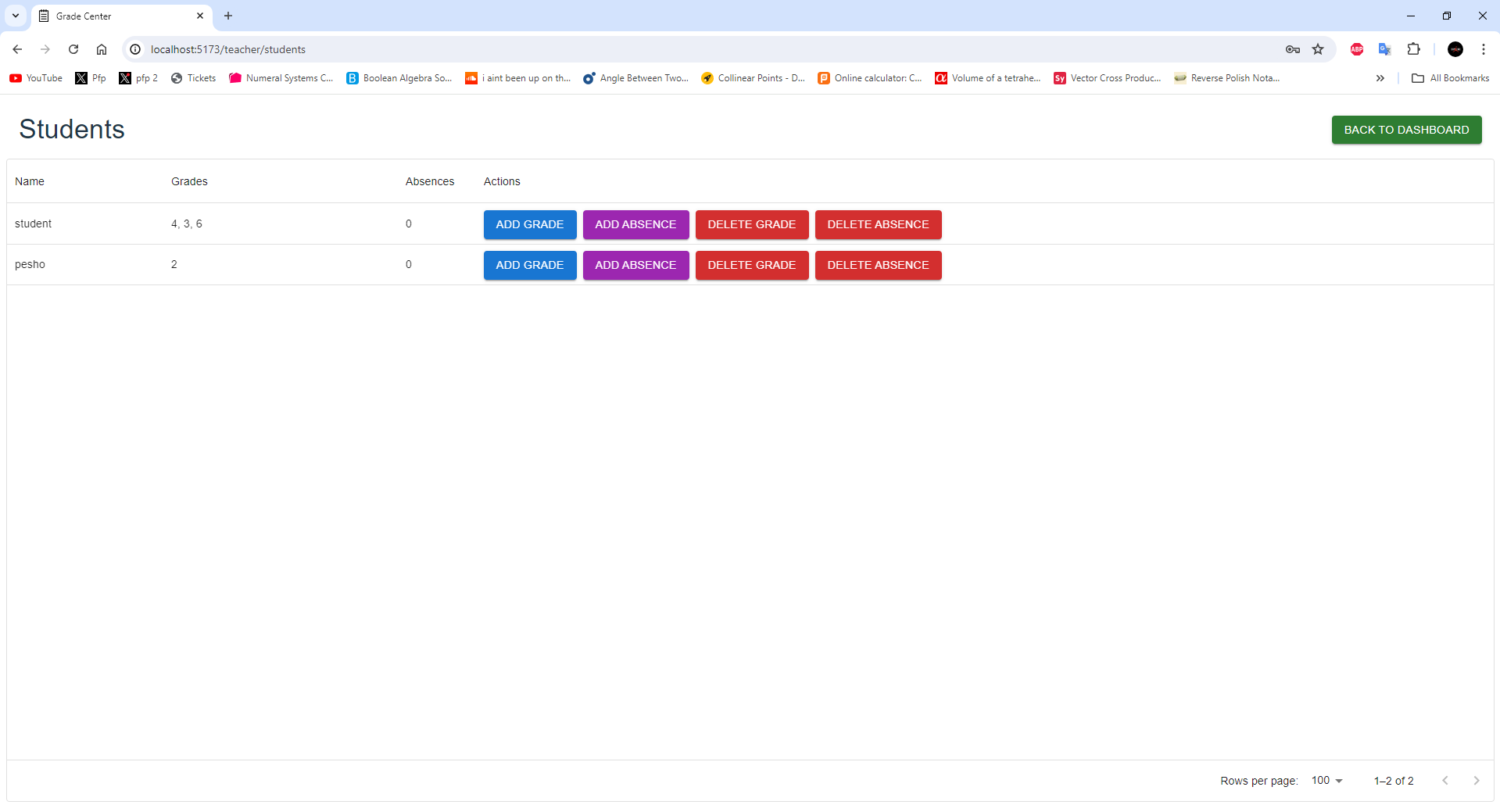


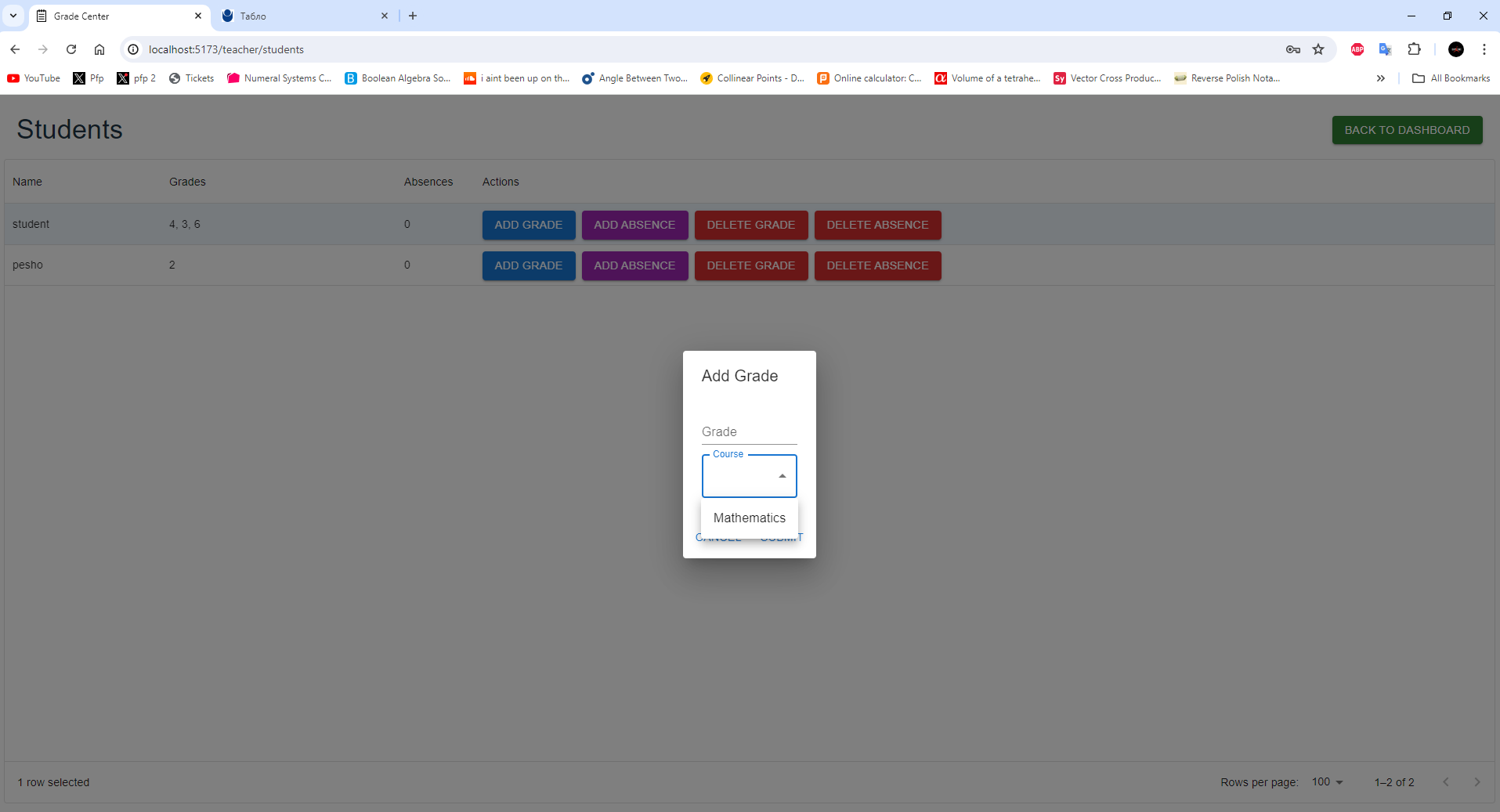


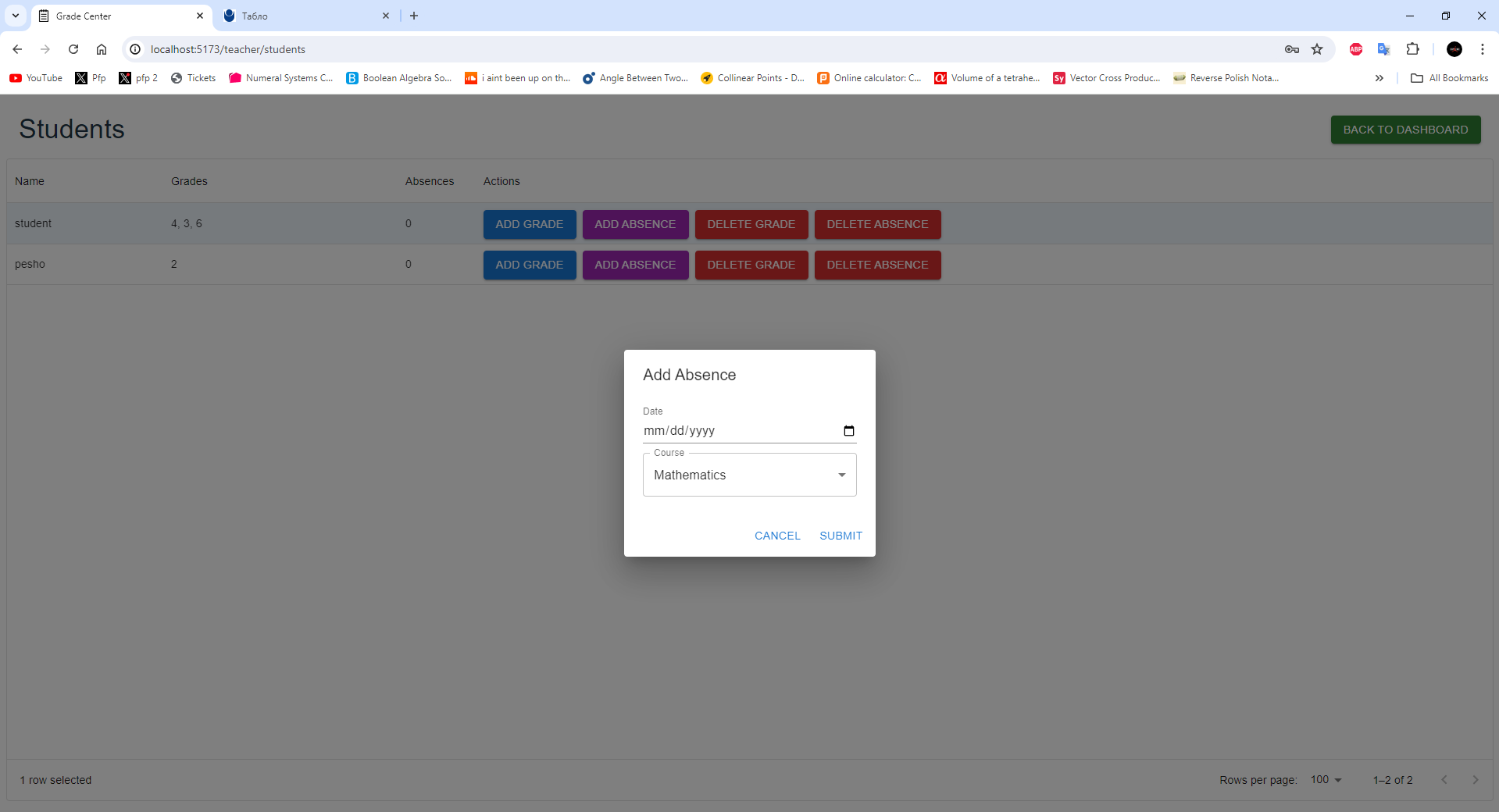


**Всички следващи страници са за потребители с ролята учител:**

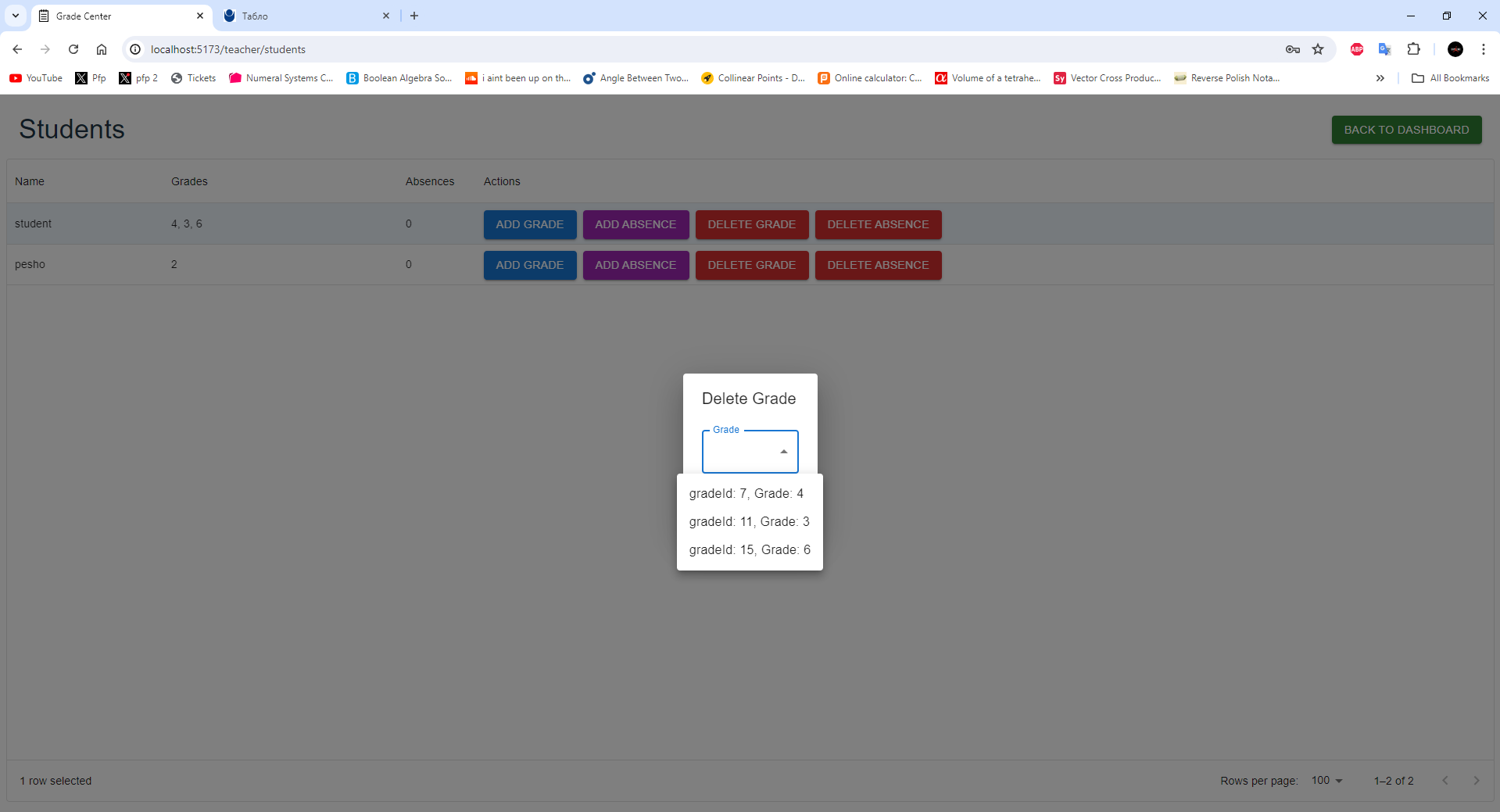


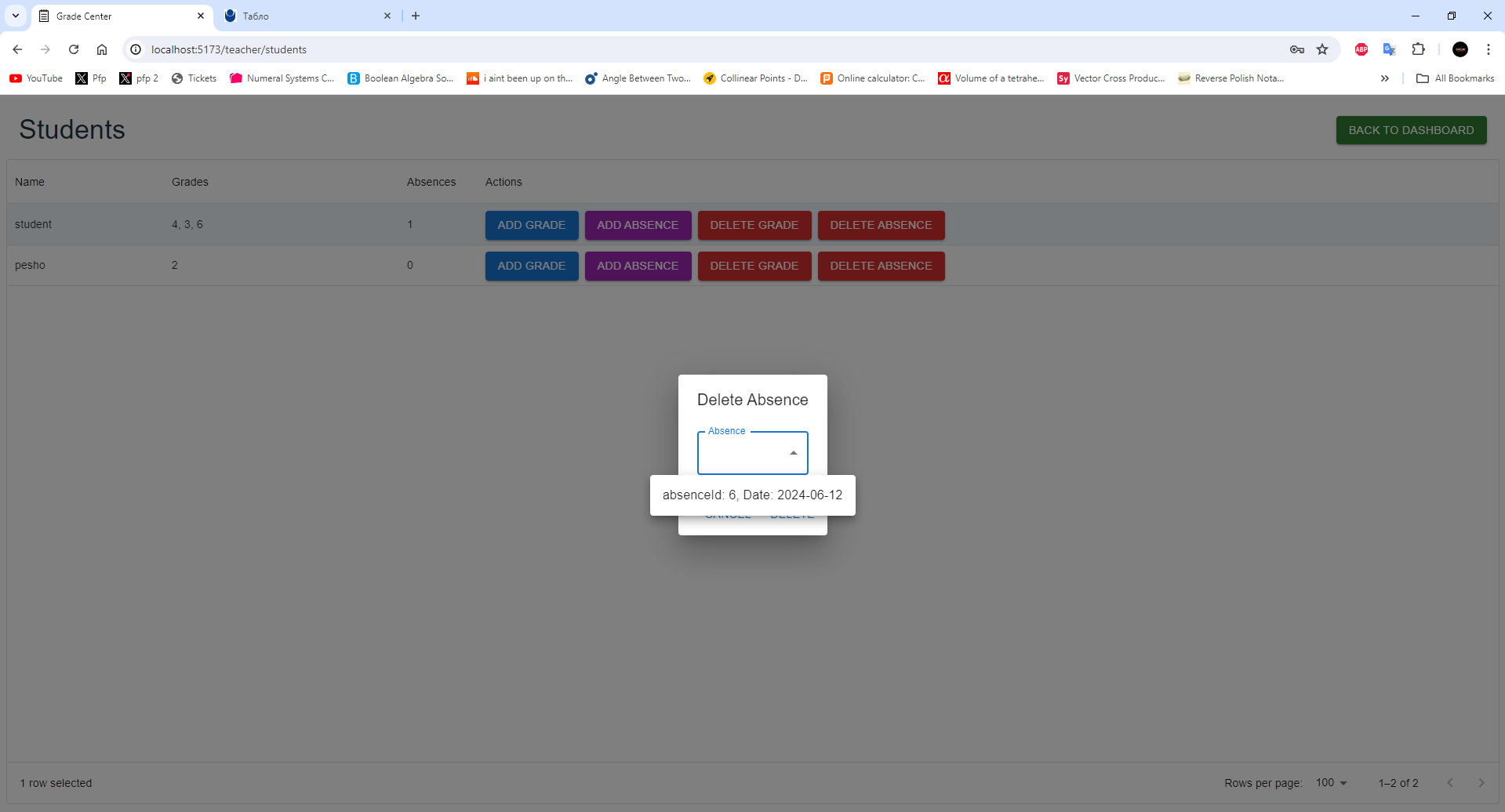




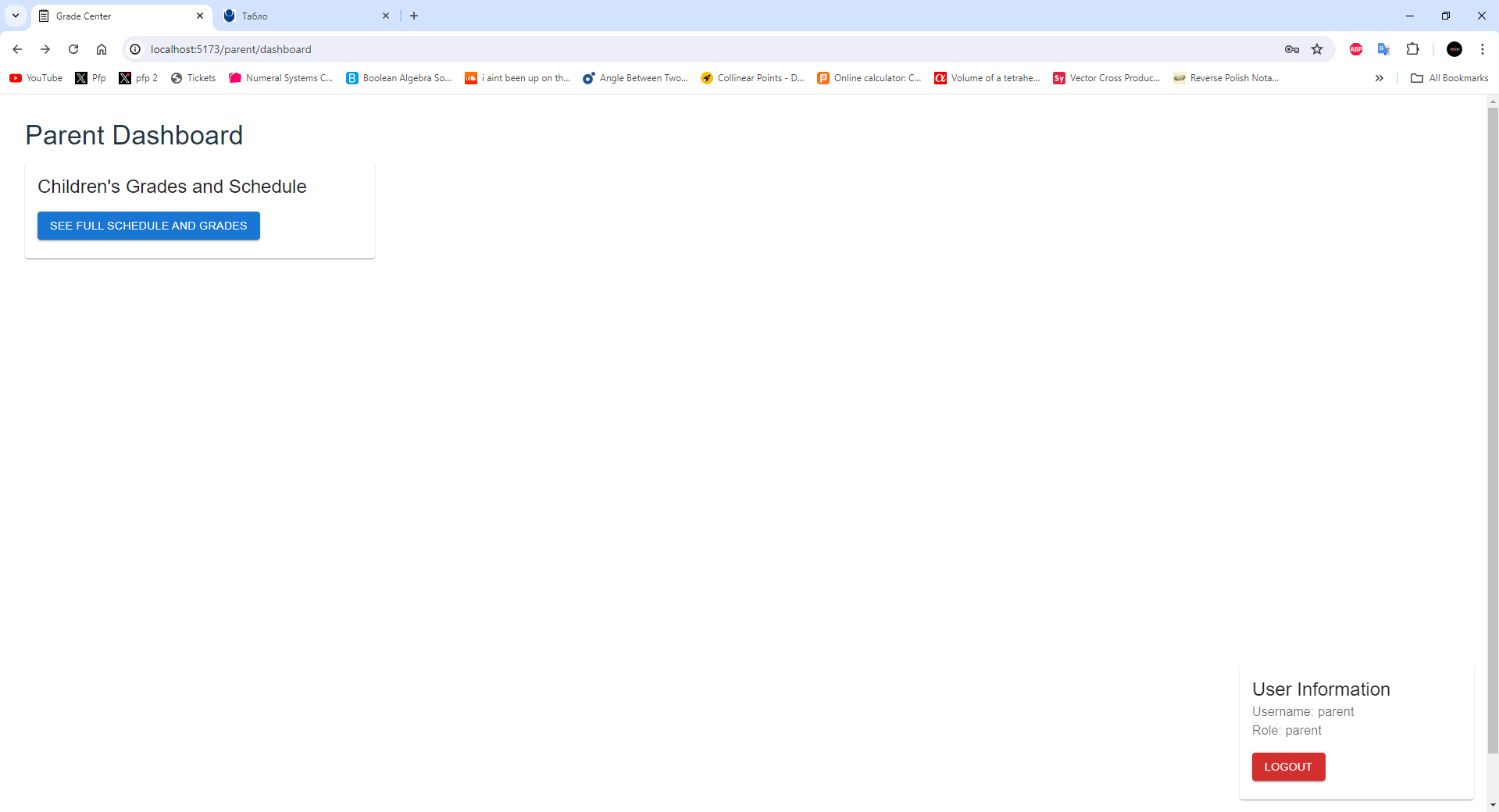


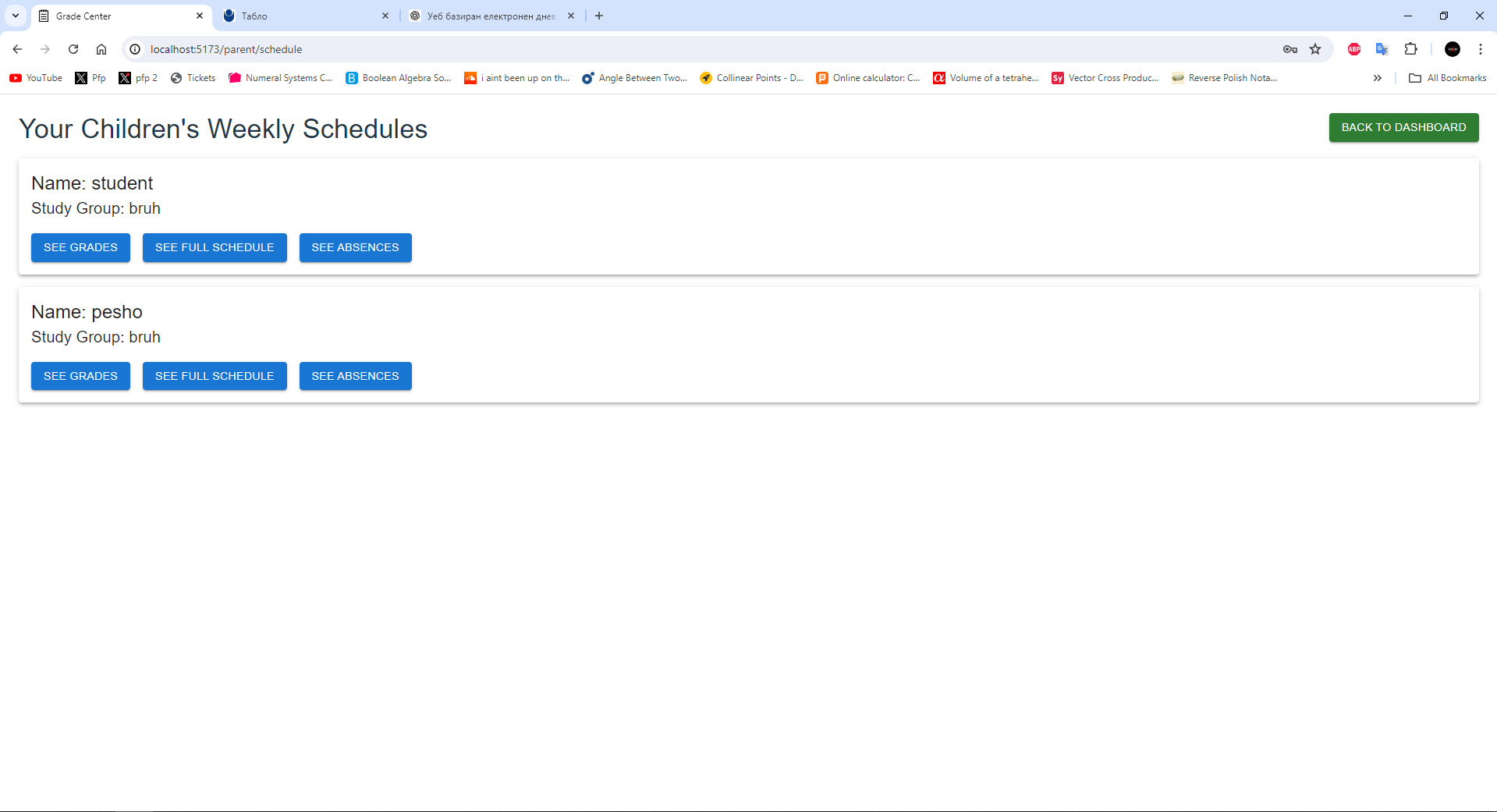


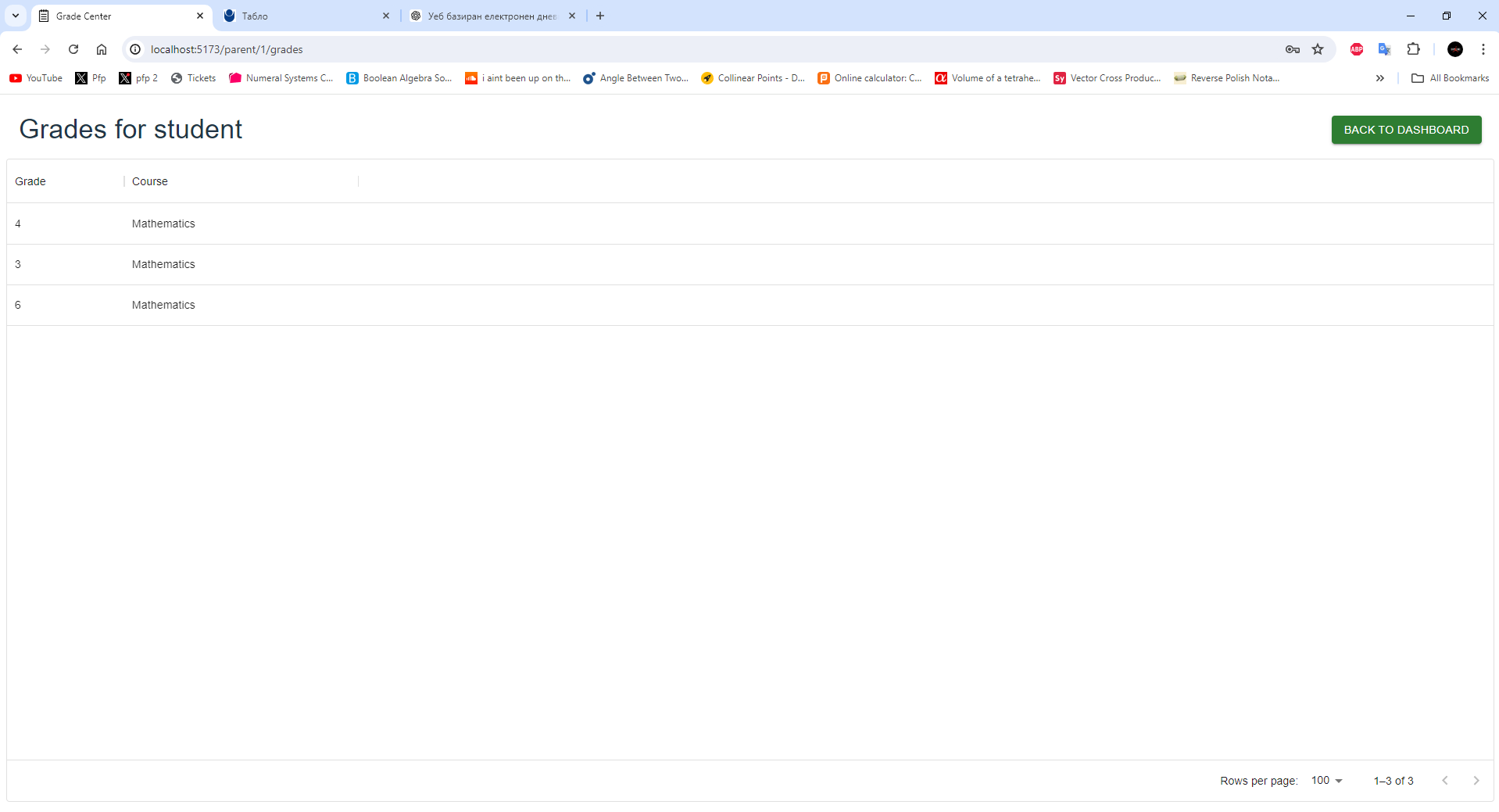


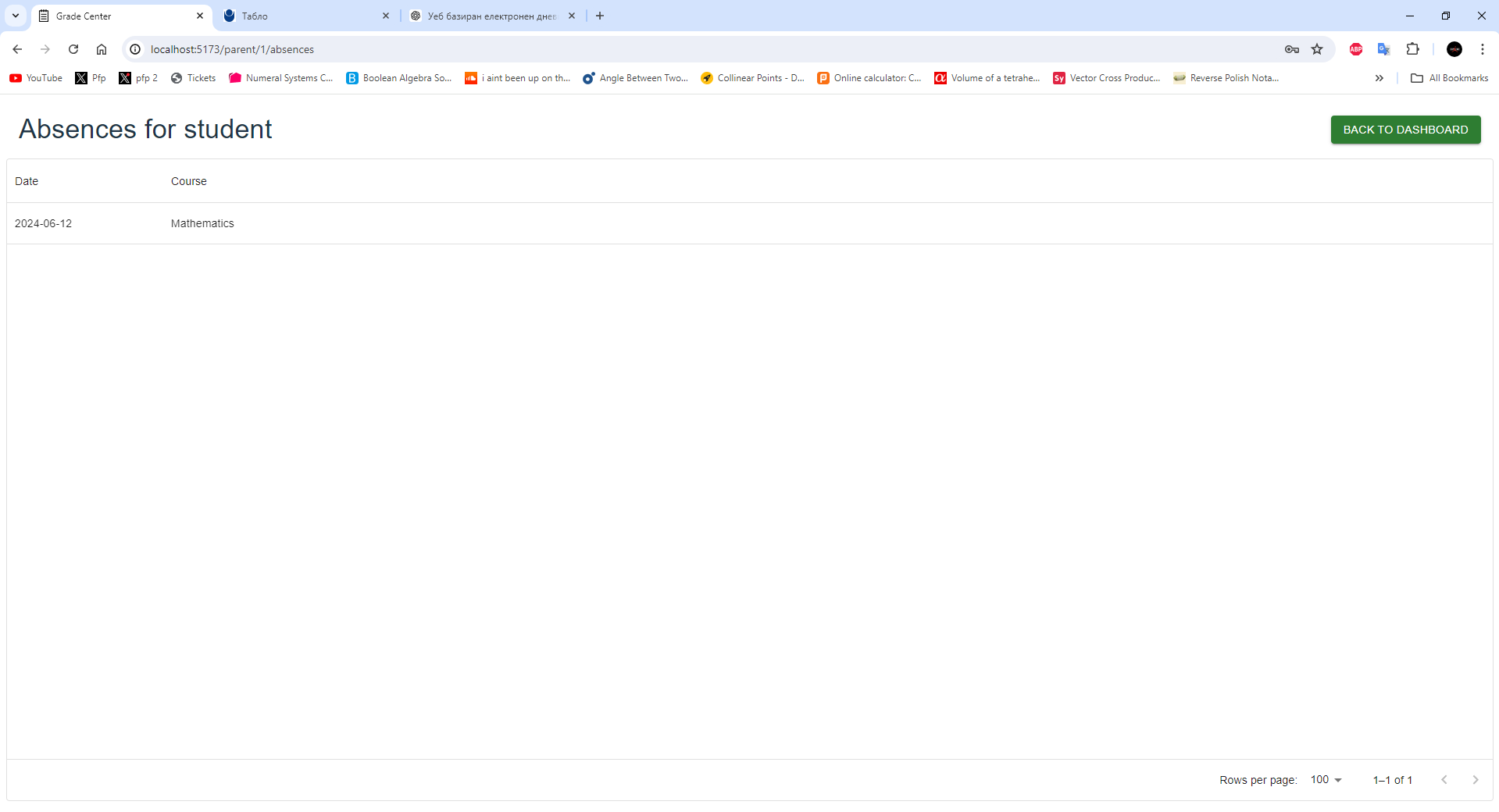


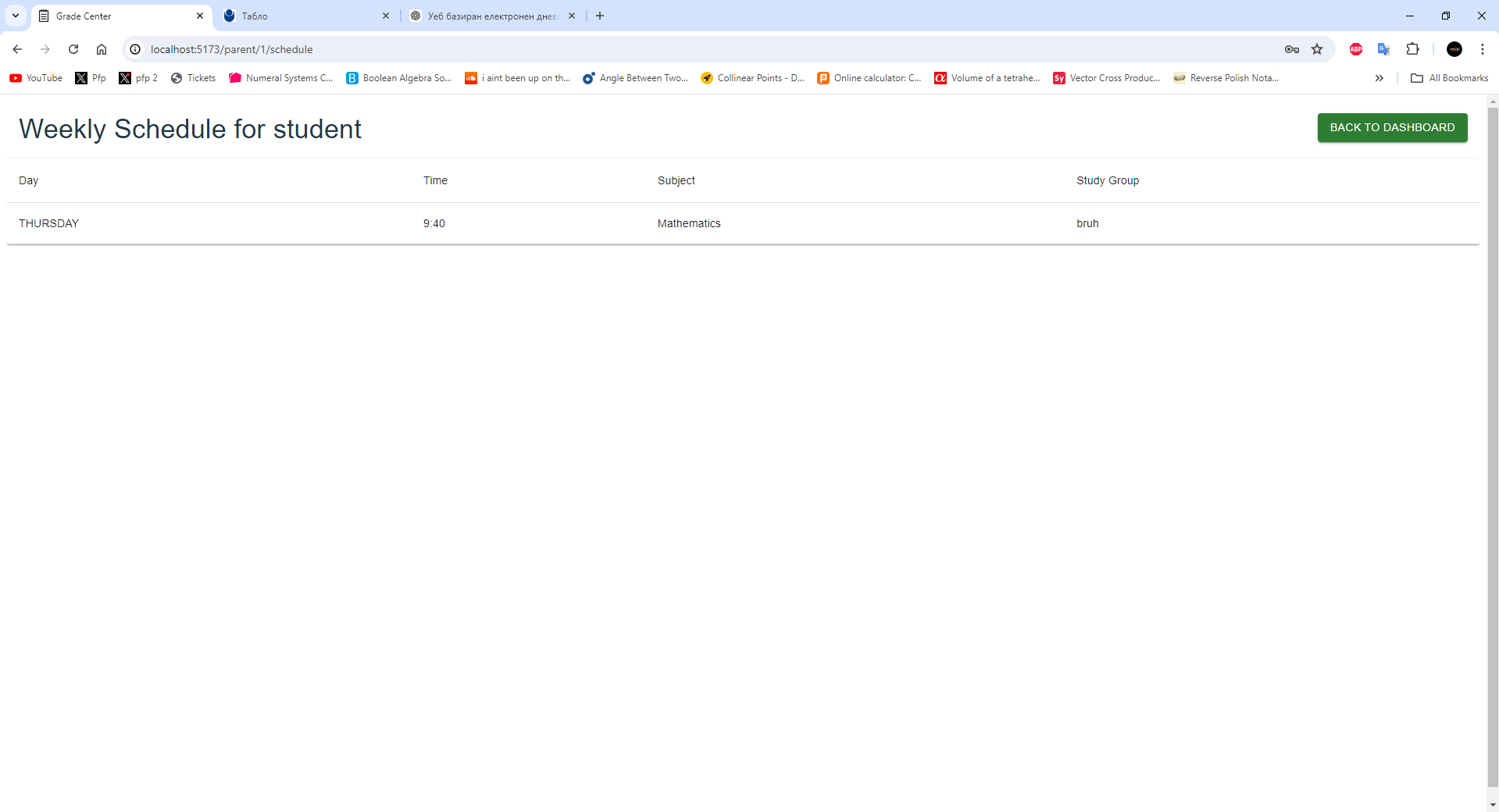
**Всички следващи страници са за потребителите с роля родител:**



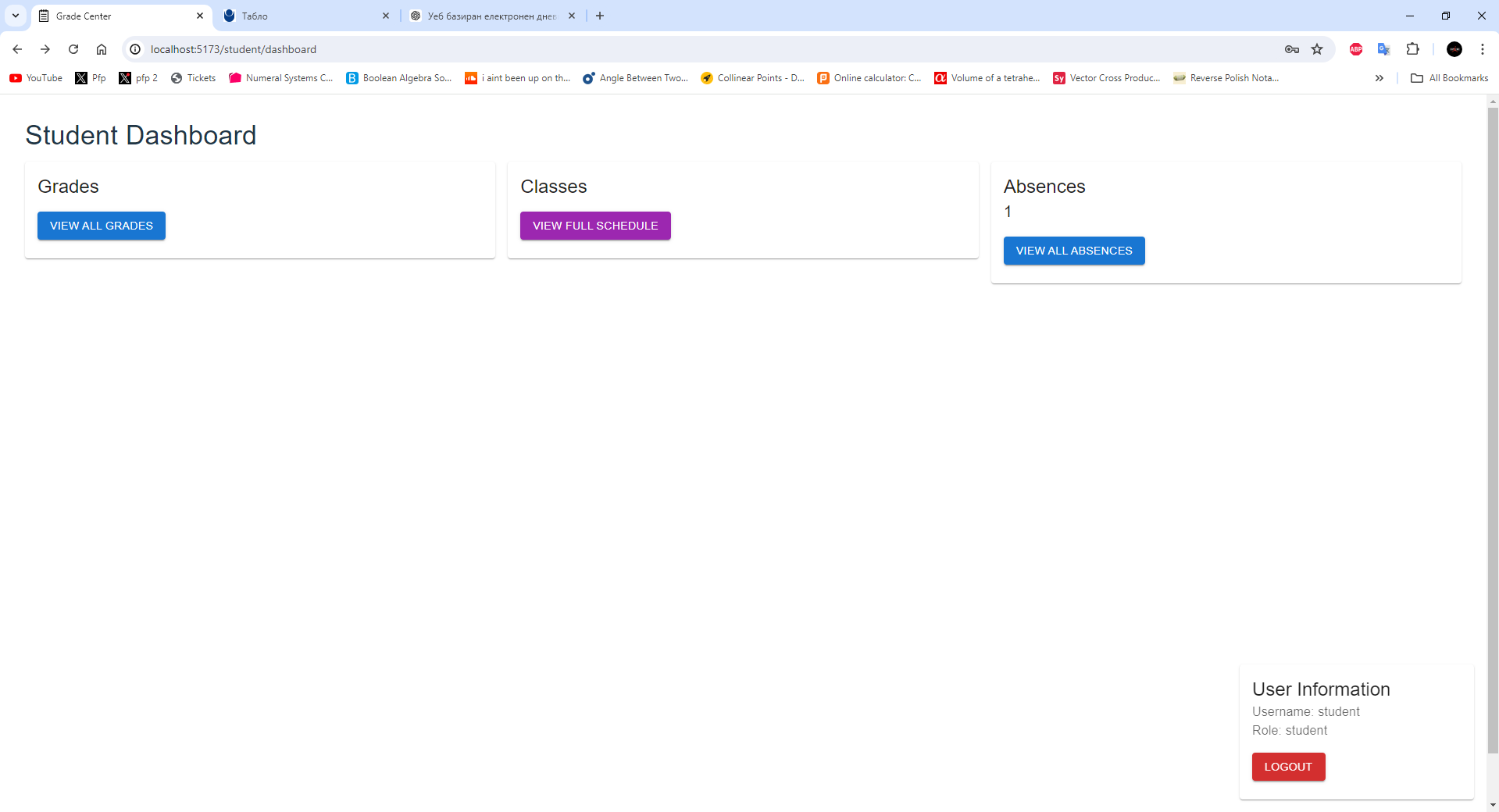


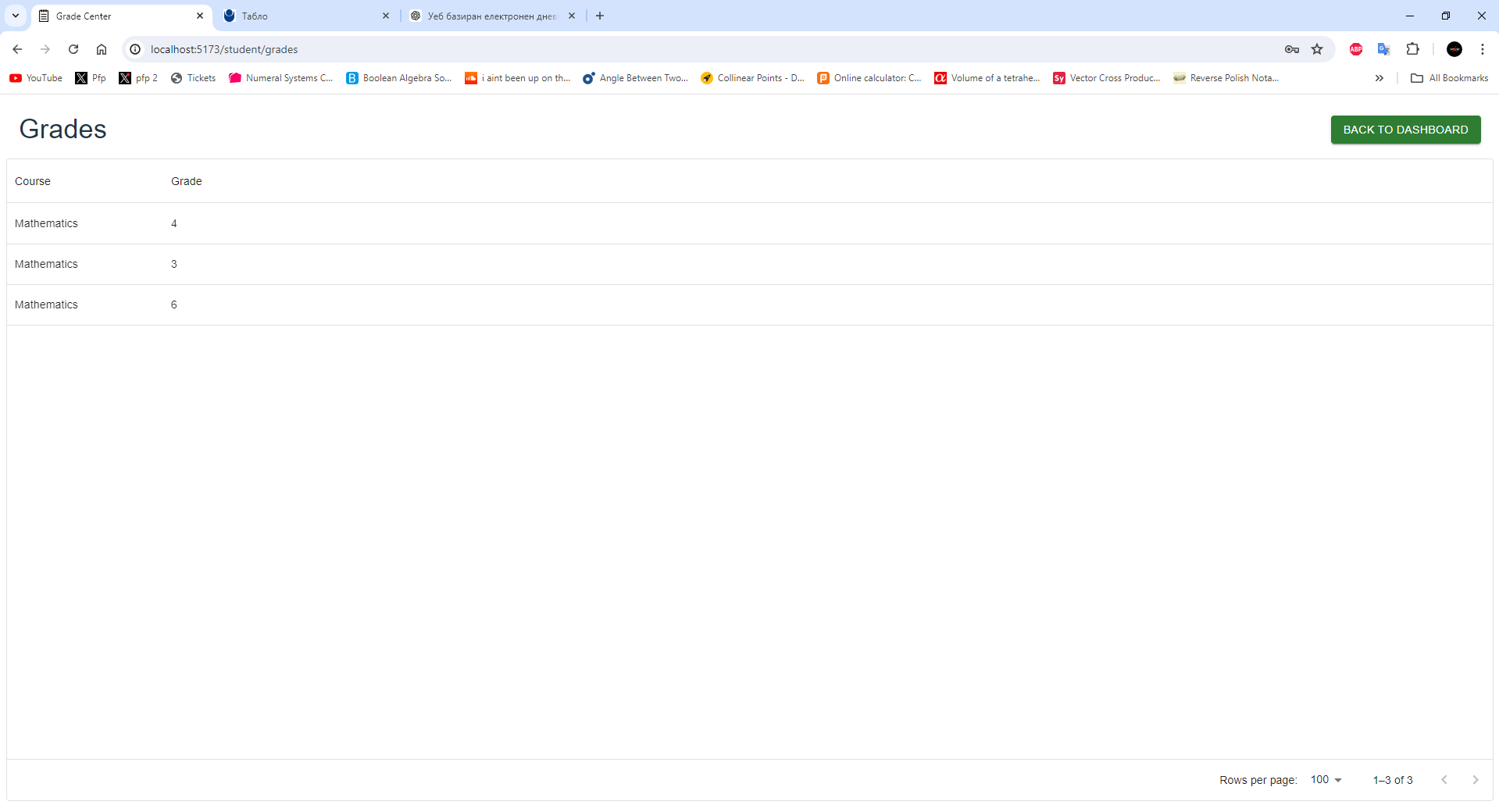


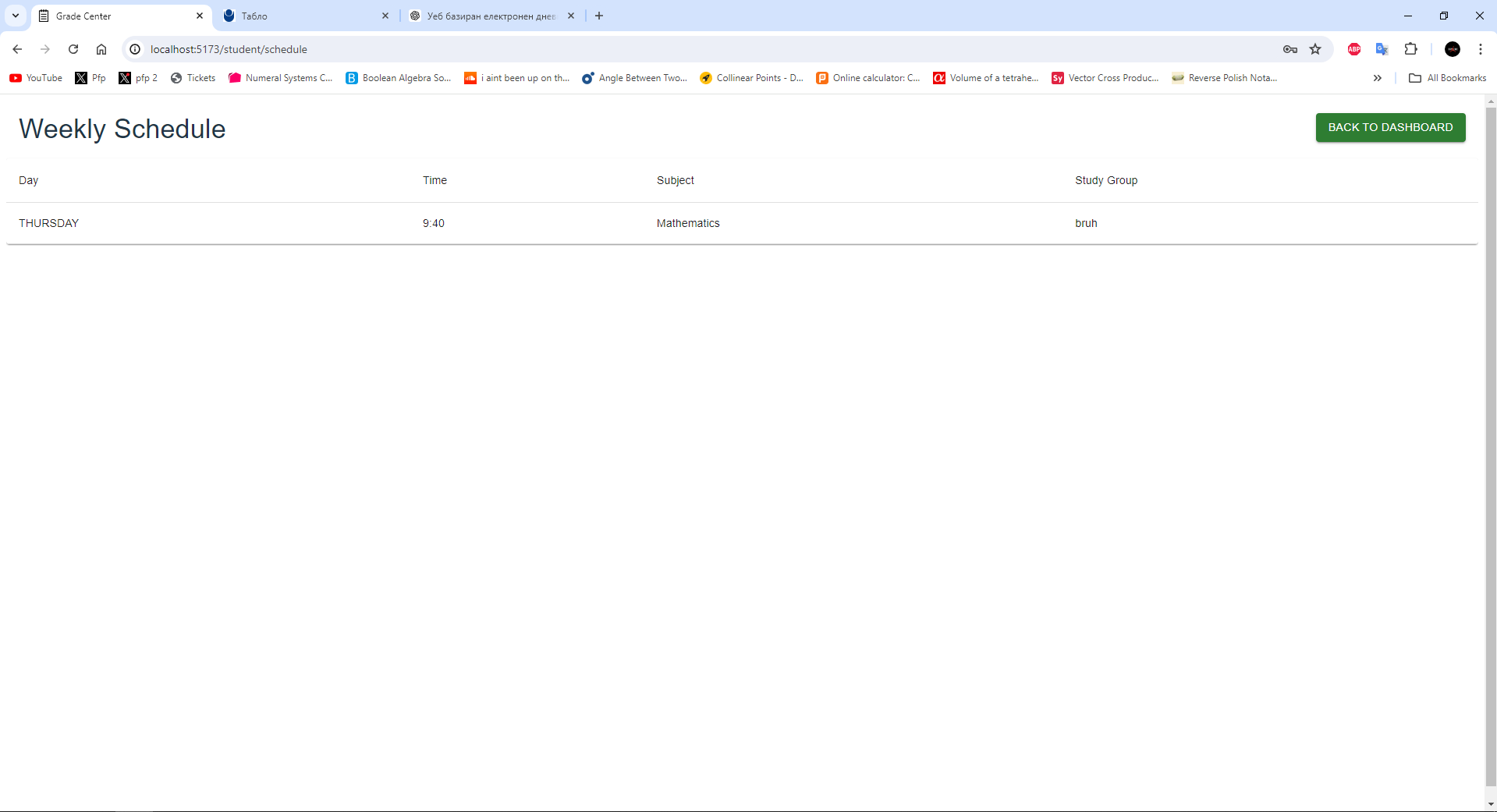


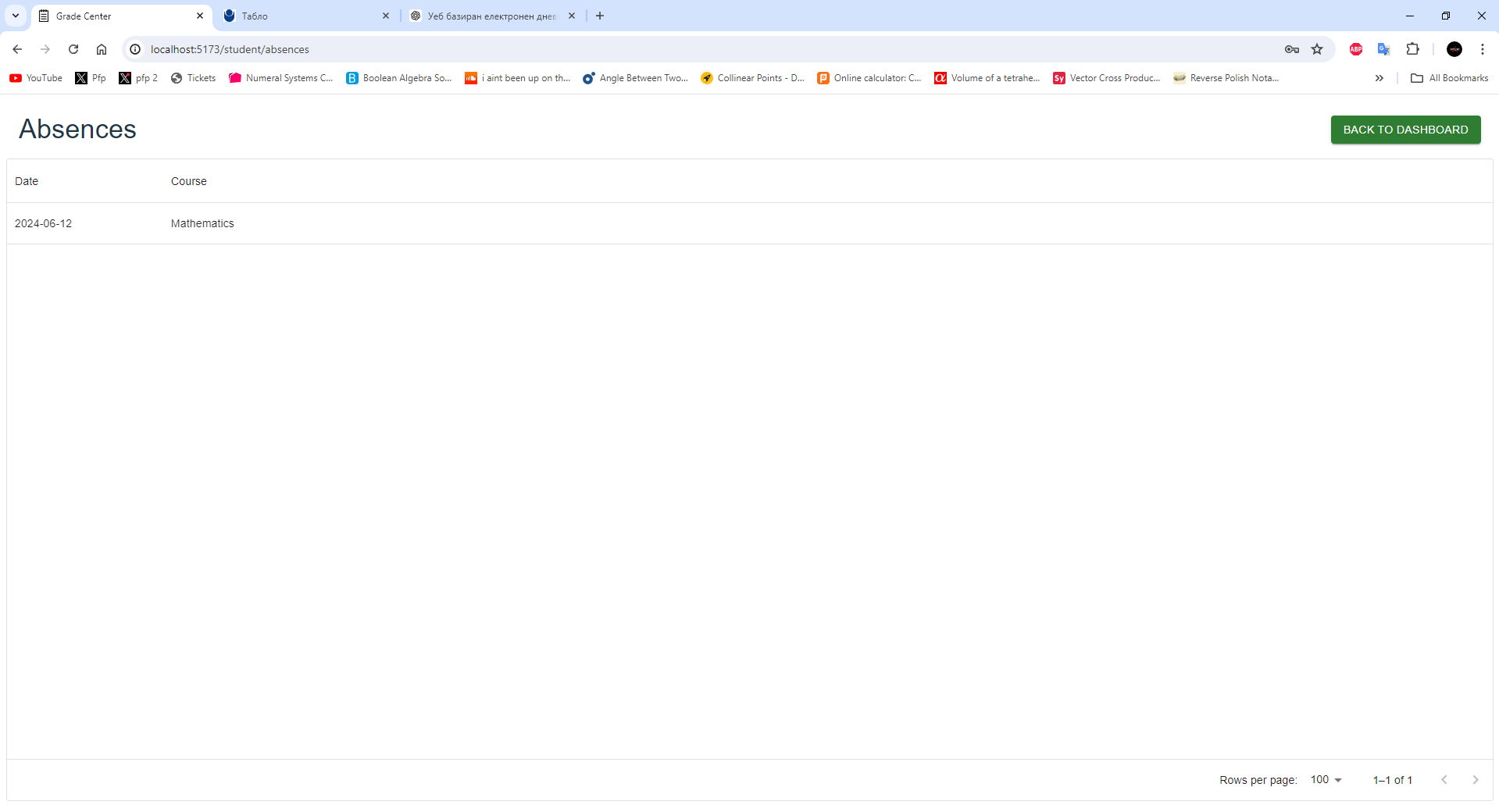


**Всички следващи снимки са за потребителите с роля ученик:**









**Обобщение**

Сайтът е напълно респонсив и е правен изцяло самостоятелно от Денис Велиев (F104630). Всички данни, включително и в екранните снимки, са взети от базата данни, използвайки бак-енда на апликацията.