**Документация**

на проекта

„Уеб базиран електронен дневник“

по дисциплината

„CSCB634 Практика по програмиране и по реализация на бази данни“

Изготвил: Веселина Послийска

Преподавател: гл. ас. д-р Христина Костадинова

Contents

[1. Функционалности 4](#_Toc201009772)

[2. Използвани технологии 4](#_Toc201009773)

[3. ER диаграма и описание на схемата на базата от данни 4](#_Toc201009774)

[4. Разпределяне на работата 7](#_Toc201009775)

[5. Екрани 10](#_Toc201009776)

Проектът „Уеб базиран електронен дневник“ има за цел да създаде приложение, с което да бъде възможно управлението на учебния процес на учениците.

1. **Функционалности**

Функционалностите на приложението са следните:

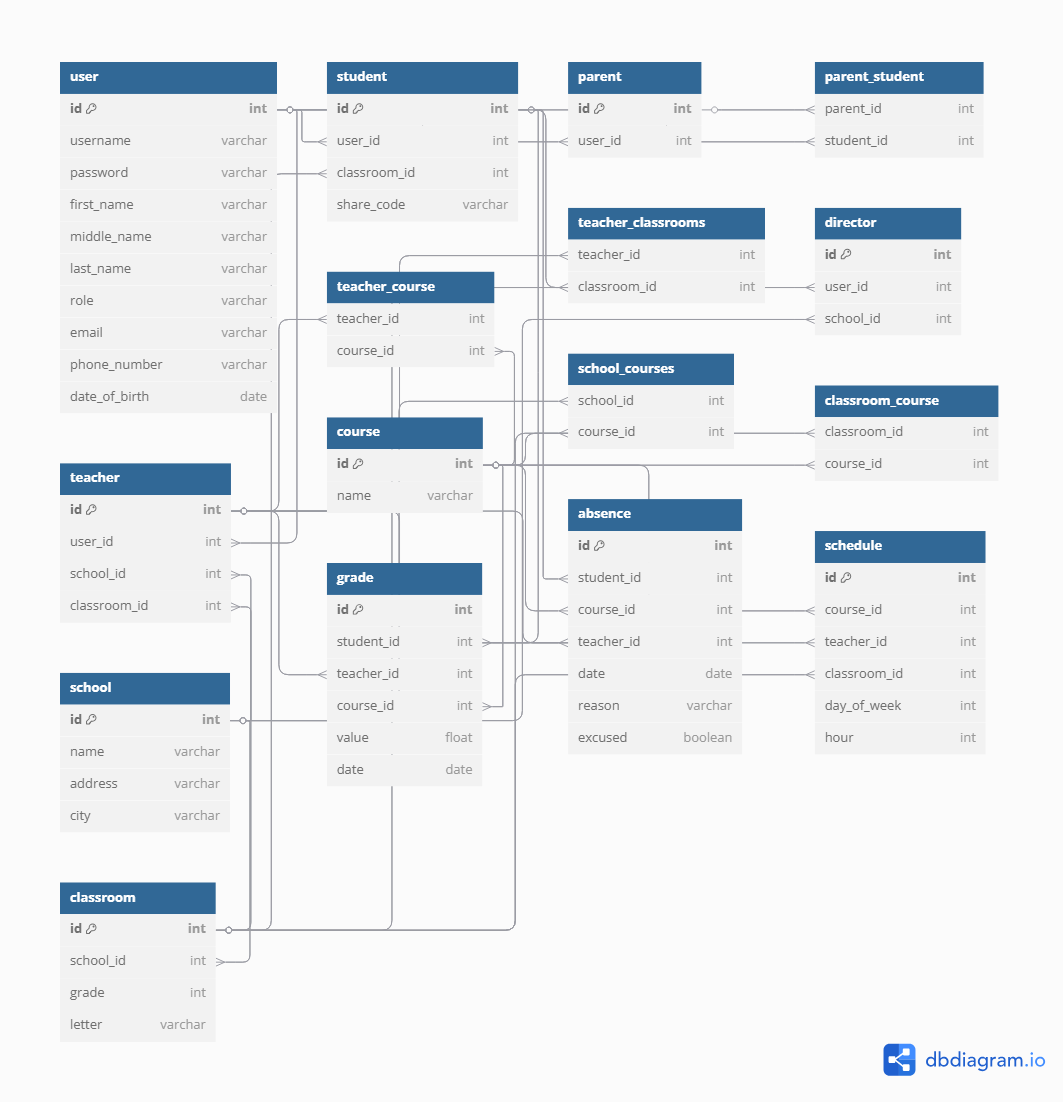
* Създаване и управляване на информация за ученици
* Създаване и управляване на информация за учители
* Създаване и управляване на информация за учебни курове
* Създаване и управляване на информация за оценки
* Създаване и управляване на информация за отсъствия
* Създаване и управляване на информация за учебно разписание
* Филтриране на информация за студенти, лектори и курсове
* Създаване на статистика за оценки по предмети, училища, учители
* Създаване на статистика за отсъствия по предмети, училища, учители

1. **Използвани технологии**

Уеб приложението е разработено с помощта на програмния език Python и микрофреймуърка Flask, който предоставя лека и гъвкава основа за изграждане на съвременни уеб системи. За работа с базата от данни е използвана библиотеката Flask-SQLAlchemy, която осигурява удобен начин за дефиниране и манипулиране на модели чрез обектно-релационно съпоставяне (ORM). Като база данни е използвана SQLite, което улеснява разработката и тестването, без необходимост от инсталация на външни сървъри.

Интерфейсът на приложението е създаден с помощта на HTML, CSS и фреймуърка Bootstrap, който допринася за адаптивен и естетичен дизайн. Формите са реализирани чрез WTForms, което позволява лесна валидация и обработка на входните данни. За добавяне на интерактивност е използван JavaScript, а Jinja2 служи като шаблонизираща система, интегрирана във Flask, за динамично генериране на съдържание.

1. **ER диаграма и описание на схемата на базата от данни**



Таблици в базата данни:

User – съхранява данните за потребителите (username, password, име, фамилия, роля, email, телефон и др.). Връзки към профили на ученици, учители, родители и директори.

Student – съдържа данните за учениците, включително връзка към потребител и към клас, код за споделяне (share\_code), оценки и отсъствия.

Parent – съхранява данните за родителите, връзка към потребител и много към много връзка към децата (учениците).

Teacher – съхранява данните за учителите, връзка към потребител, към училище и към клас, както и много към много връзки към предмети и класове, оценки и отсъствия.

Director – съдържа информация за директорите, връзка към потребител и към училище.

School – съхранява информация за училищата (име, адрес, град), връзки към класове, учители, директор и предмети.

Course – съдържа предметите/курсовете (уникално име), връзки към учители, класове, училища, оценки.

Classroom – съхранява данните за класовете (към кое училище принадлежи, клас, паралелка), връзки към ученици, курсове, учители, график и класен ръководител.

Grade – съхранява оценките, дадени на учениците, с връзки към ученик, учител, предмет и дата.

Absence – съдържа данни за отсъствията на учениците, включително връзки към ученик, учител, предмет, дата, причина и дали отсъствието е извинено.

Schedule – съхранява разписанието (кой предмет, учител, ден от седмицата, час) за даден клас.

parent\_student – асоциативна таблица, реализираща много към много връзката между родители и ученици.

teacher\_course – асоциативна таблица за много към много връзката между учители и предмети.

teacher\_classrooms – асоциативна таблица за много към много връзката между учители и класове.

classroom\_course – асоциативна таблица за много към много връзката между класове и предмети.

school\_courses – асоциативна таблица за много към много връзката между училища и предмети.

**Връзки между таблиците:**

* **User – Role:** много към едно (ролята на потребителя се съхранява като поле в User).
* **User – Student:** едно към едно (всеки ученик има точно един потребителски профил).
* **User – Parent:** едно към едно (всеки родител има точно един потребителски профил).
* **User – Teacher:** едно към едно (всеки учител има точно един потребителски профил).
* **User – Director:** едно към едно (всеки директор има точно един потребителски профил).
* **Parent – Student:** много към много (родител може да има няколко деца, дете може да има до двама родители), реализирана чрез parent\_student.
* **Teacher – Course:** много към много (учител преподава няколко предмета, предмет може да има много учители), чрез teacher\_course.
* **Teacher – Classroom:** много към много (учител може да преподава в няколко класа, клас има много учители), чрез teacher\_classrooms.
* **School – Course:** много към много (училището предлага няколко предмета, предмет се изучава в няколко училища), чрез school\_courses.
* **Classroom – Course:** много към много (клас изучава няколко предмета, предмет се изучава в няколко класа), чрез classroom\_course.
* **School – Classroom:** едно към много (училище има много класове).
* **School – Teacher:** едно към много (училището има много учители).
* **School – Director:** едно към едно (всяко училище има един директор).
* **Classroom – Student:** едно към много (класът има много ученици).
* **Classroom – Teacher (класен ръководител):** едно към едно (класен ръководител е един учител на даден клас).
* **Student – Grade:** едно към много (ученик има много оценки).
* **Teacher – Grade:** едно към много (учител дава много оценки).
* **Course – Grade:** едно към много (предмет има много оценки).
* **Student – Absence:** едно към много (ученик има много отсъствия).
* **Teacher – Absence:** едно към много (учител отчита много отсъствия).
* **Course – Absence:** едно към много (предмет има много отсъствия).
* **Classroom – Schedule:** едно към много (клас има разписание с много записи).
* **Course – Schedule:** едно към много (предмет има много разписания).
* **Teacher – Schedule:** едно към много (учител има много разписания).

1. **Разпределяне на работата**

**Веселина Послийска**

* Създаване на базата от данни, таблиците и връзките между тях
* Authentication и Authorization  
  - създаване на механизъм за запазване на потребителите на приложението

- определяне на роли

- ограничаване на достъпа до различни части на приложението (имплементирано чрез персонализирани анотации)

* Back-end на управлението на студентите

- създаване/редактиране/изтриване на студенти;

- четене на студенти по дадени критерии;

- възможност за sorting

* Back-end на управлението на лекторите

- създаване/редактиране/изтриване на лектори;

- четене на лектори по дадени критерии;

- възможност за sorting и pagination

* Back-end на управлението на курсовете

- създаване/редактиране/изтриване на курсове;

- четене на курсове по дадени критерии;

- възможност за разпределяне на седмична програма;

- добавяне/редактиране/изтриване на седмични раздели

- добавяне/премахване на студенти от курс

* Back-end на управлението на седмичните дейности

- създаване/редактиране/изтриване на седмични дейности

- четене и сортиране на седмични дейности

* Управление на грешки, предизвикани при:

- невалидно въведени данни

- опит да бъде достъпен несъществуващ ресурс

- опит да бъде достъпен защитен ресурс, за който потребителят няма нужните права

* Thymeleaf връзка между back-end и front-end на Login екрана
* Thymeleaf връзка между back-end и front-end на основите екрани на администратора

- създаване/редактиране/изтриване на потребител

- създаване/редактиране/изтриване на лектор

- показване на списък с потребители

- показване на списък с лектори

* Thymeleaf връзка между back-end и front-end на основните екрани на лектора

- показване на списък с всички курсове

- показване на списък със създадените от него курсове

- създаване/редактиране/изтриване на курс

- екран с детайлна информация за курс, включваща списък със седмични активности и списък със студенти (с възможност за добавяне и премахване на студент)

- възможност за добавяне на учебни материали/домашни работи и тестове в екрана с детайлната информация на курса

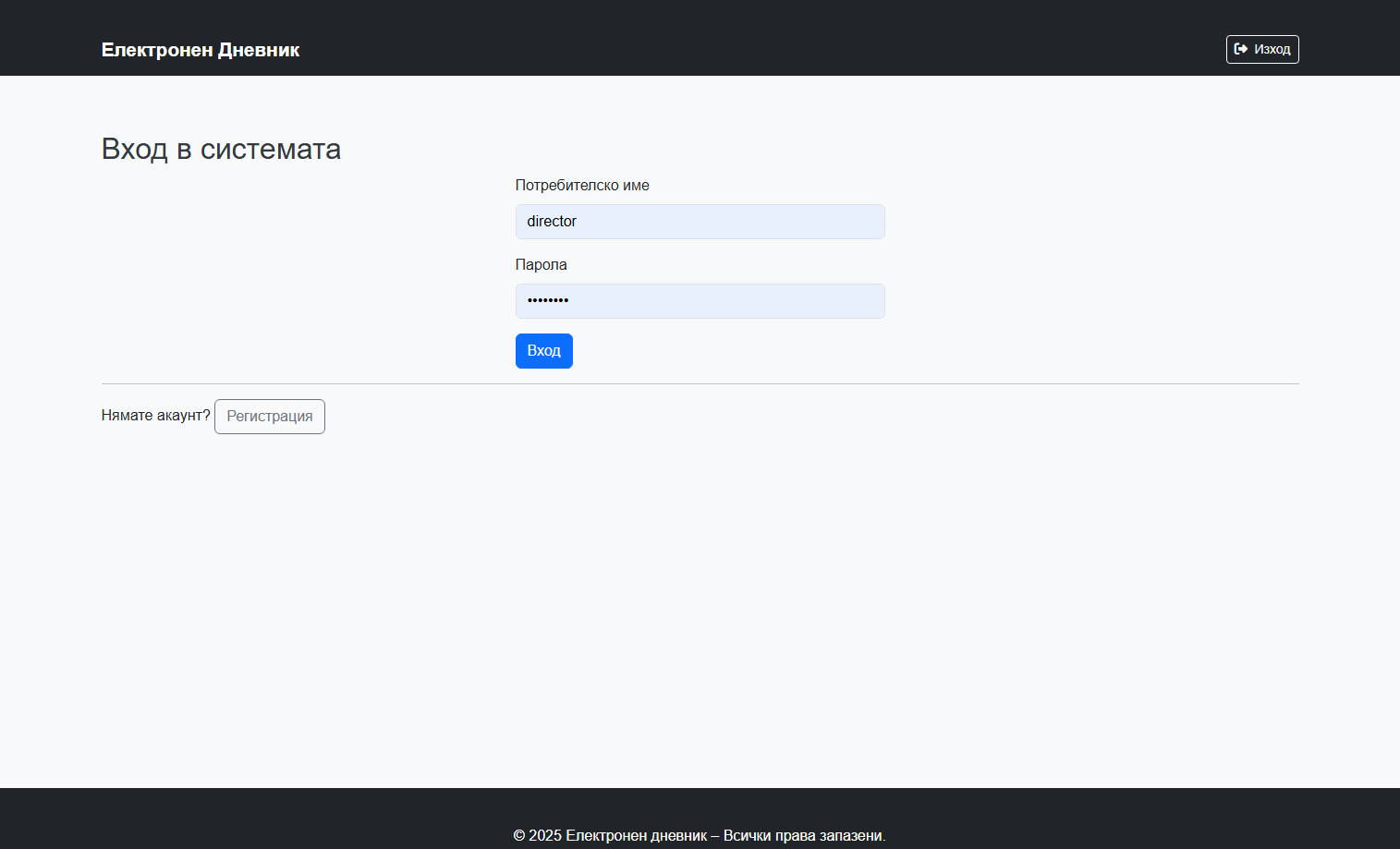
* Thymeleaf връзка между back-end и front-end на основните екрани на студента

- показване на списък с всички курсове

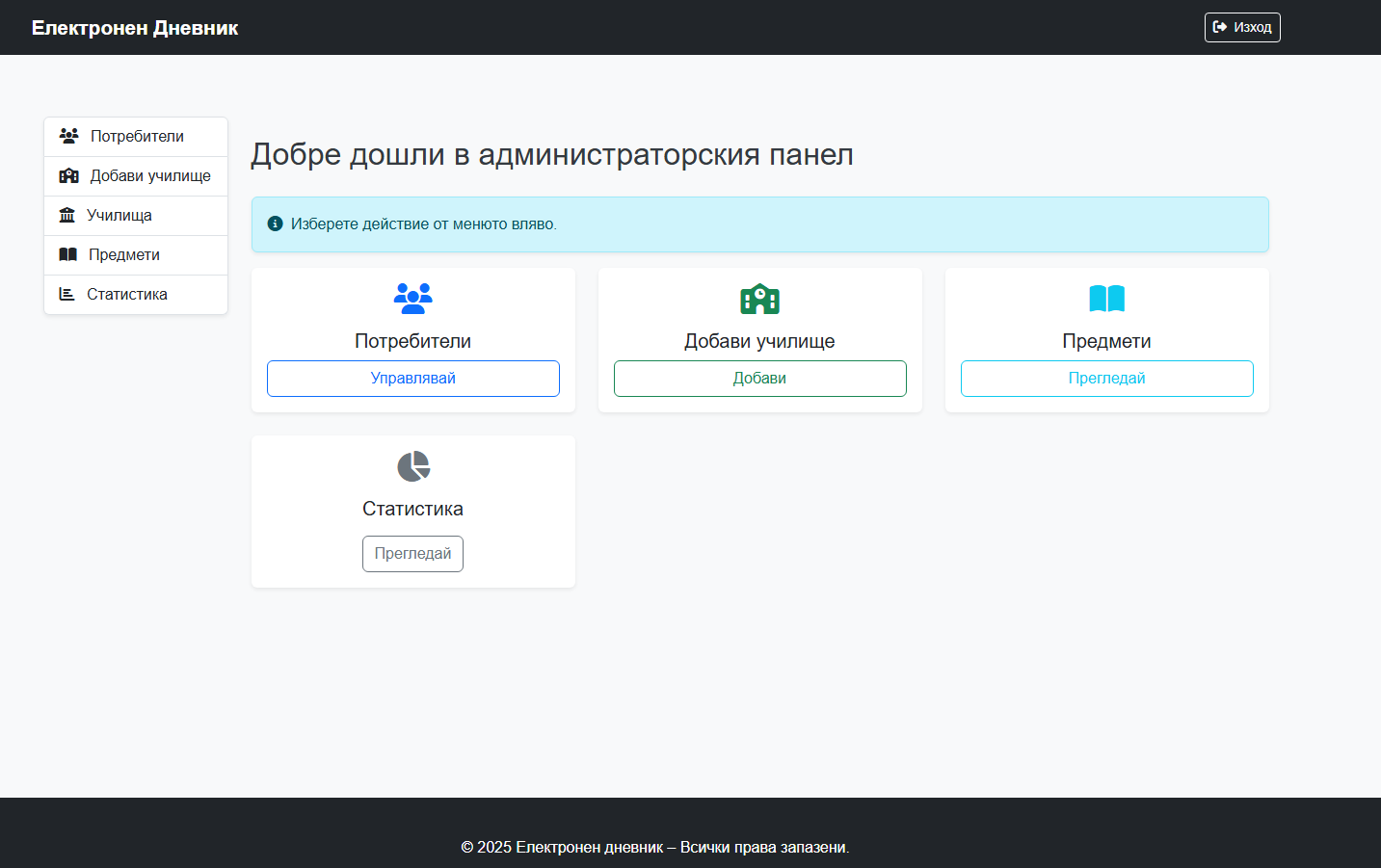
- показване на списък с курсовете, в които е записан

* Изготвяне на документация – база данни и описание на функционалностите
* Front-end на логин страница – използване на Bootstrap, HTML, CSS за форматиране на компонентите на формата за влизане в системата
  + login.html
  + Във fragments.css са стилизирани някои от компонентите
* Front-end създаване на фрагменти и стилизиране, които се използват повече от веднъж в няколко html файла
  + Fragments.html
    - Header-all – хедър, който съществува във всички страници и съдържа логото на сайта и бутон за излизане от системата
    - Header-admin – хедър, който съдържа навигация, която е видима само за администратора
    - Header-lecturer– хедър, който съдържа навигация, която е видима само за лектора
    - Header-student – хедър, който съдържа навигация, която е видима само за студент
    - Students-list – съдържа списък на всички студенти
    - Activities-list - съдържа списък на всички задания
    - Courses-list – съдържа списък на всички курсове
  + Fragments.css
    - Придаване на персонализиран облик на някои от компонентите
* Front-end на заглавните екрани при влизане в системата като Администратор, лектор и студент
  + Index-lecturer.html
  + Index-admin.html
  + Index-student.html
* Front-end на всички страници в които е включено създаване на студент, курс, лектор, дейност
  + Create-student.html - страница, която се възпроизвежда само за админ
  + Create-lecturer.html - страница, която се възпроизвежда само за админ
  + Edit-student.html - страница, която се възпроизвежда само за админ
  + Edit-lecturer.html - страница, която се възпроизвежда само за админ
  + Create-activity.html - страница, която се възпроизвежда само за лектор
  + Create-course.html - страница, която се възпроизвежда само за лектор
* Front-end на всички страници на които се възпроизвеждат данните от back-end
  + Students.html – страница, която се възпроизвежда само за админ – включва списък с всички студенти
  + Students-lecturer.html – страница, която се възпроизвежда само за лектор – включва списък с всички студенти
  + Lecturers.html - страница, която се възпроизвежда само за админ – включва списък с всички лектор
  + My-courses-lecturer.html - страница, която се възпроизвежда само за лектор – включва списък с всички курсове, в които лекторът участва/води
  + My-courses-student.html - страница, която се възпроизвежда само за студент – включва списък с всички курсове, в които студентът участва
  + Course-data.html – страница, която се възпроизвежда само за лектор – включва списък с всички студенти и възможност за добавяне на студент към курс и добавяне на дейност
  + Courses-student.html - страница, която се възпроизвежда само за студента – включва списък с всички курсове
  + Courses.html - страница, която се възпроизвежда само за лектор – включва списък с всички курсове

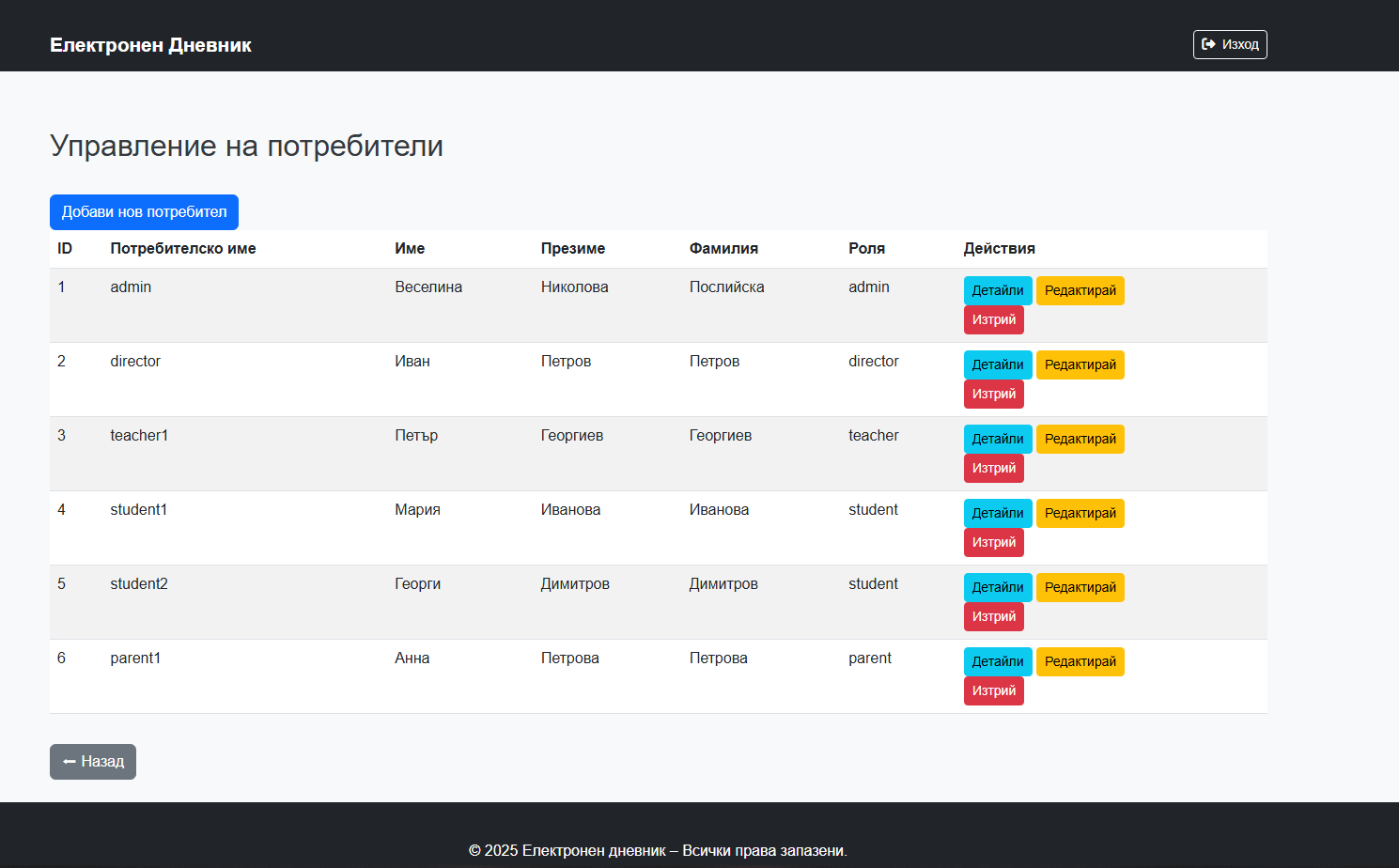
1. **Екрани**
   1. **Логин страница**



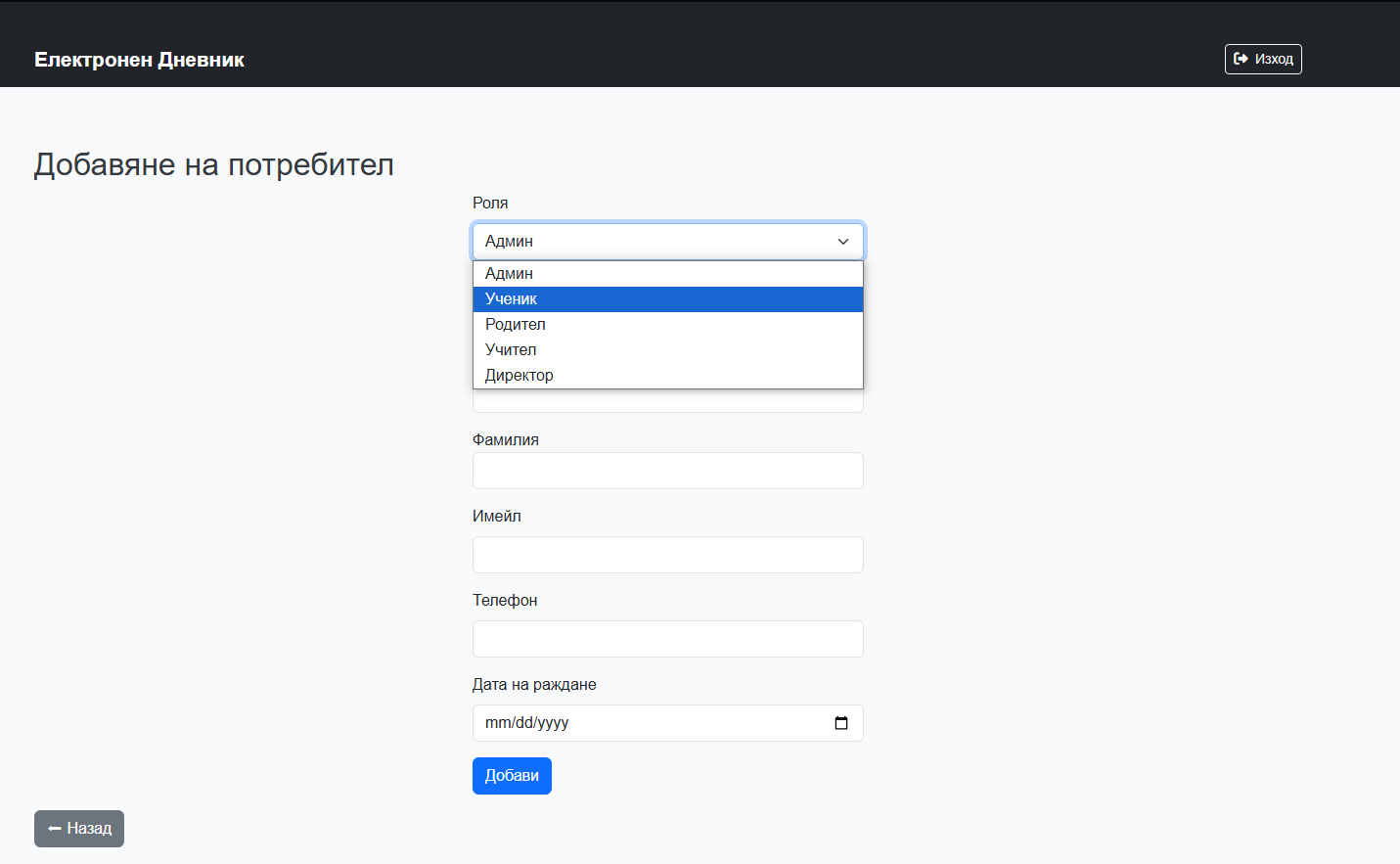
* 1. **Страници при влизане като администратор**
     1. **Заглавна страница при влизане като администратор**

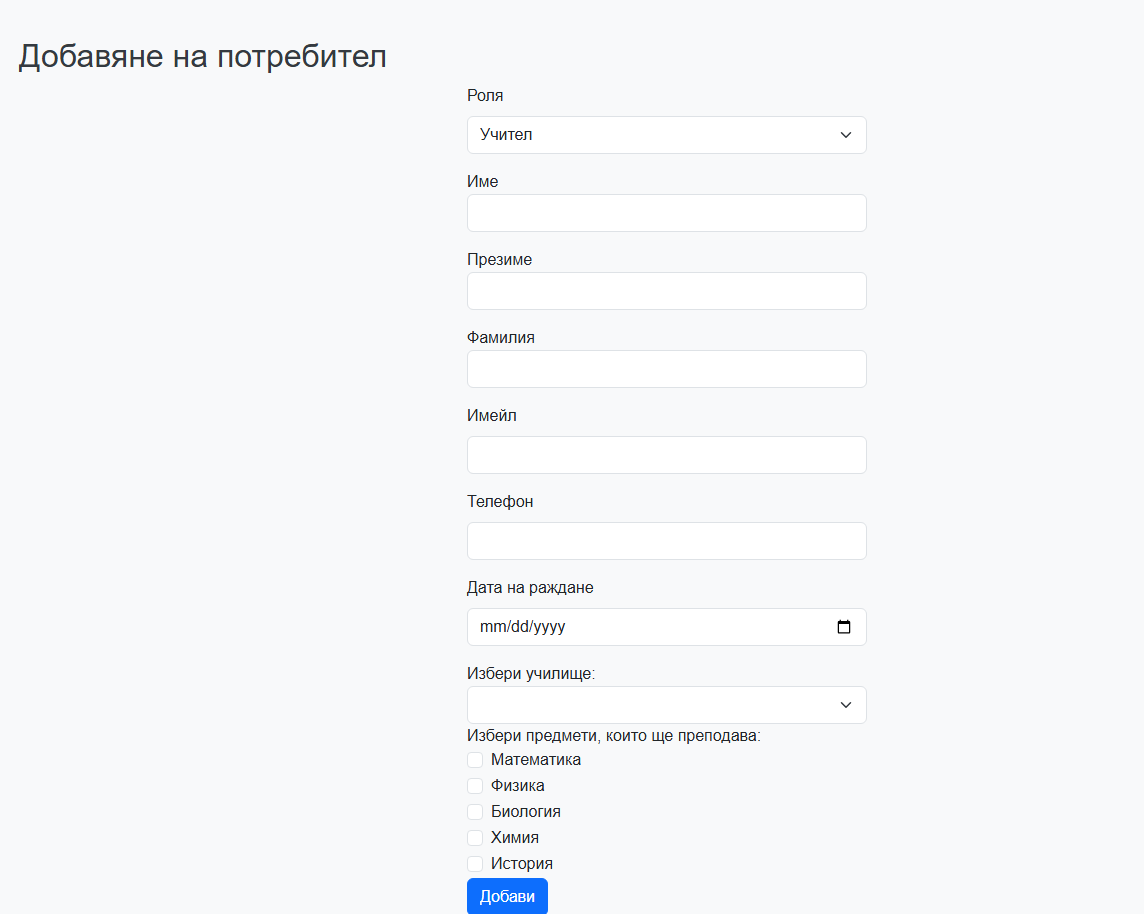
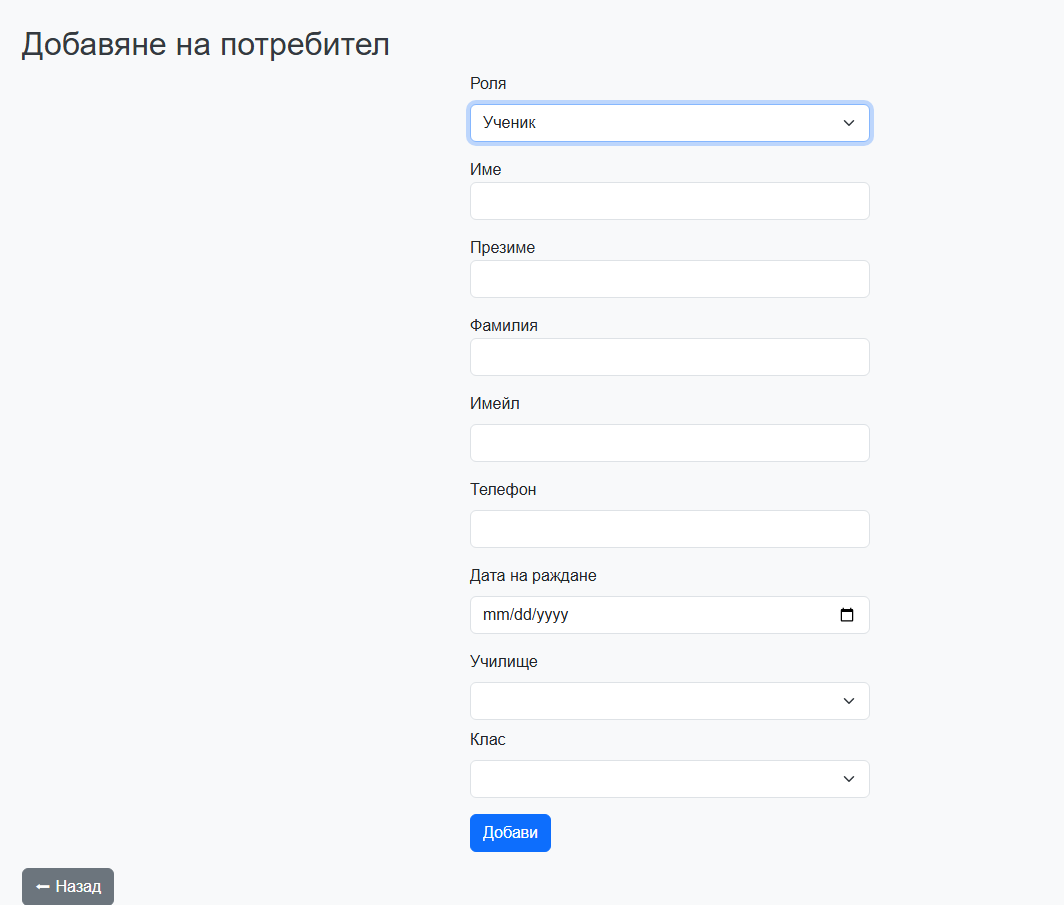
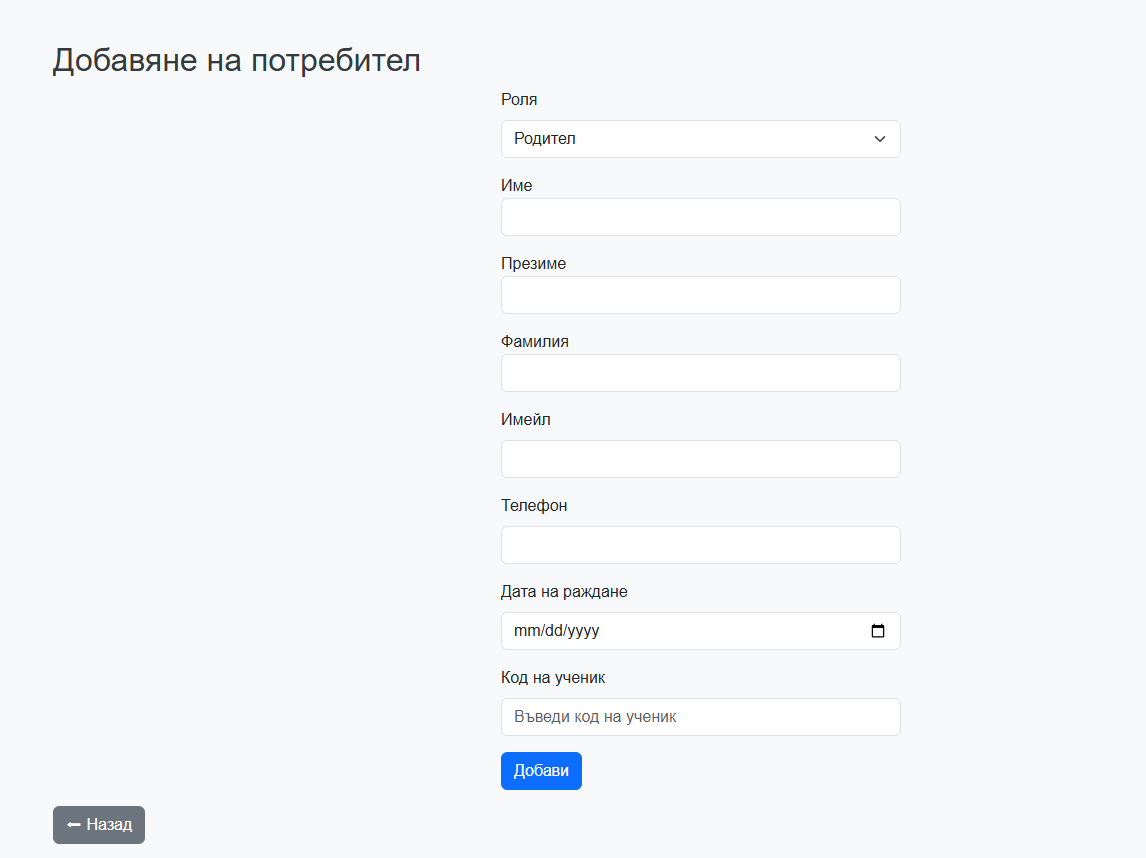
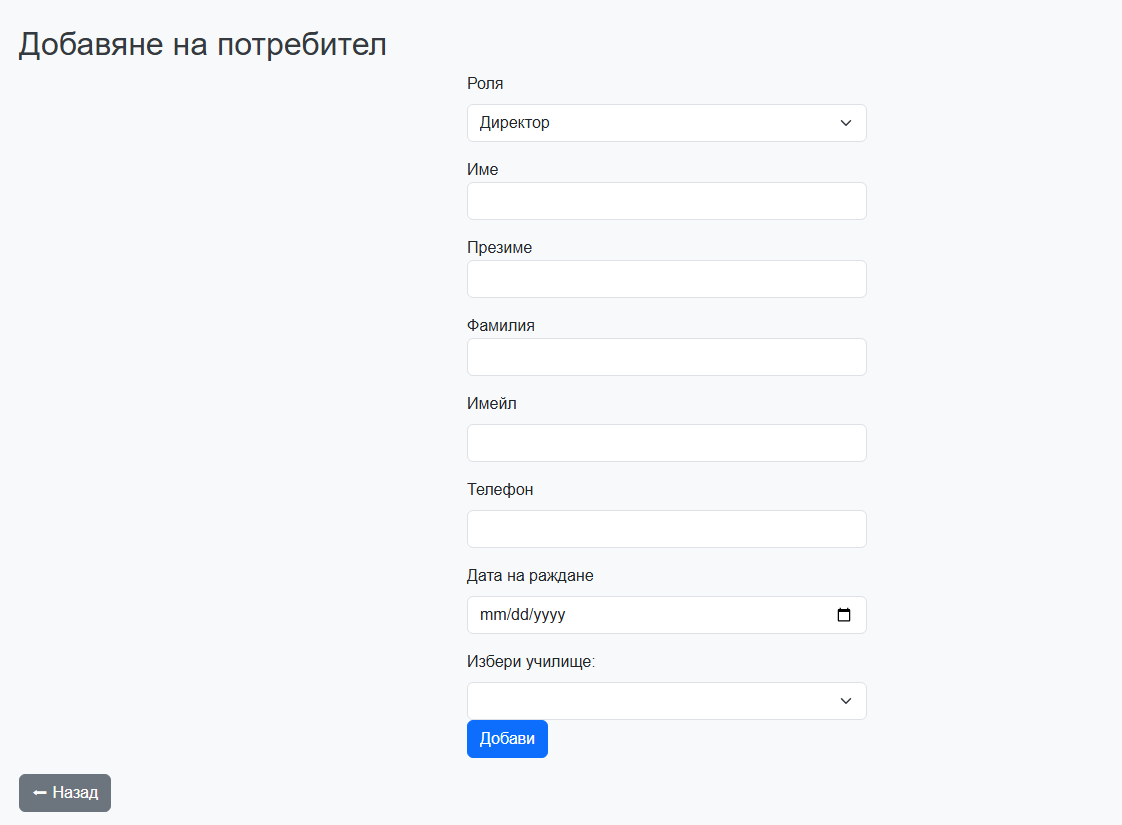


* + 1. **Страница с потребители**

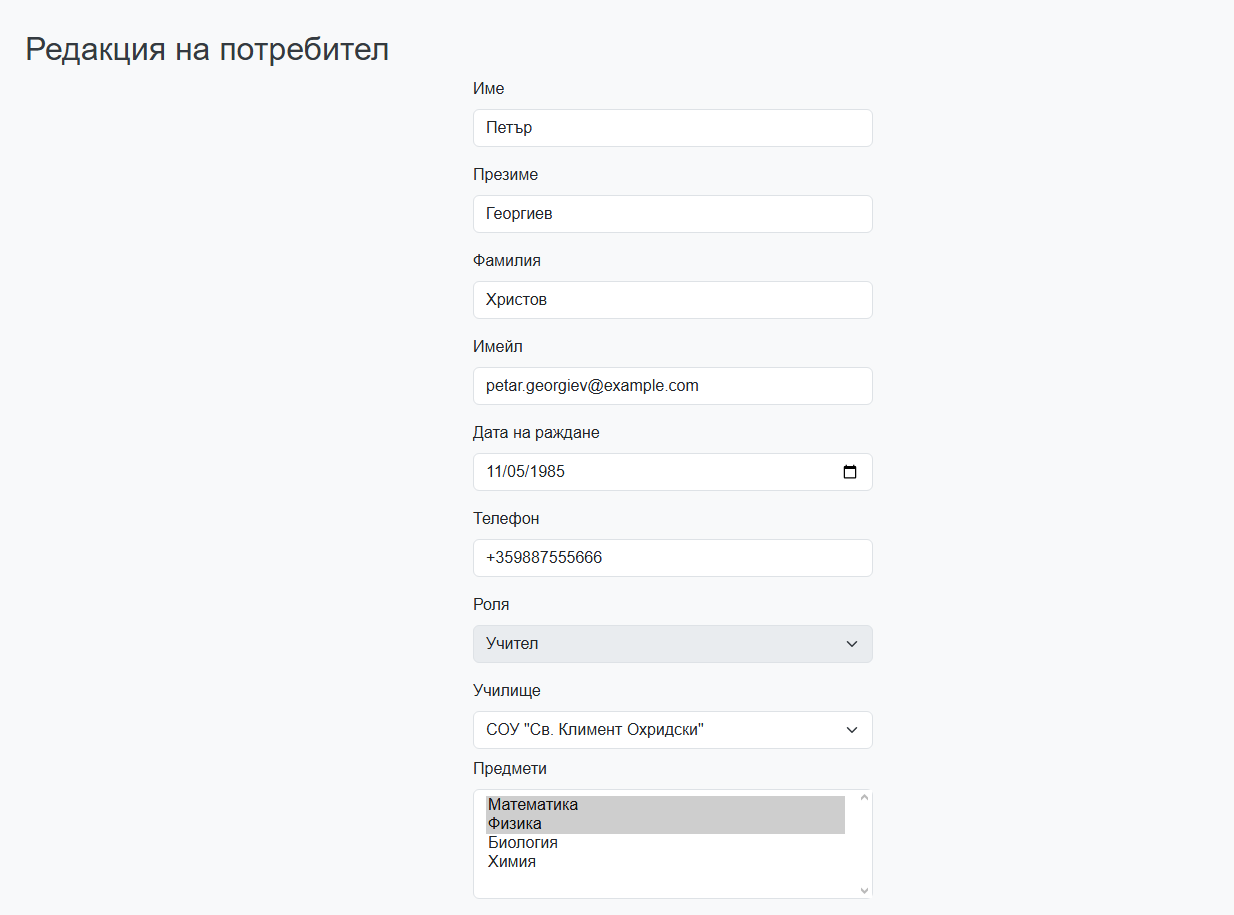
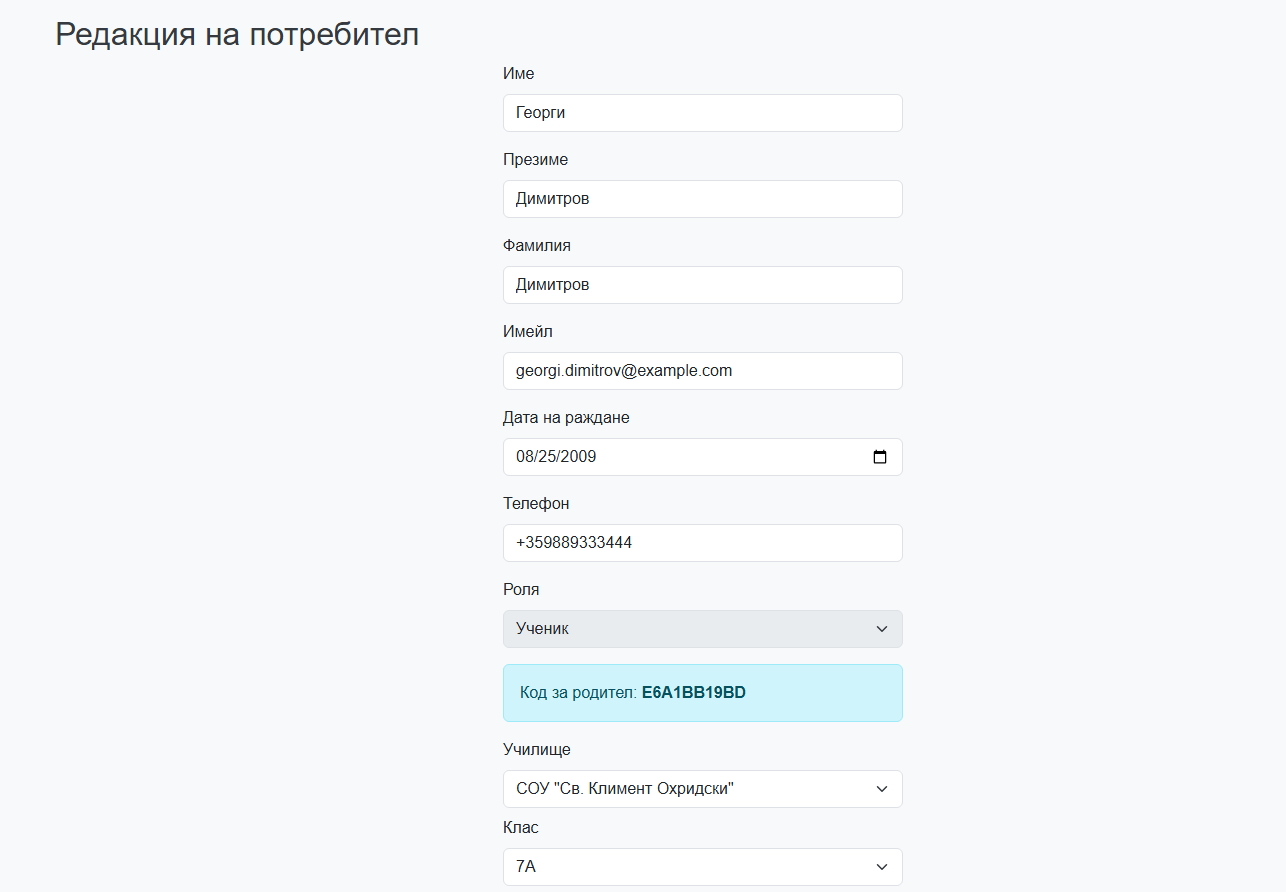
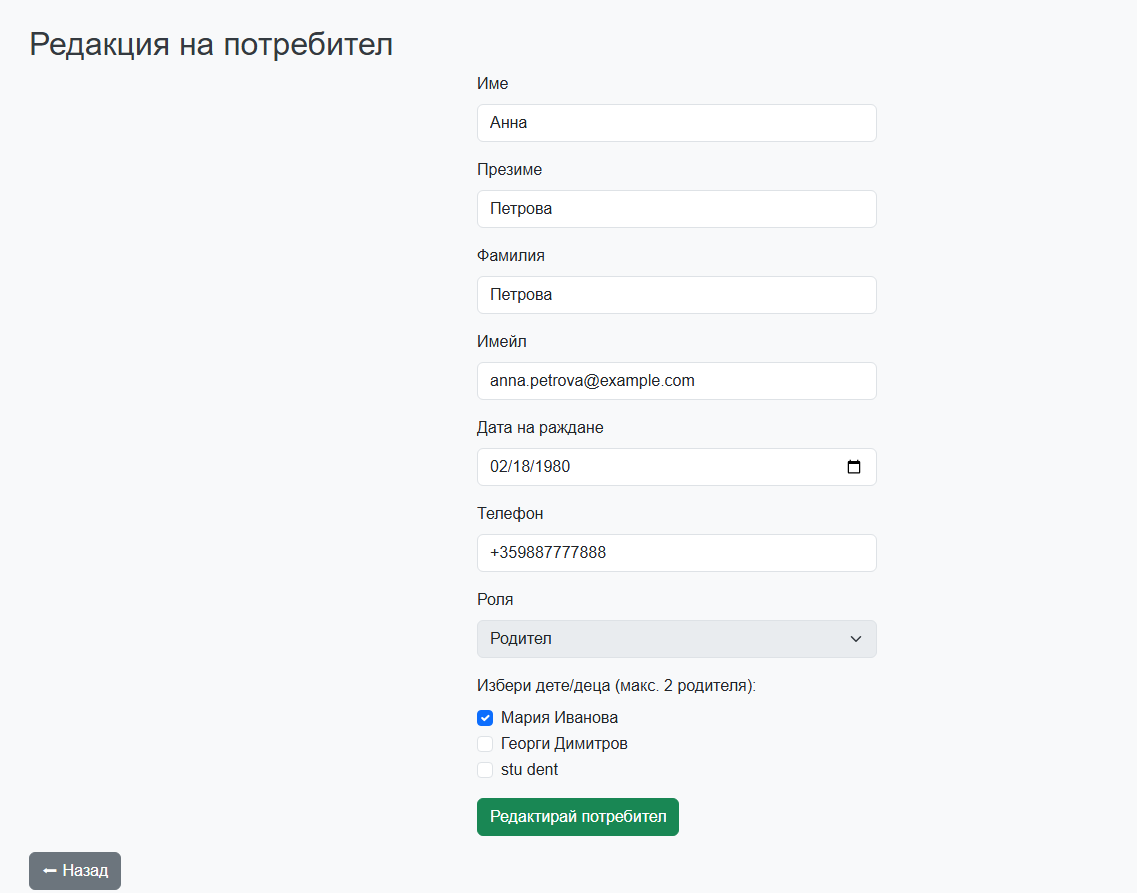
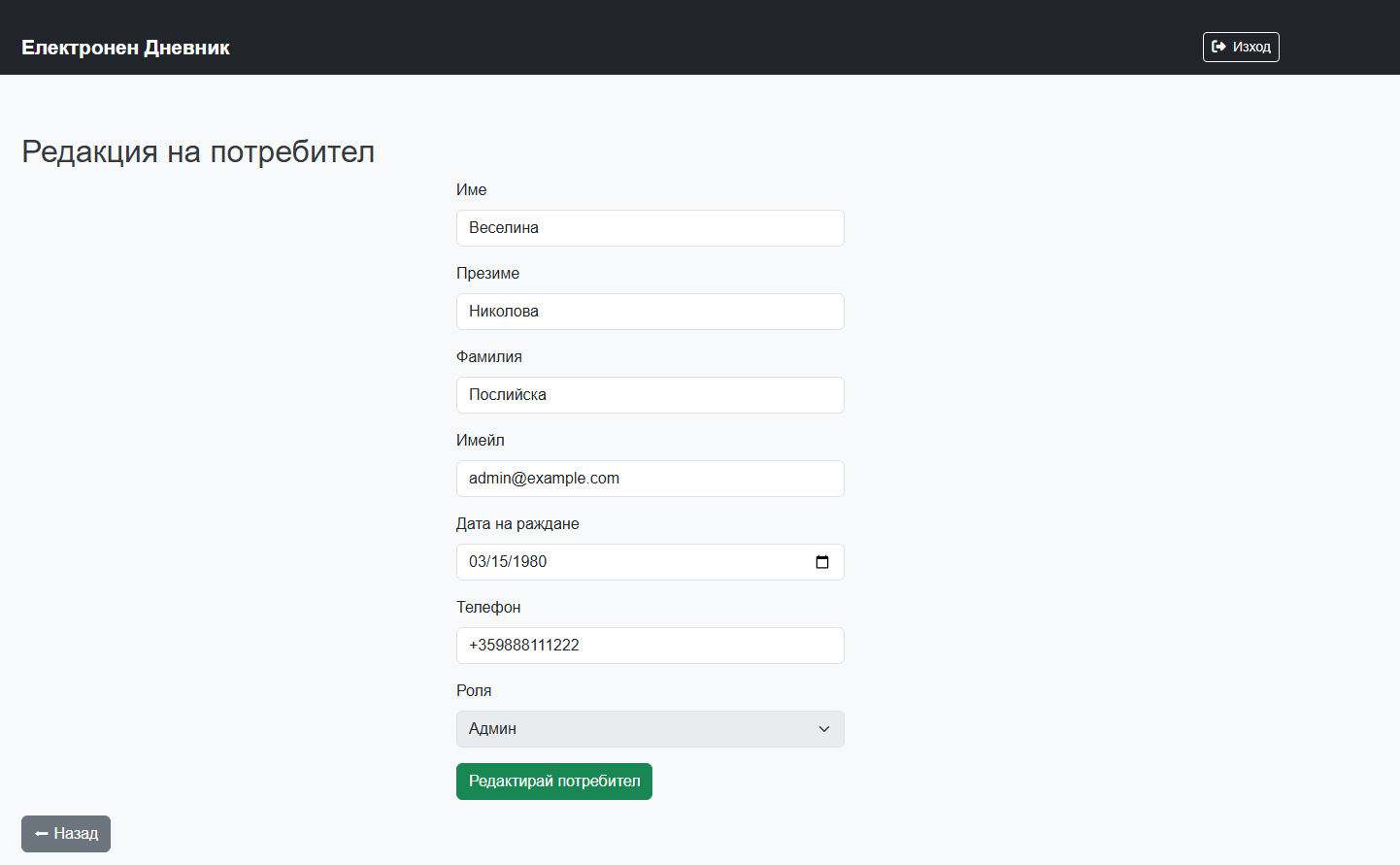
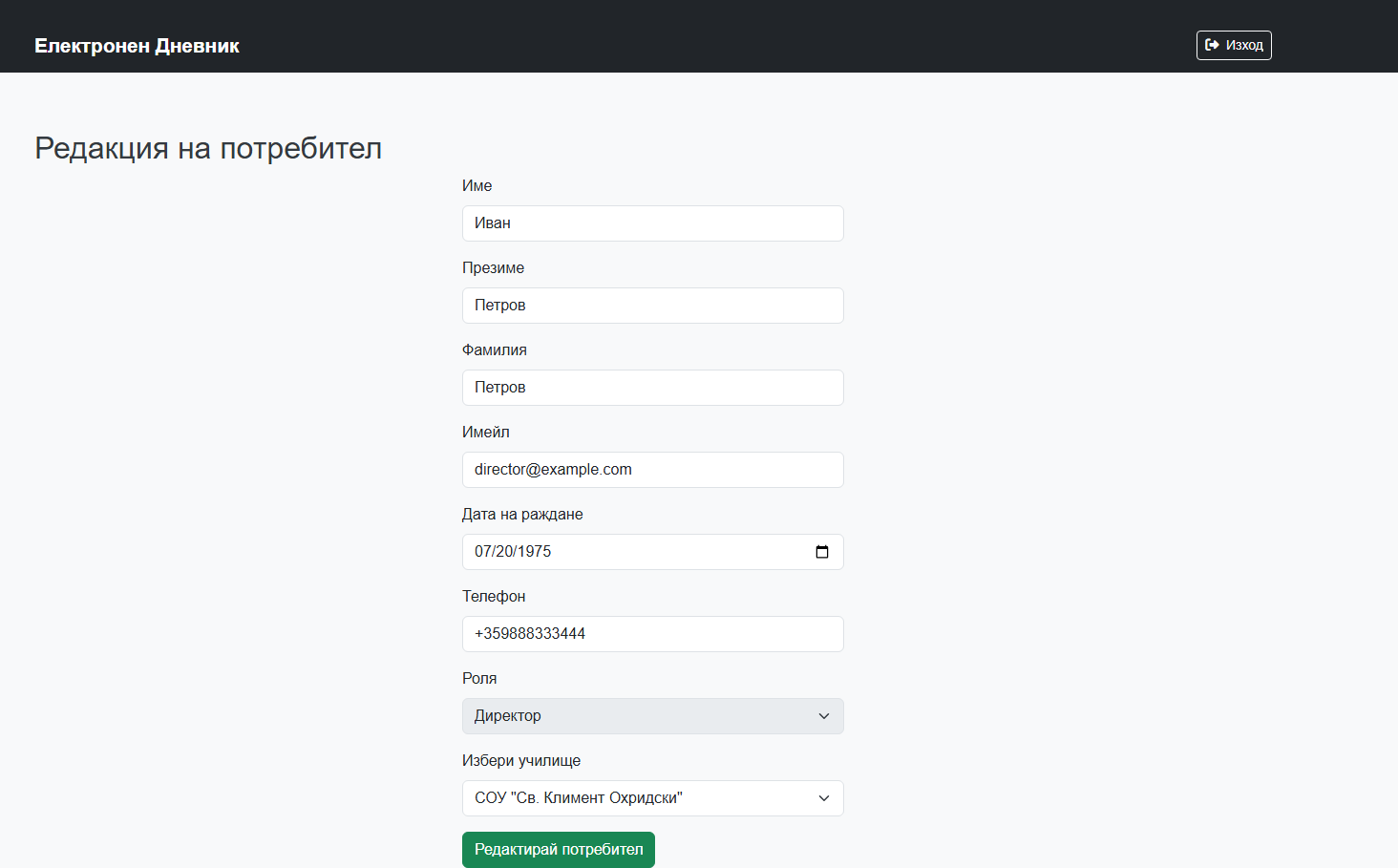


* + 1. **Страница за добавяне на потребител**

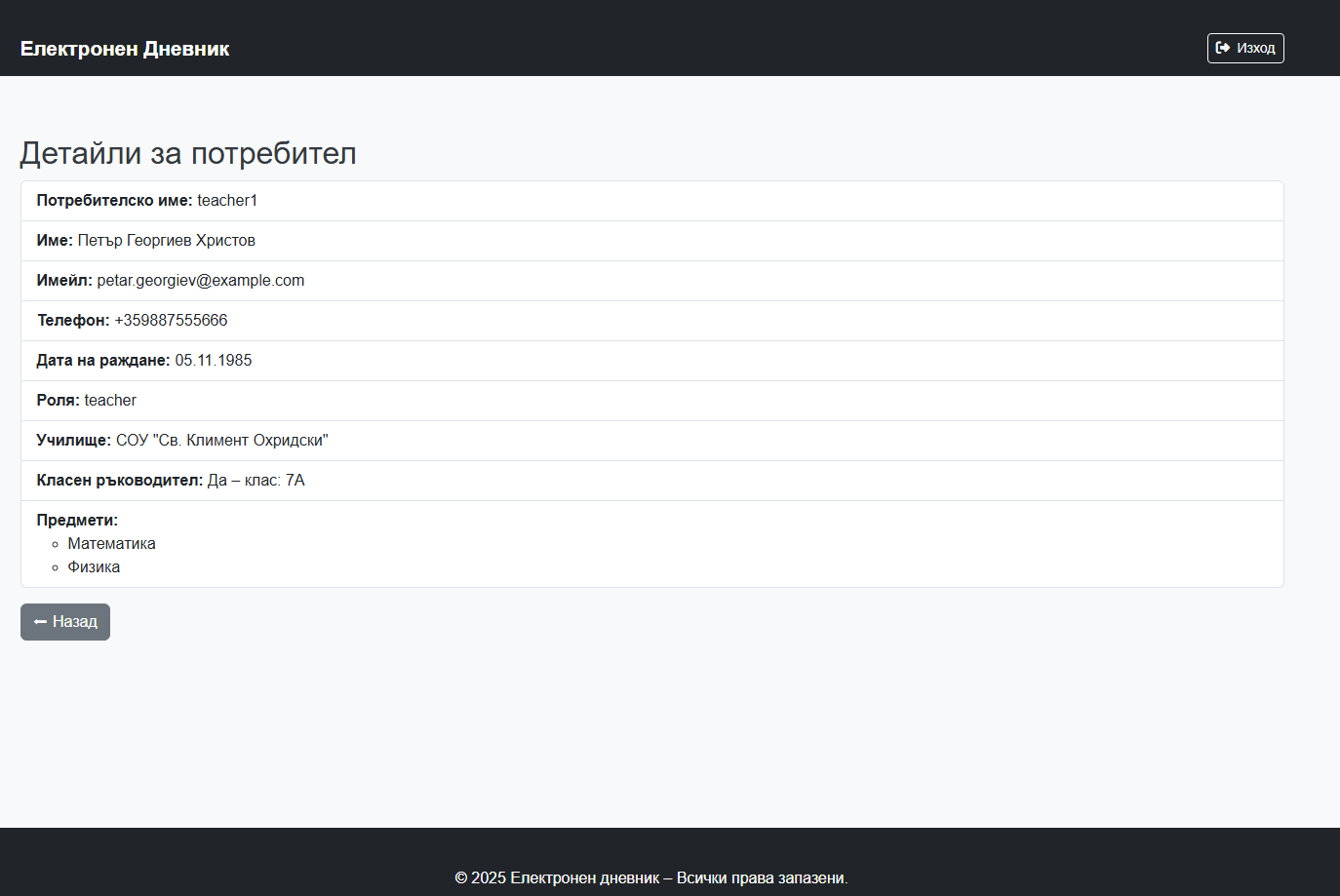
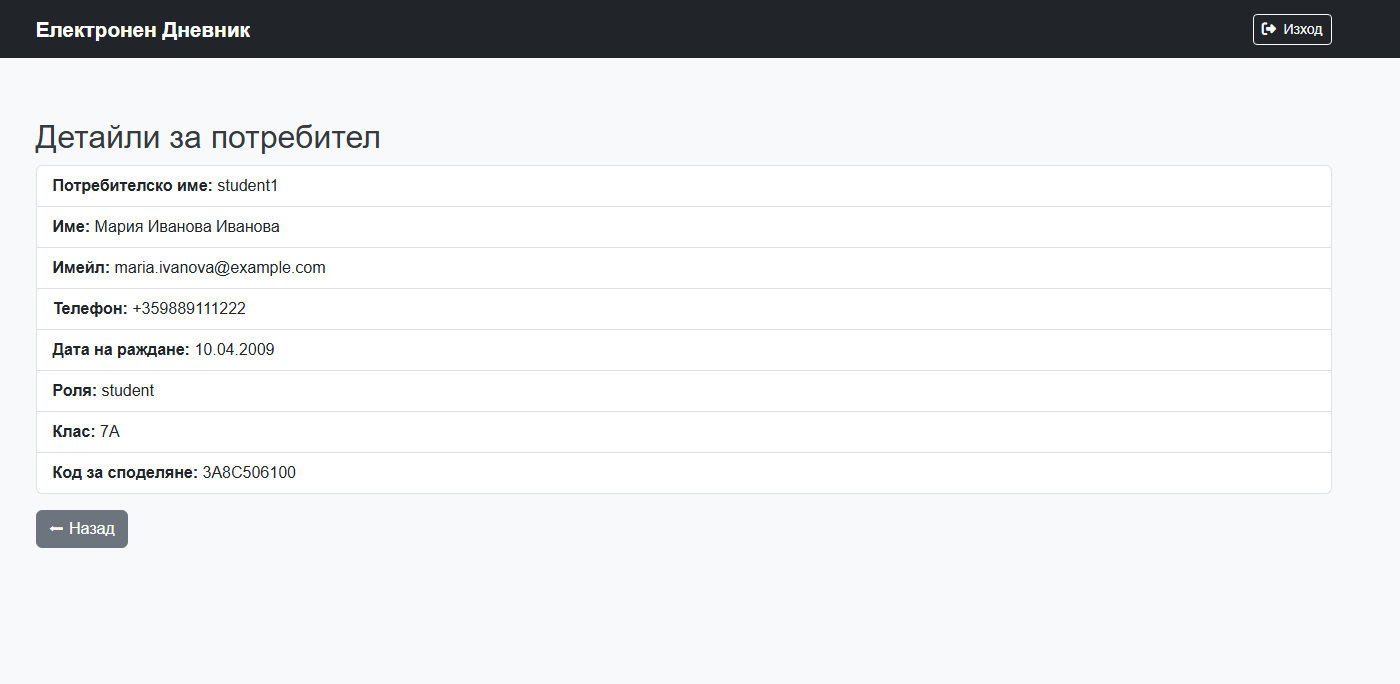
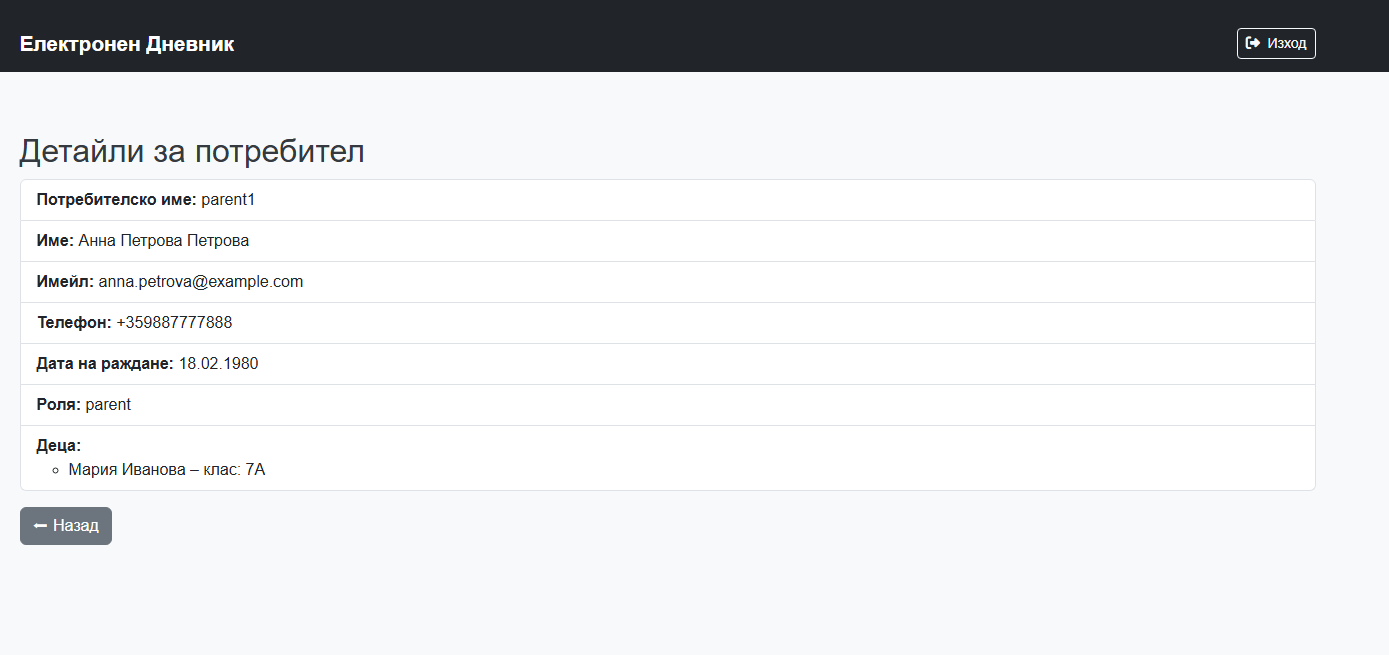
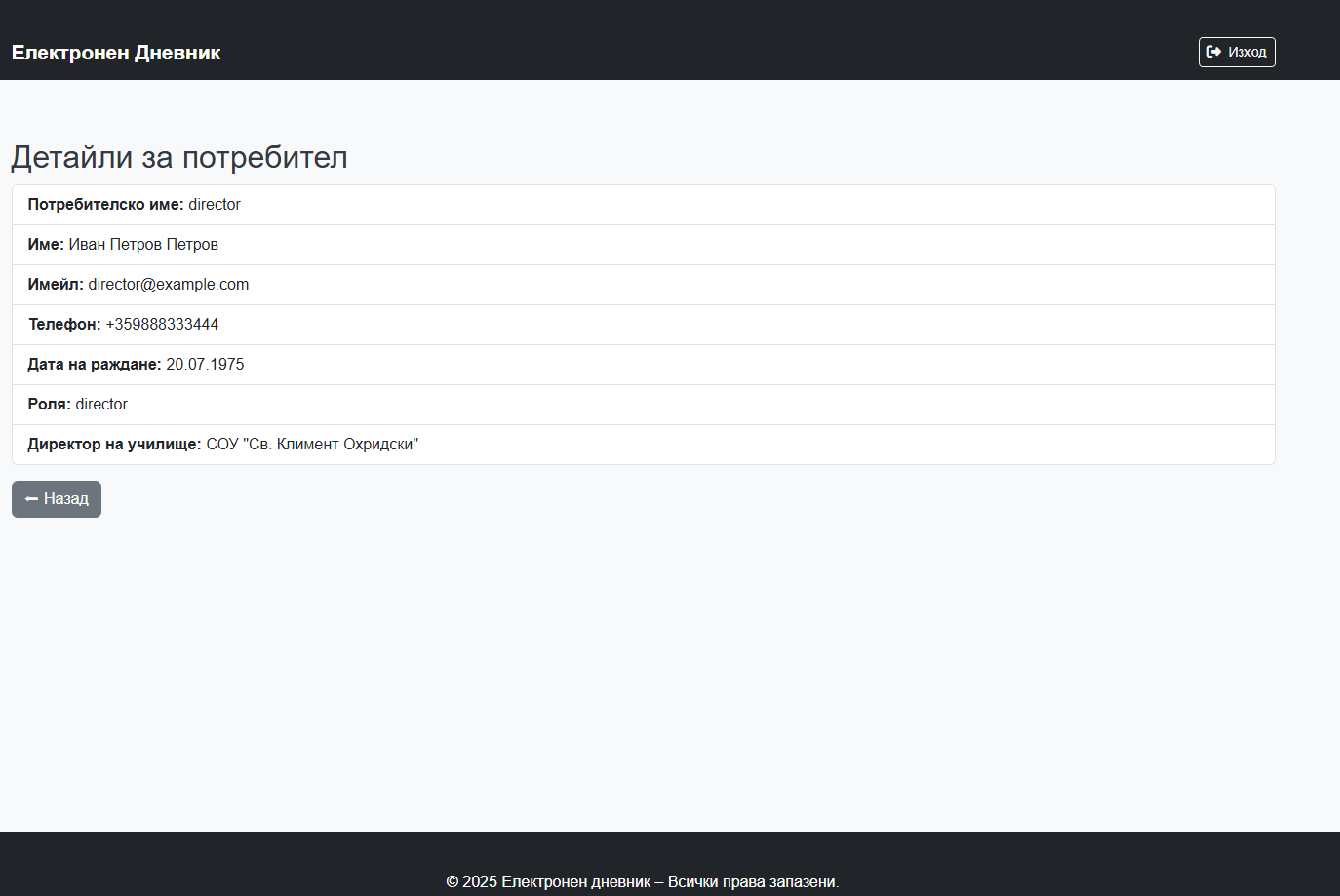
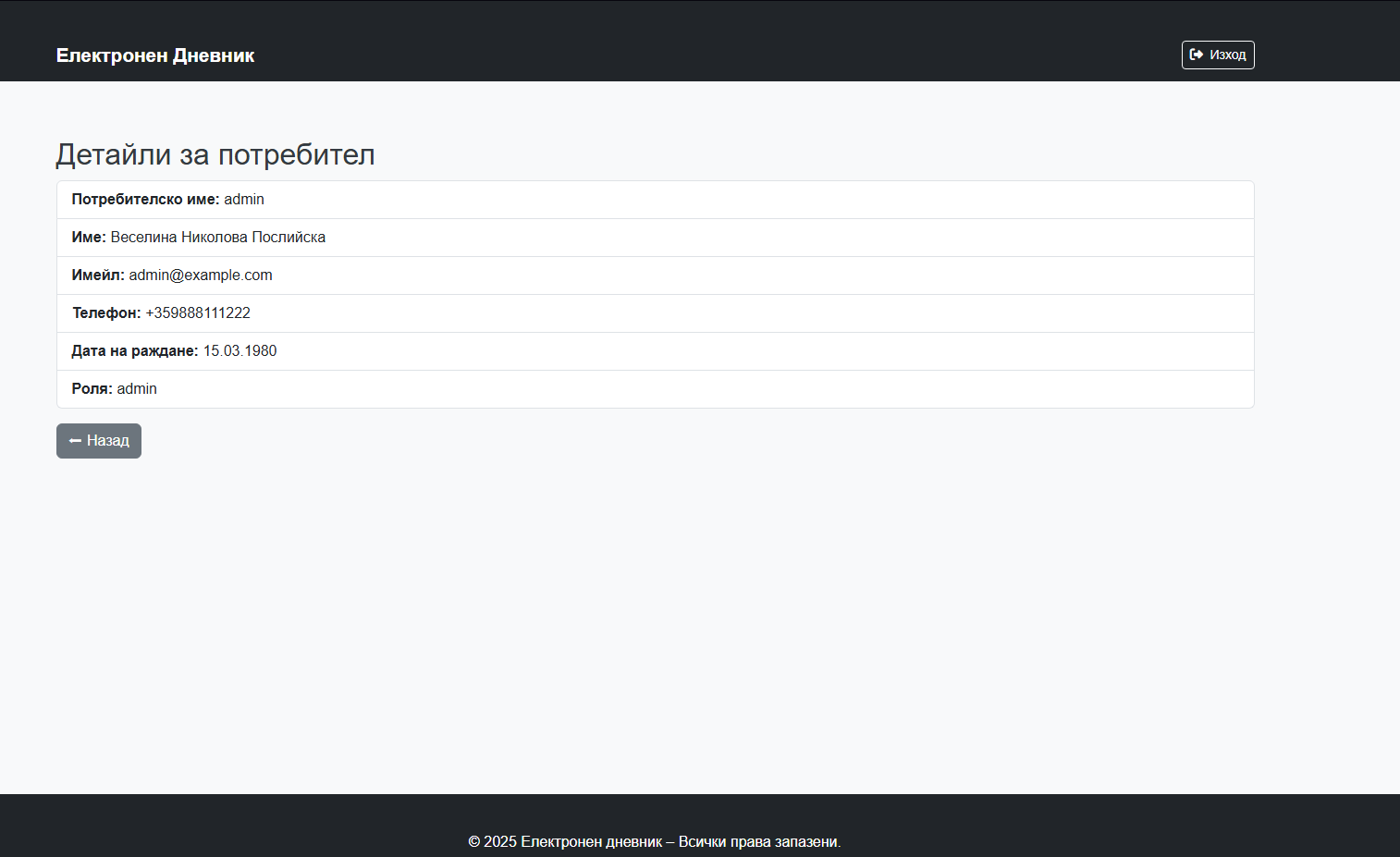


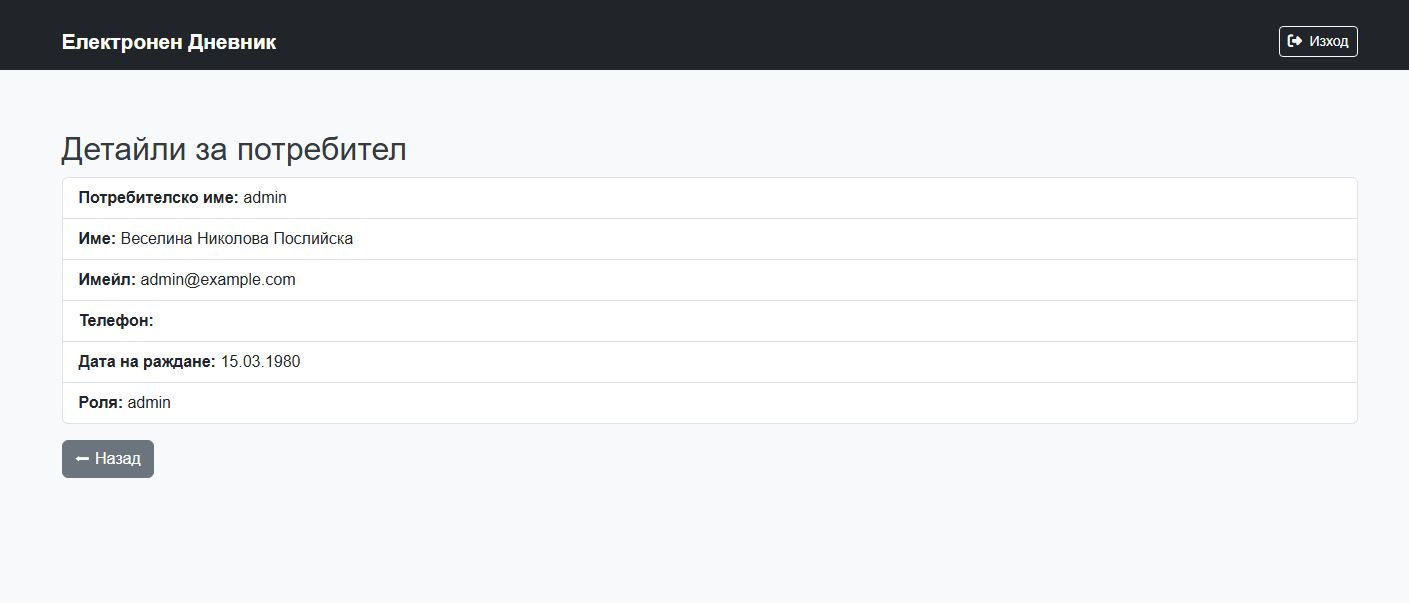
****

1. **Страница за редактиране на потребител**

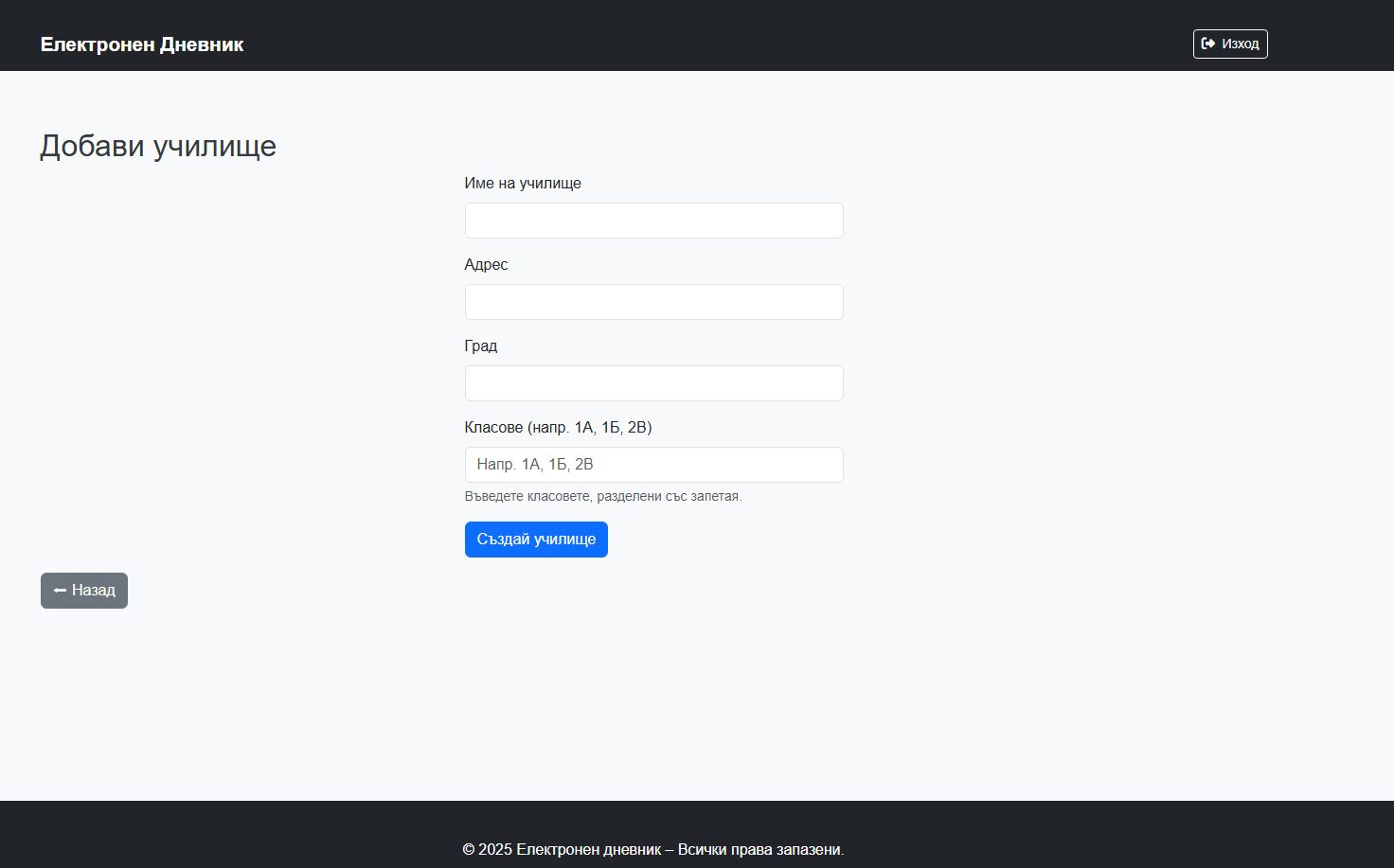


1. **Страница с детайли за потребителя**

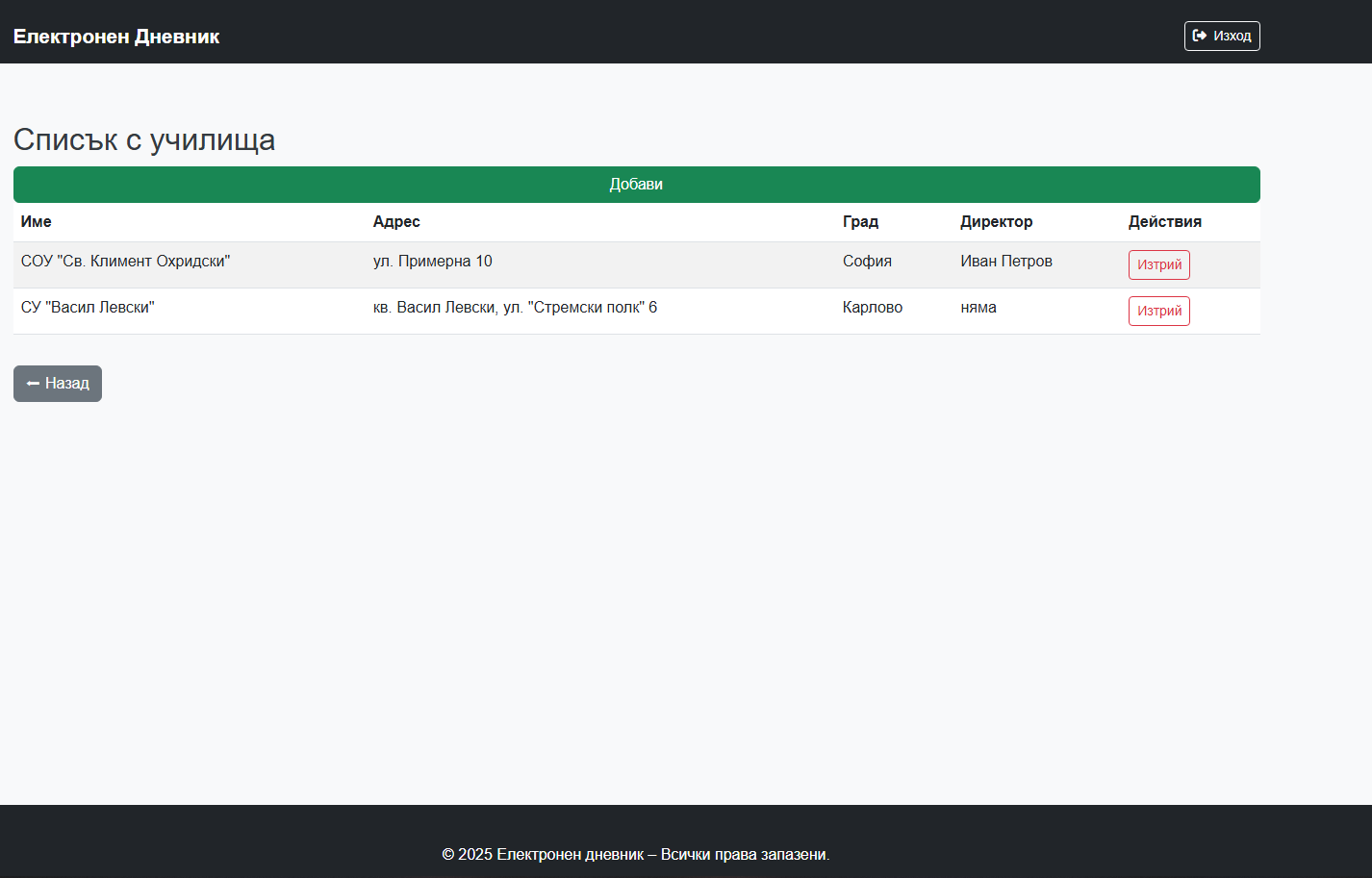
****

****

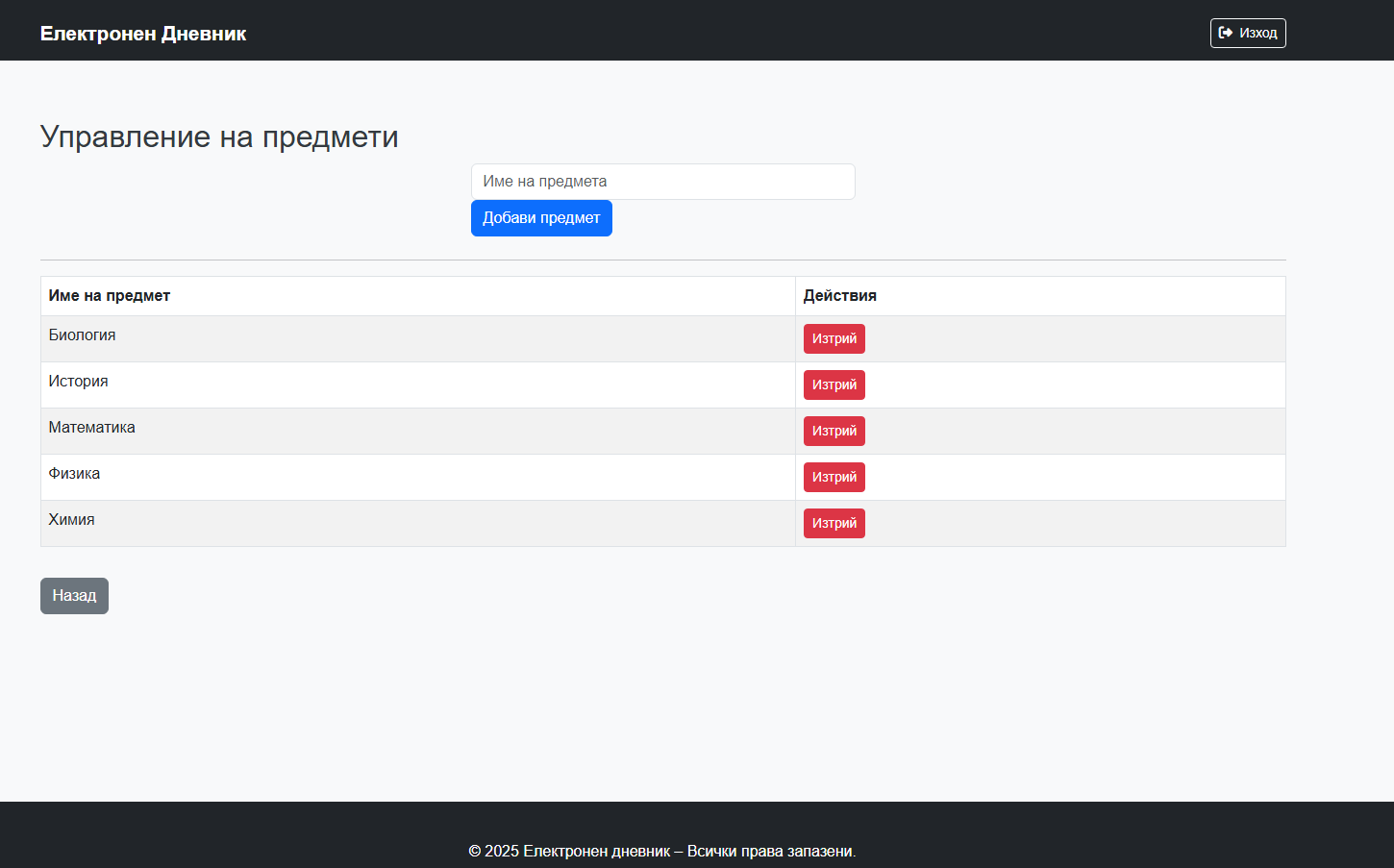
1. **Страница за добавяне на училище**



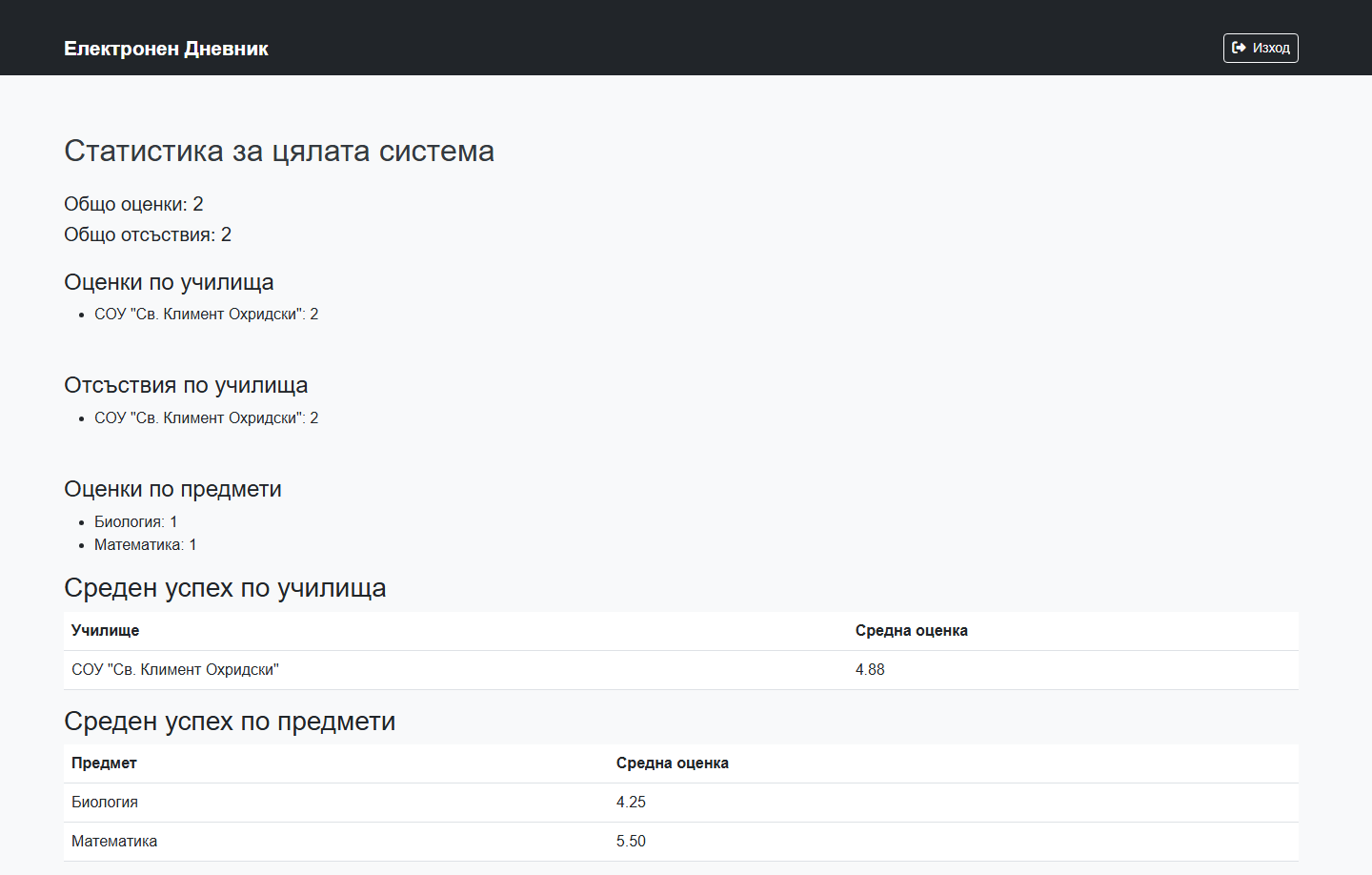
1. **Страница с училища**



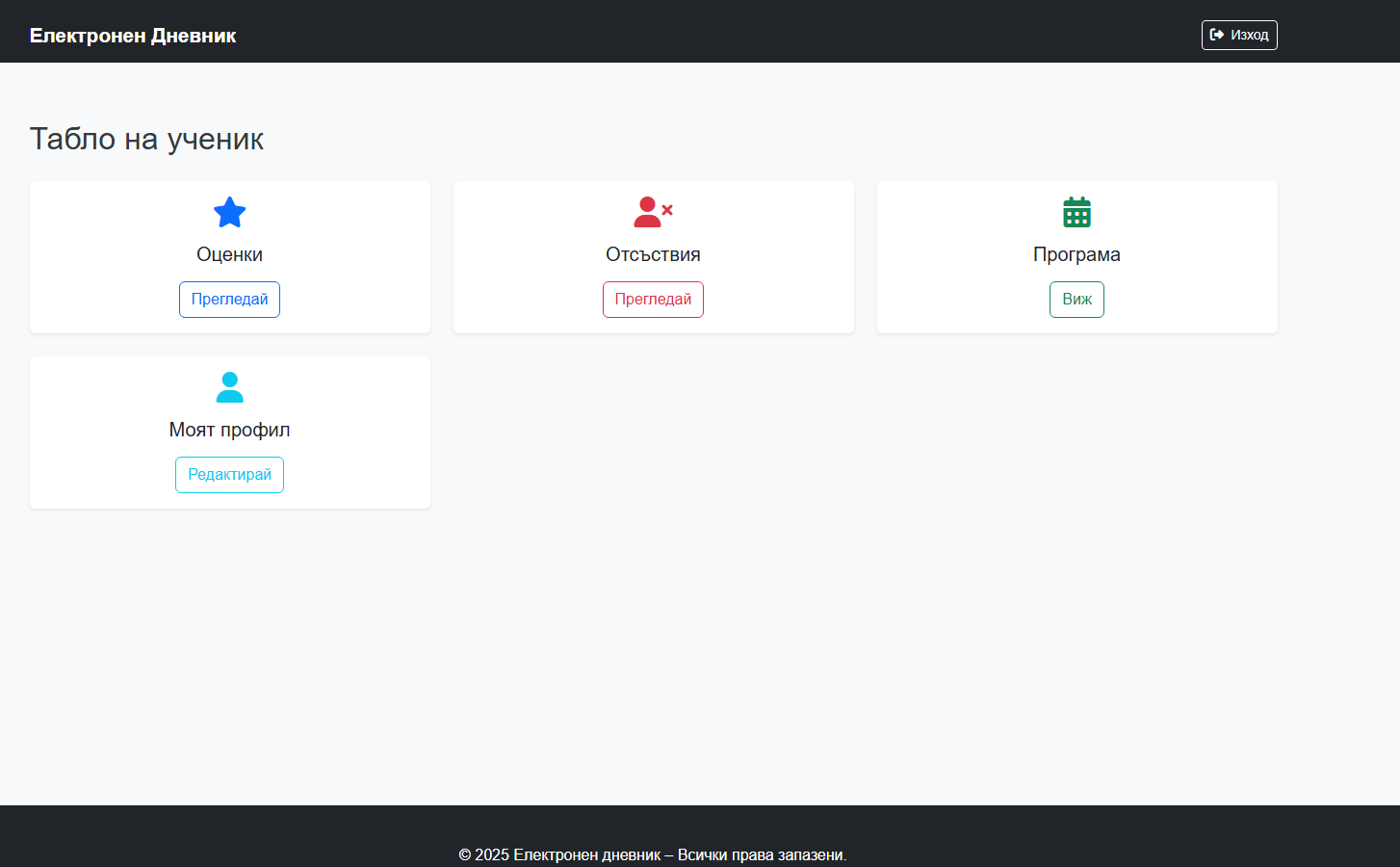
1. **Страница с предмети**



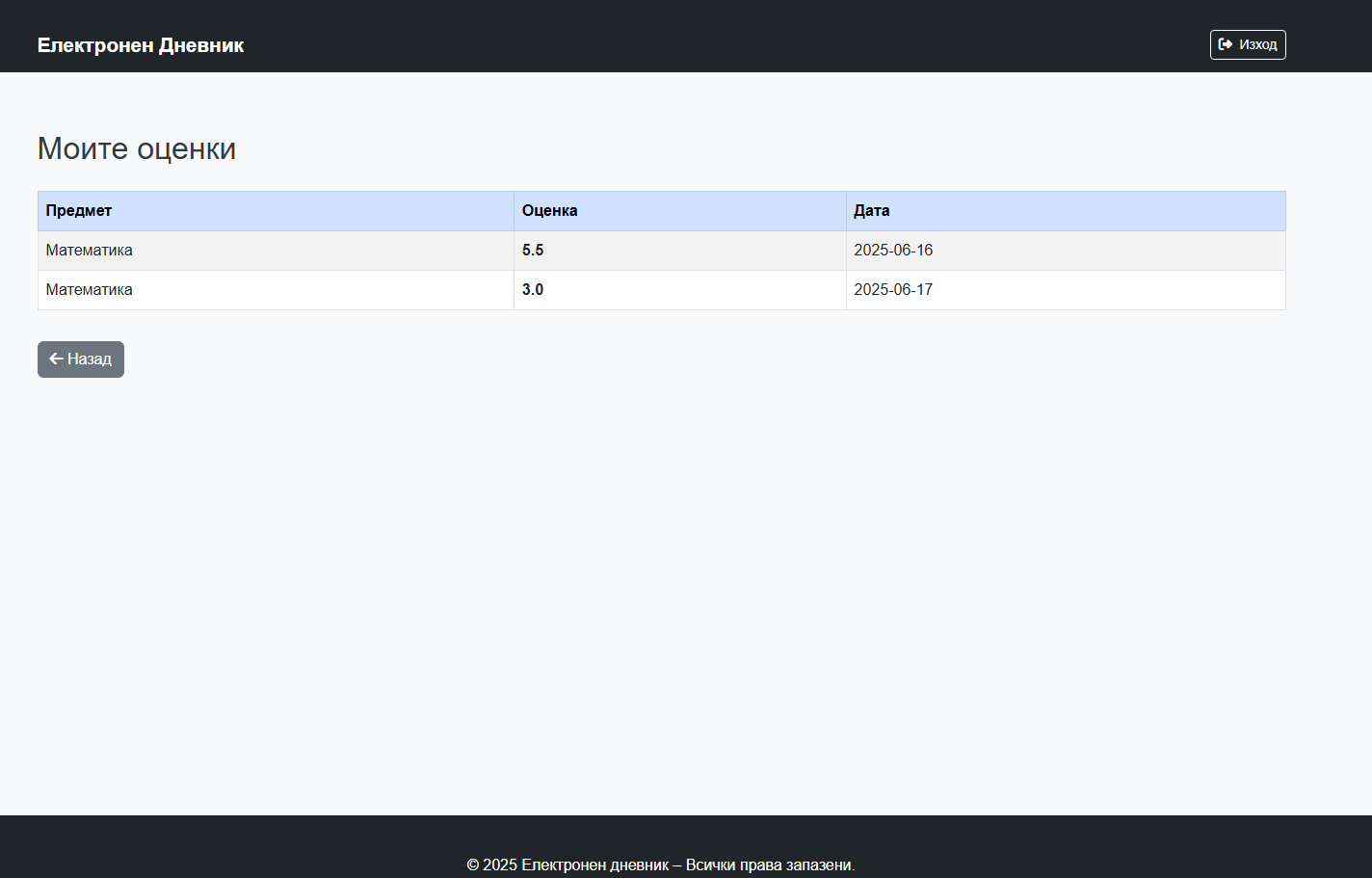
1. **Страница със статистика**



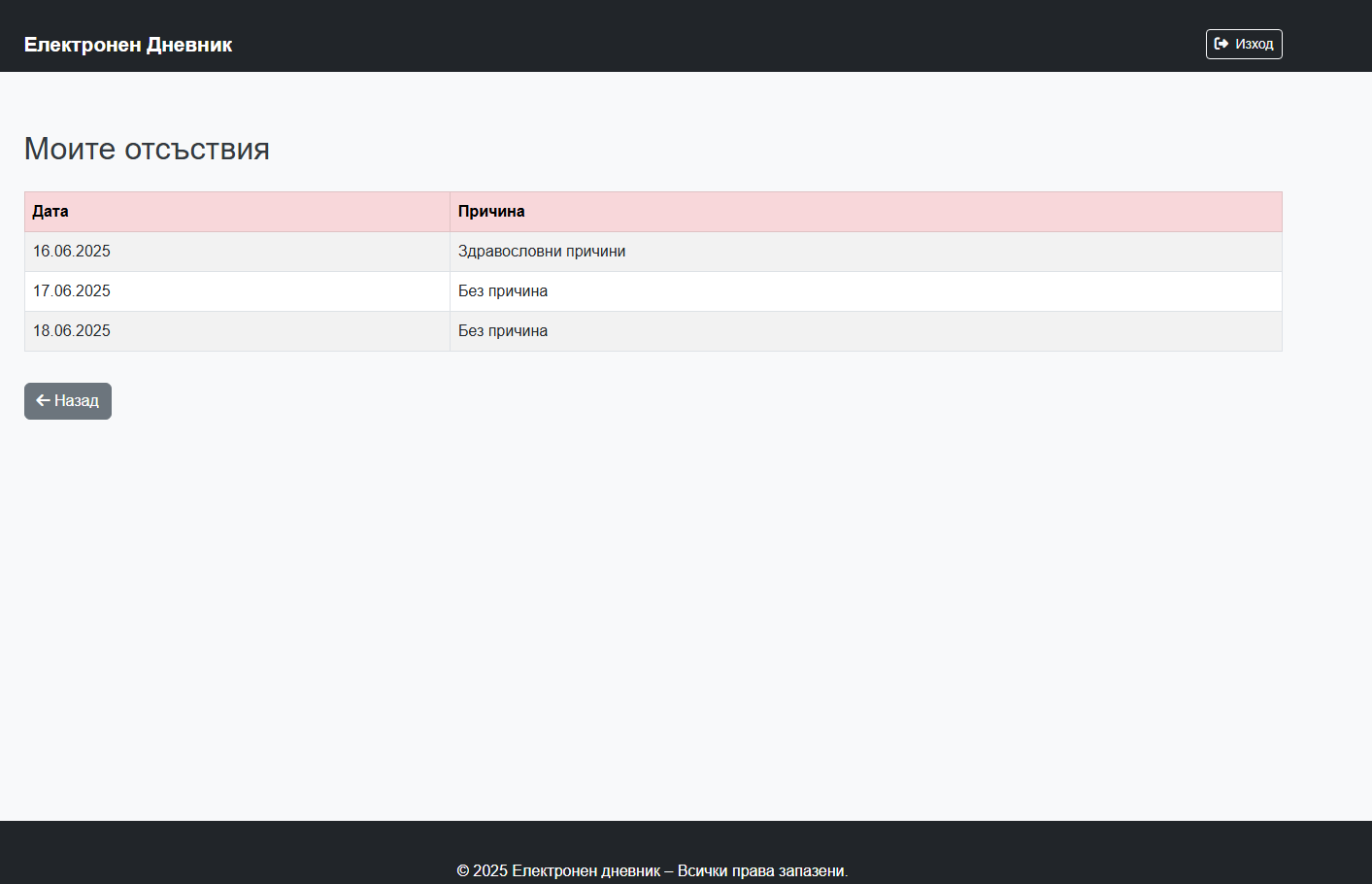
1. **Страници при влизане като студент**
   * 1. **Заглавна страница**



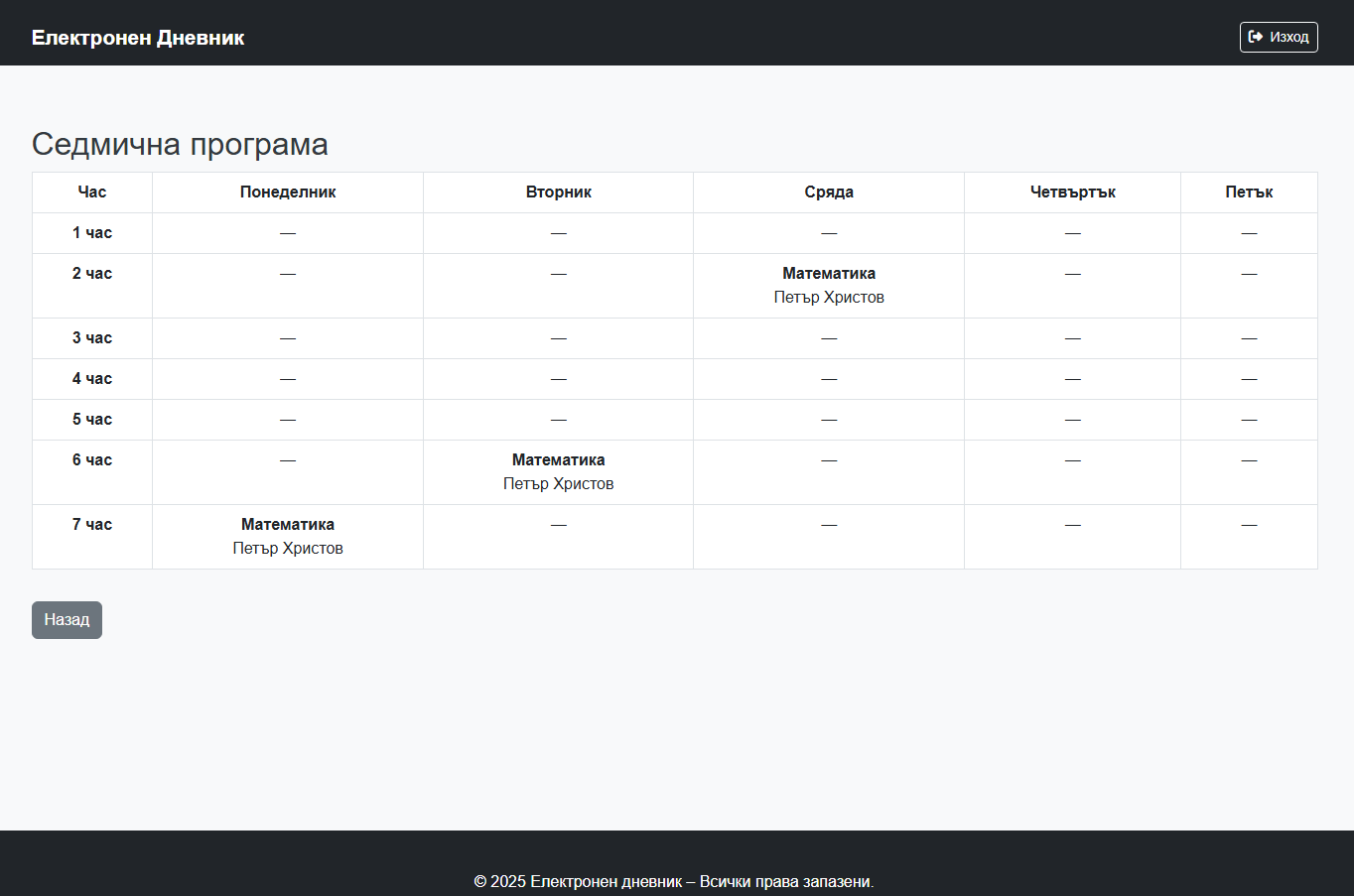
* + 1. **Списък с оценки**



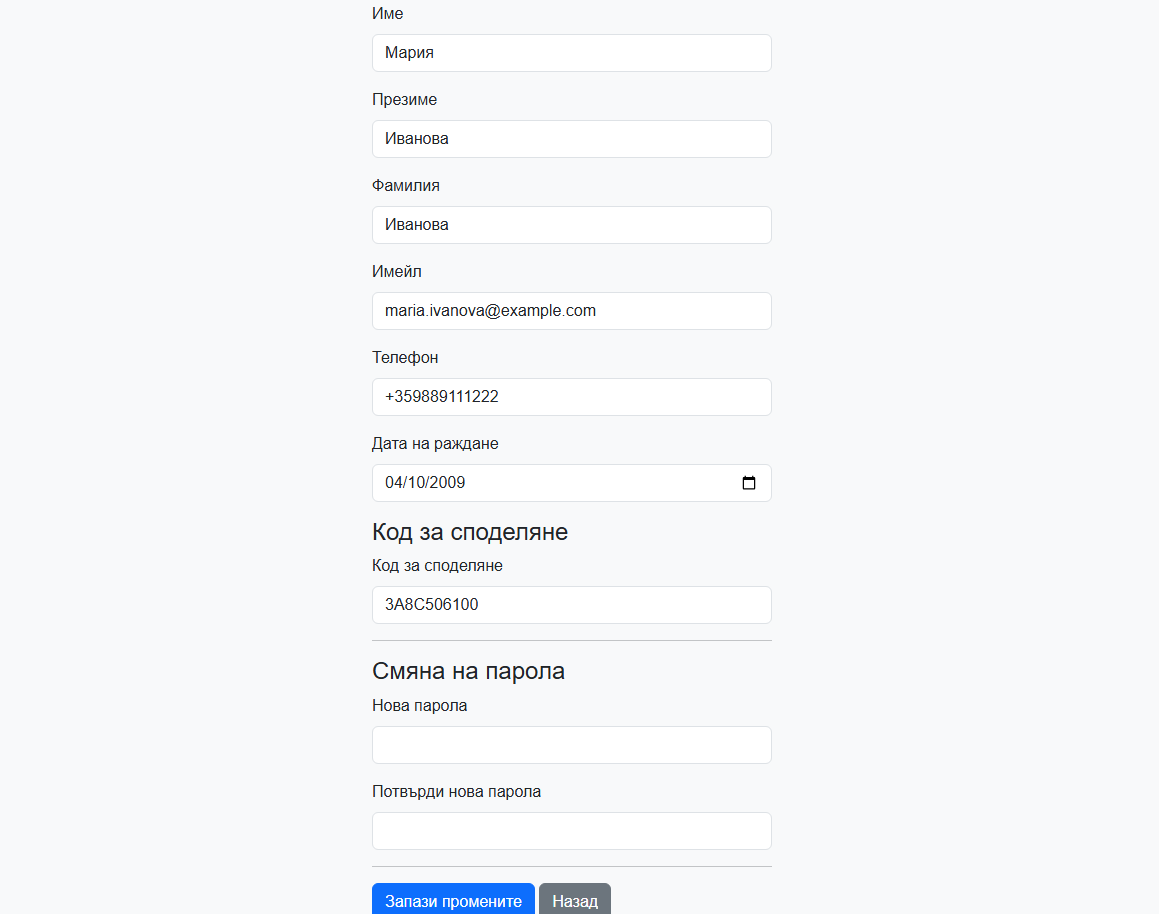
* + 1. **Списък с отсъствия**



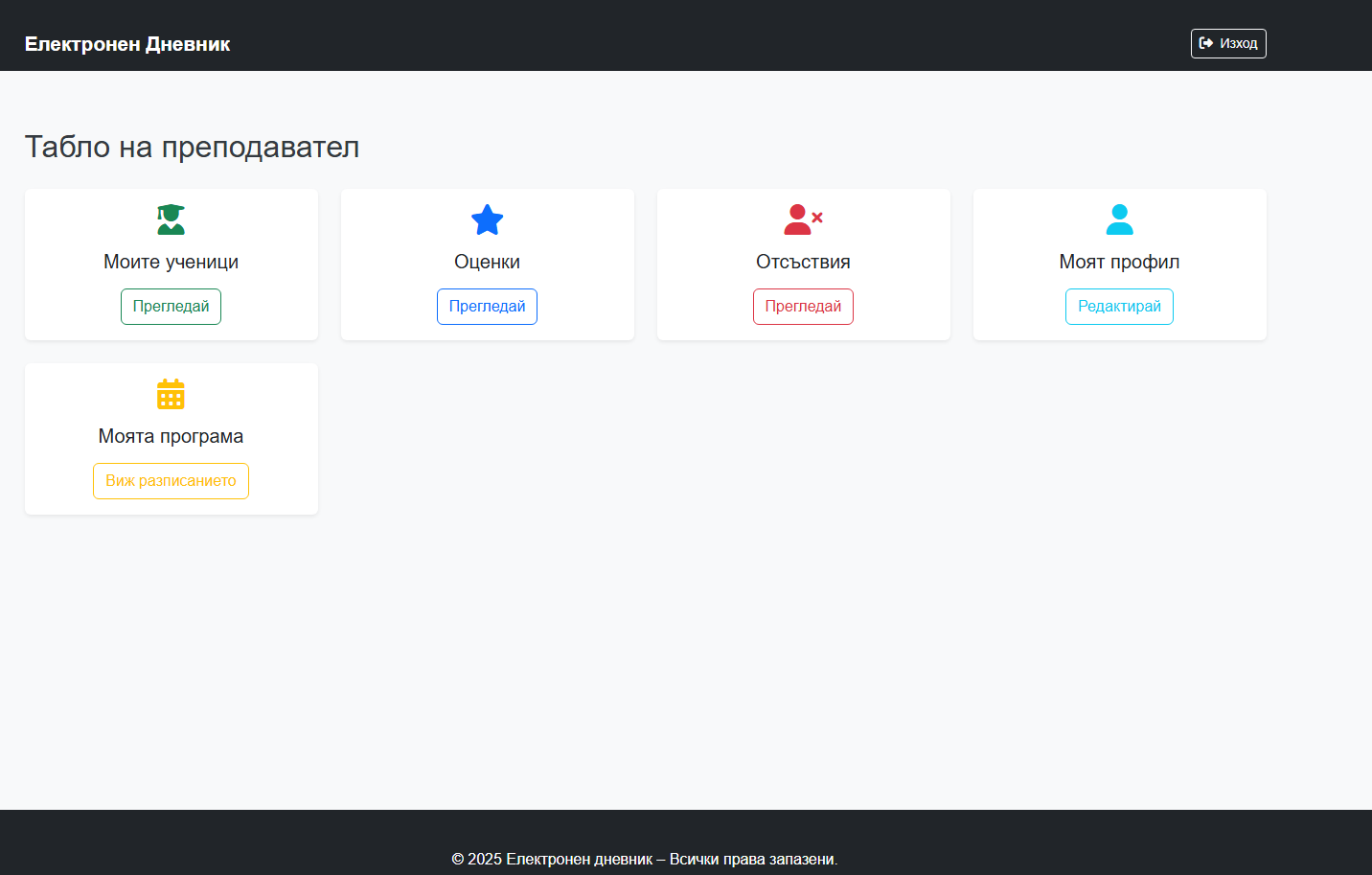
1. **Учебна програма**



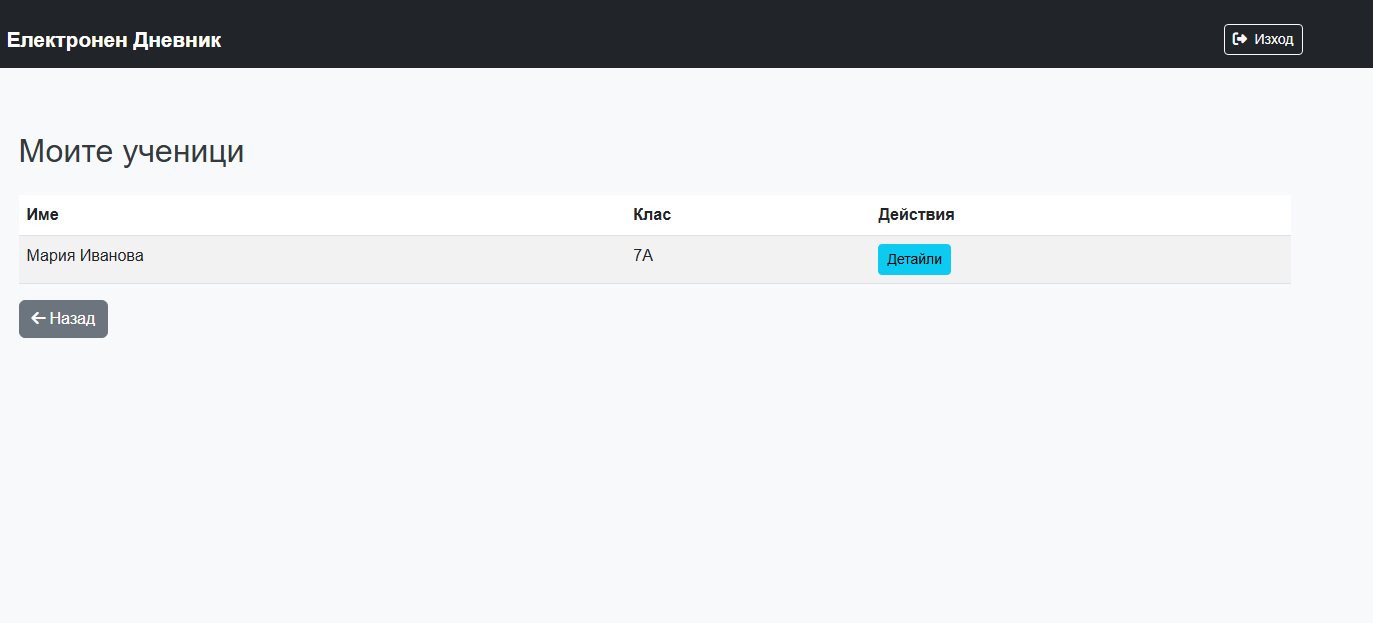
1. **Личен профил**

****

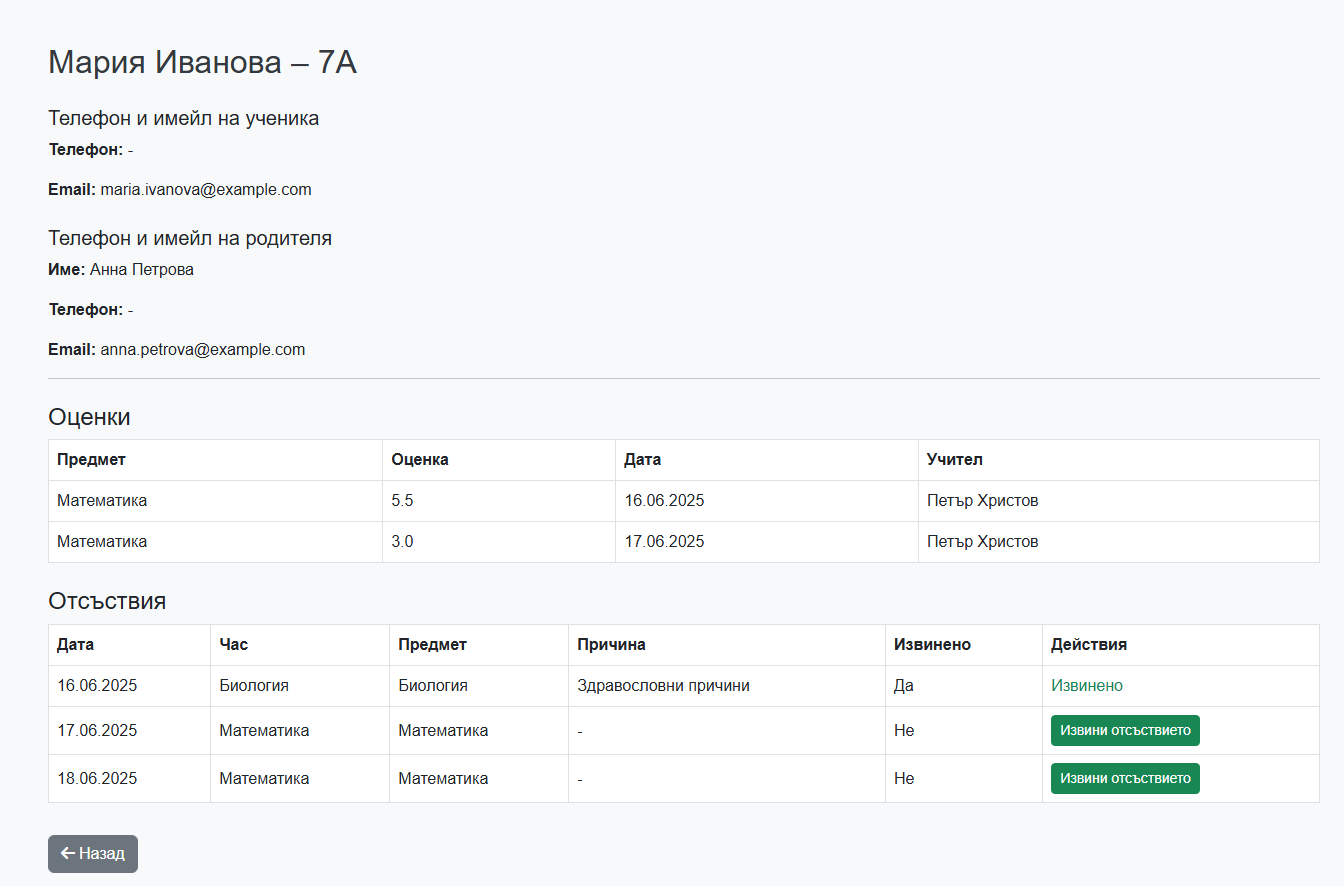
1. **Страници при влизане като учител**
   * 1. **Заглавна страница**



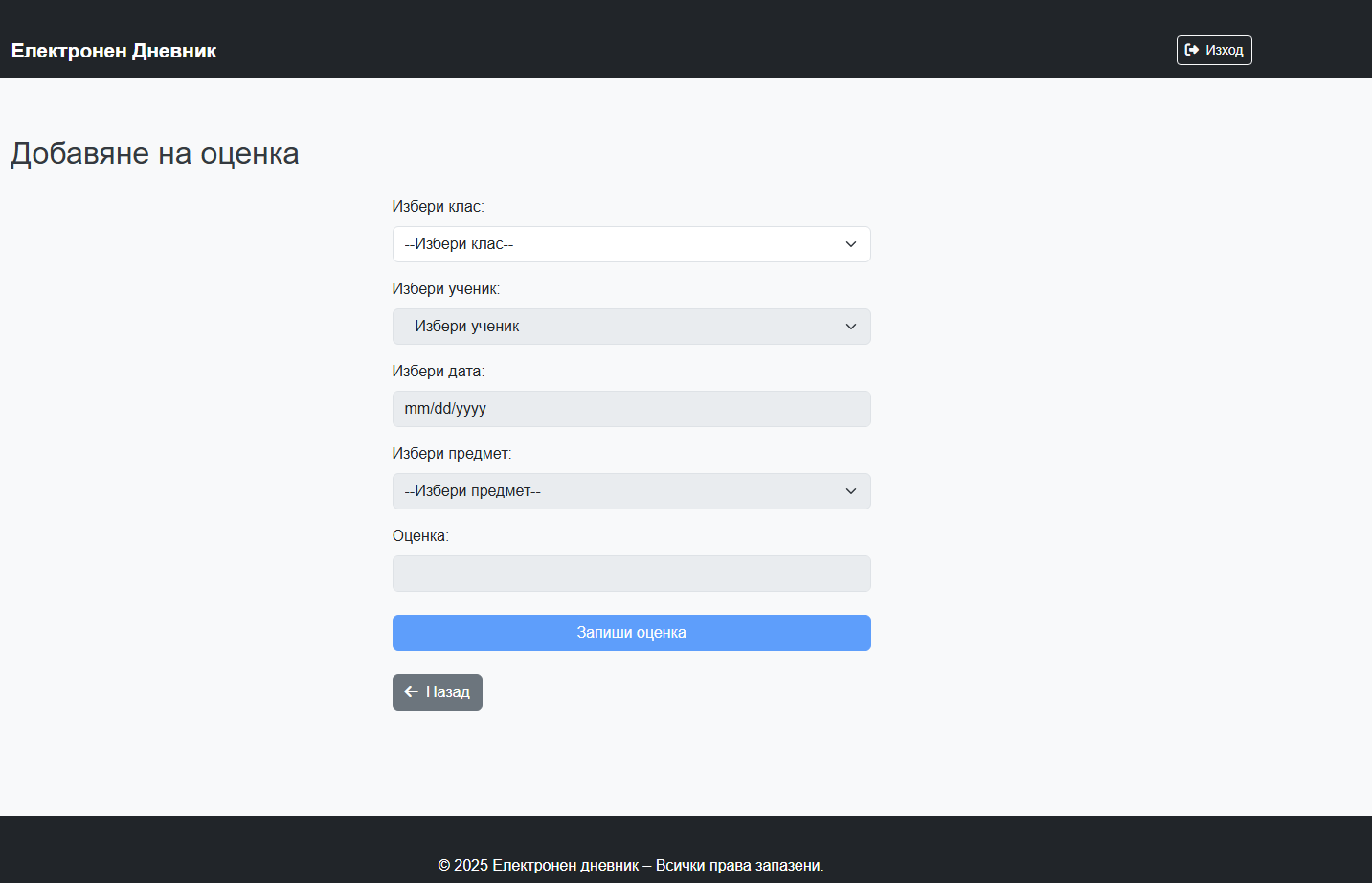
* + 1. **Страница със списък на всички ученици от неговия клас**



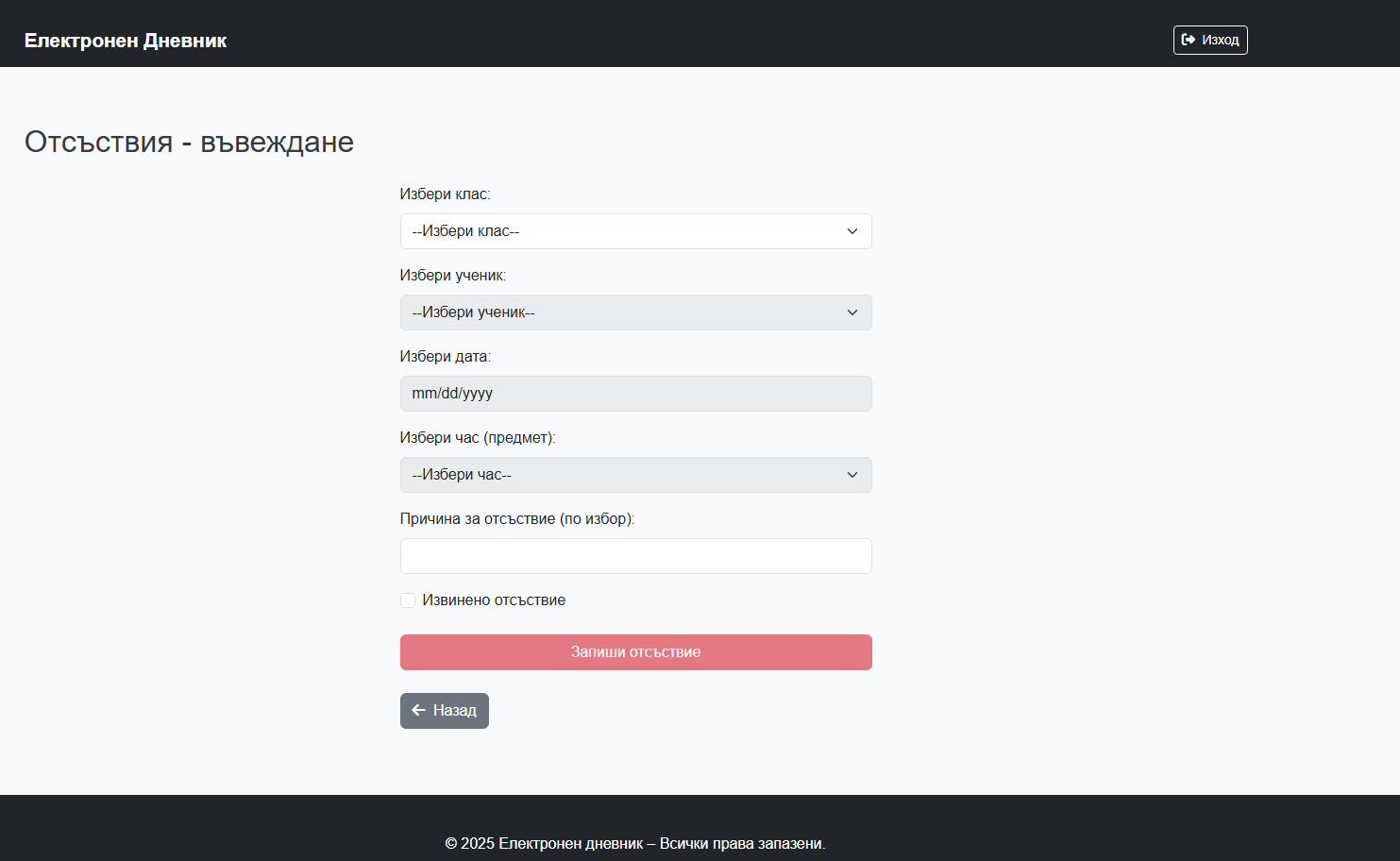
1. **Страница с детайли за ученика**



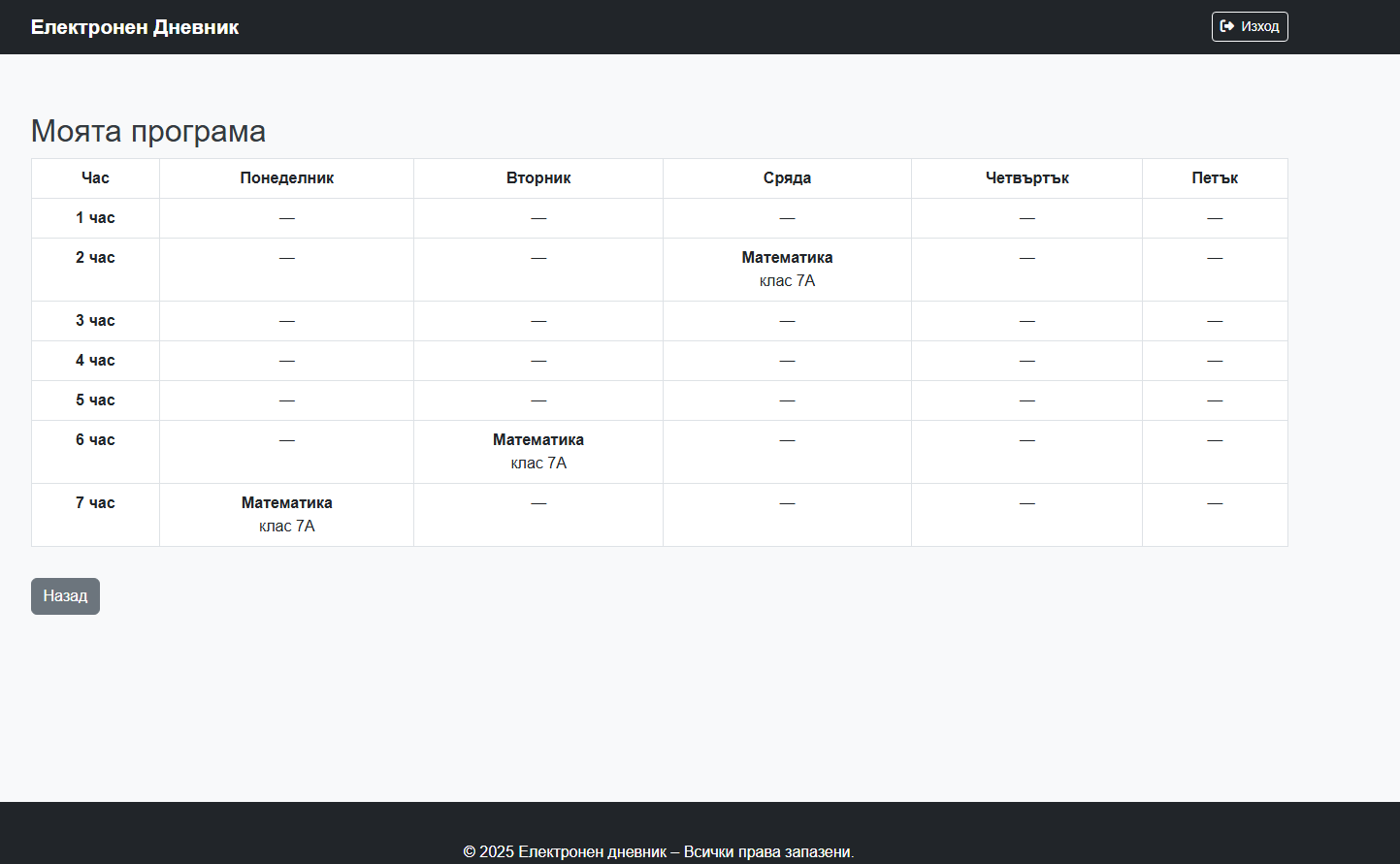
1. **Страница за добавяне на оценка**



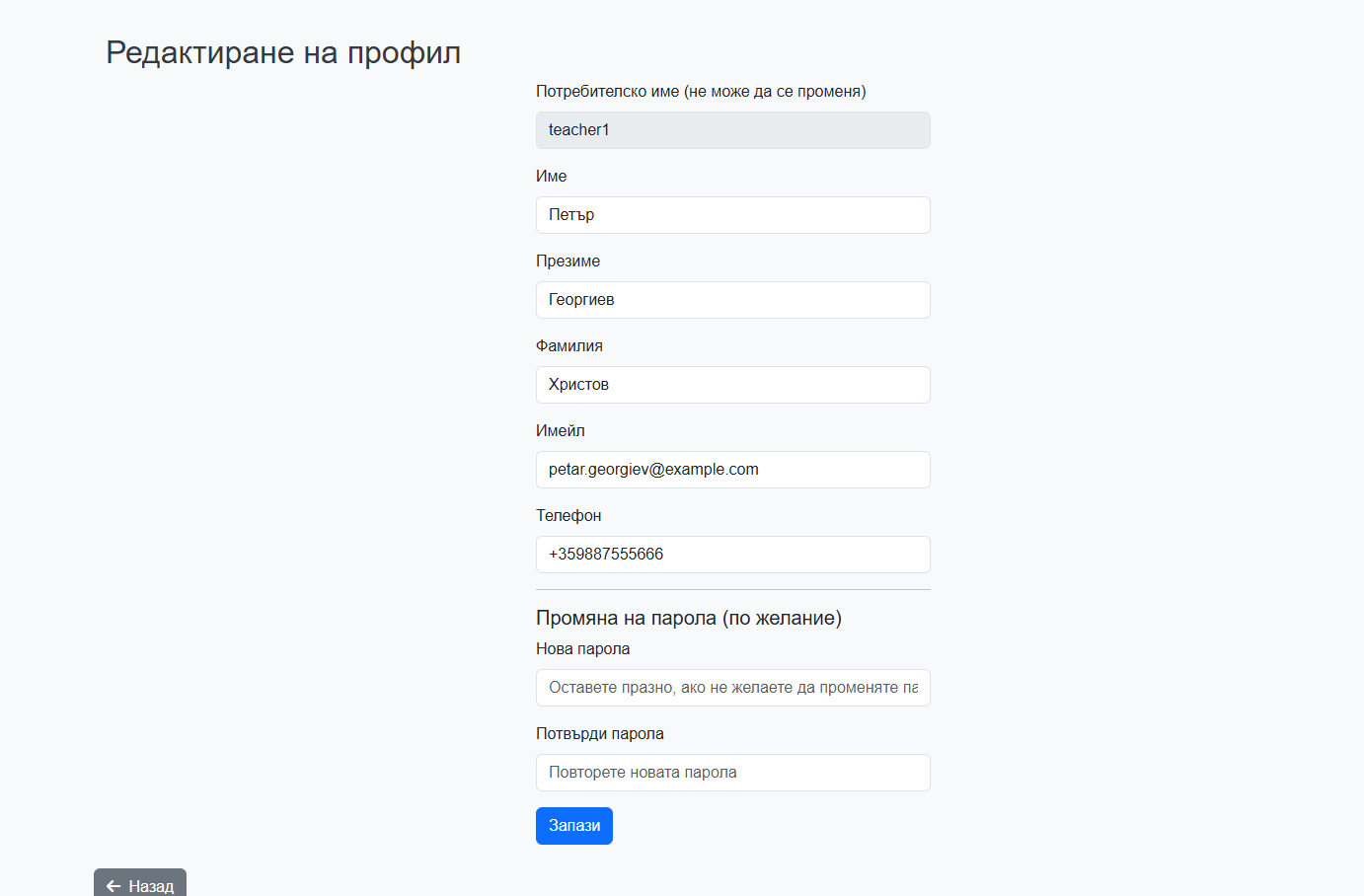
1. **Страница за добавяне на отсъствие**



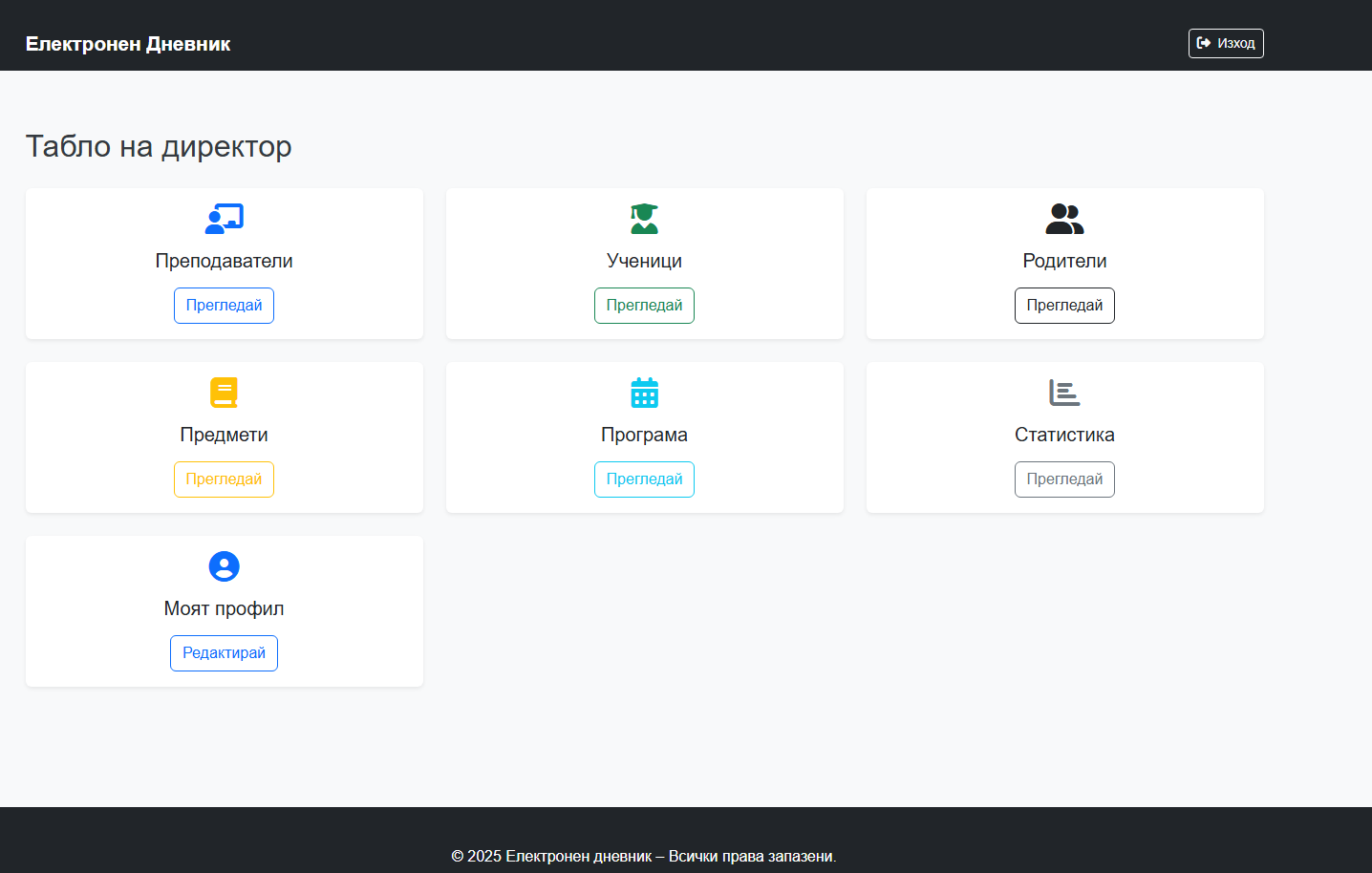
1. **Страница с програмата на преподавателя**



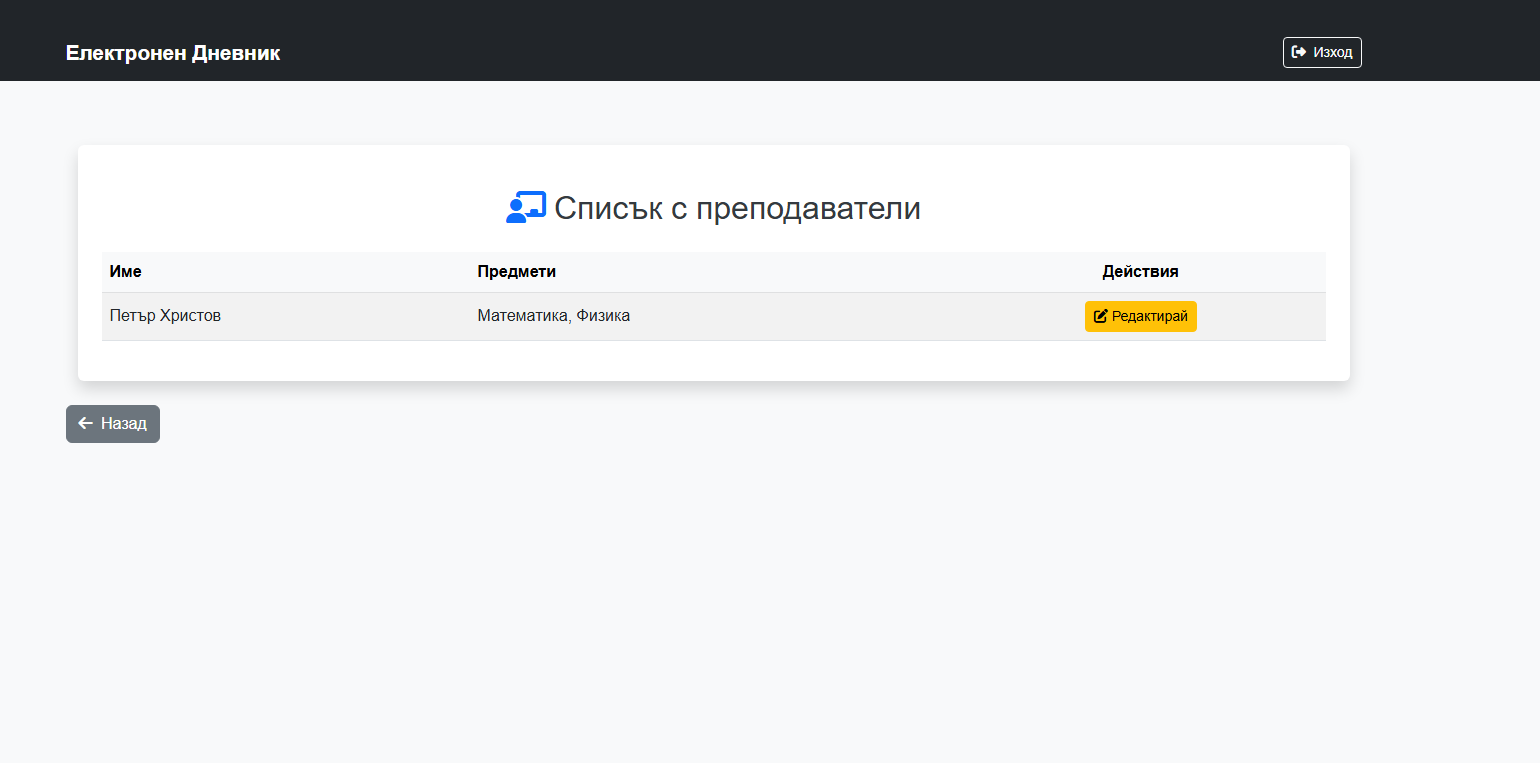
1. **Личен профил на преподавателя**

****

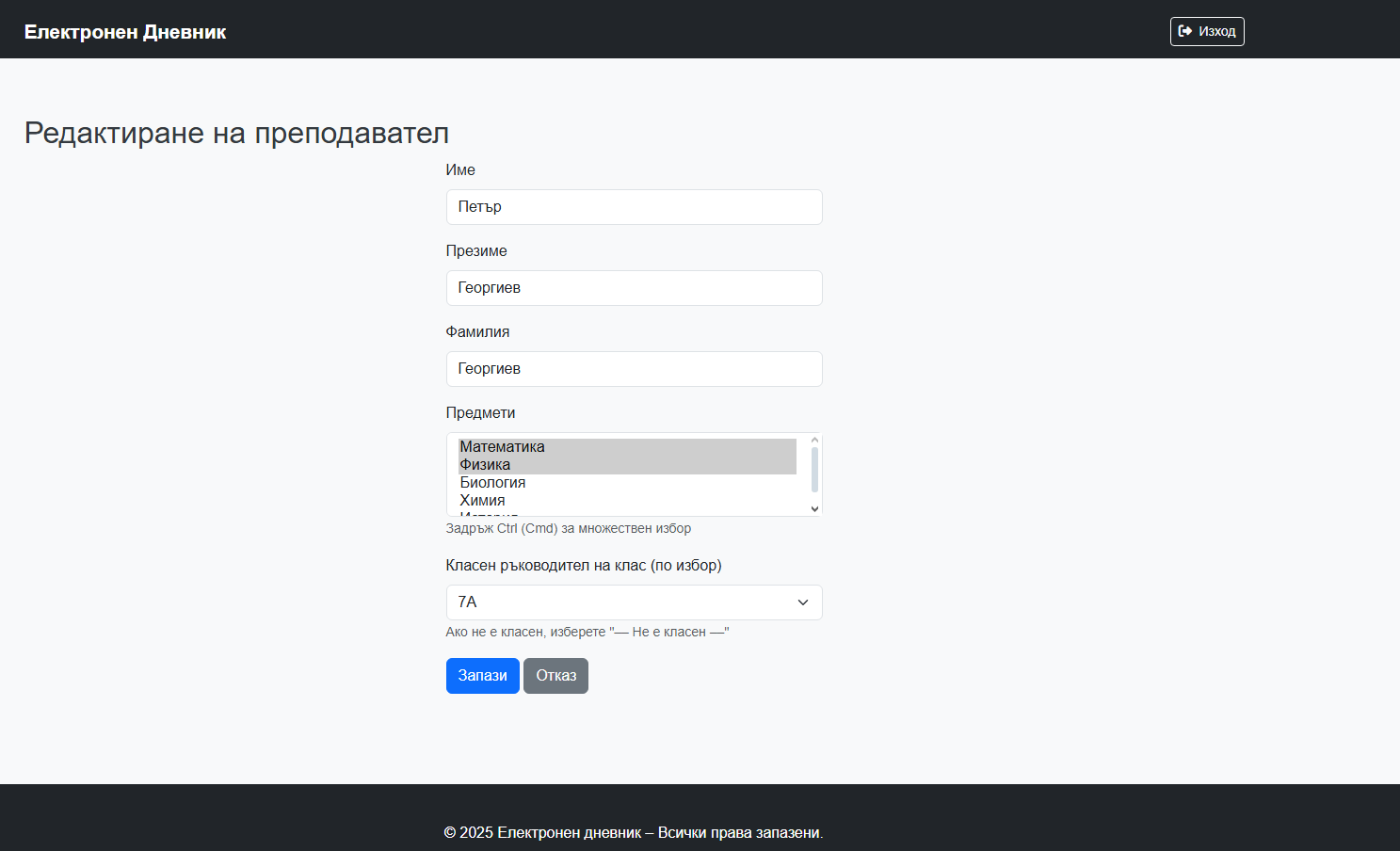
1. **Страници при влизане като директор**
   * 1. **Заглавна страница**

****

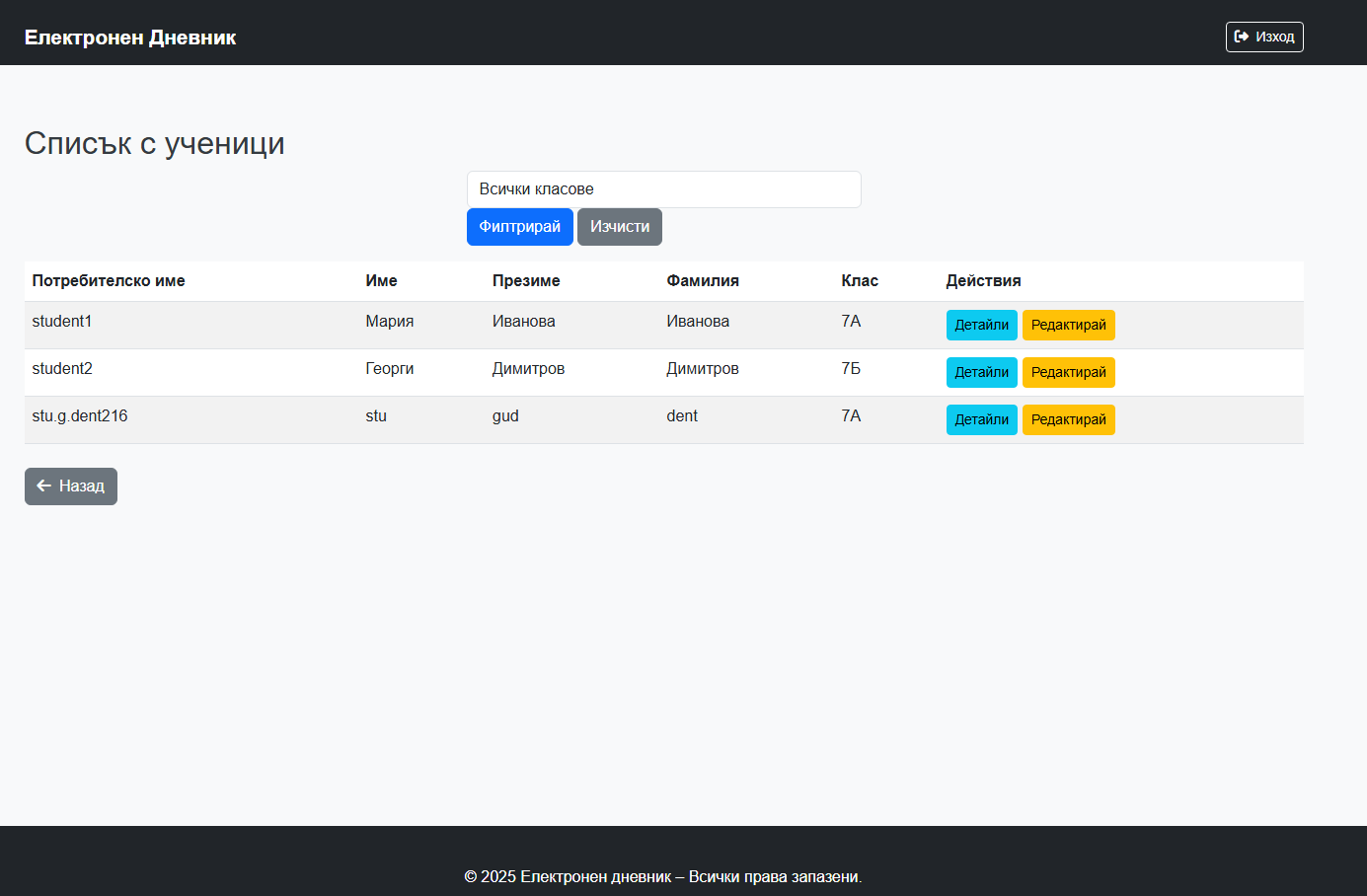
1. **Списък с преподаватели**

****

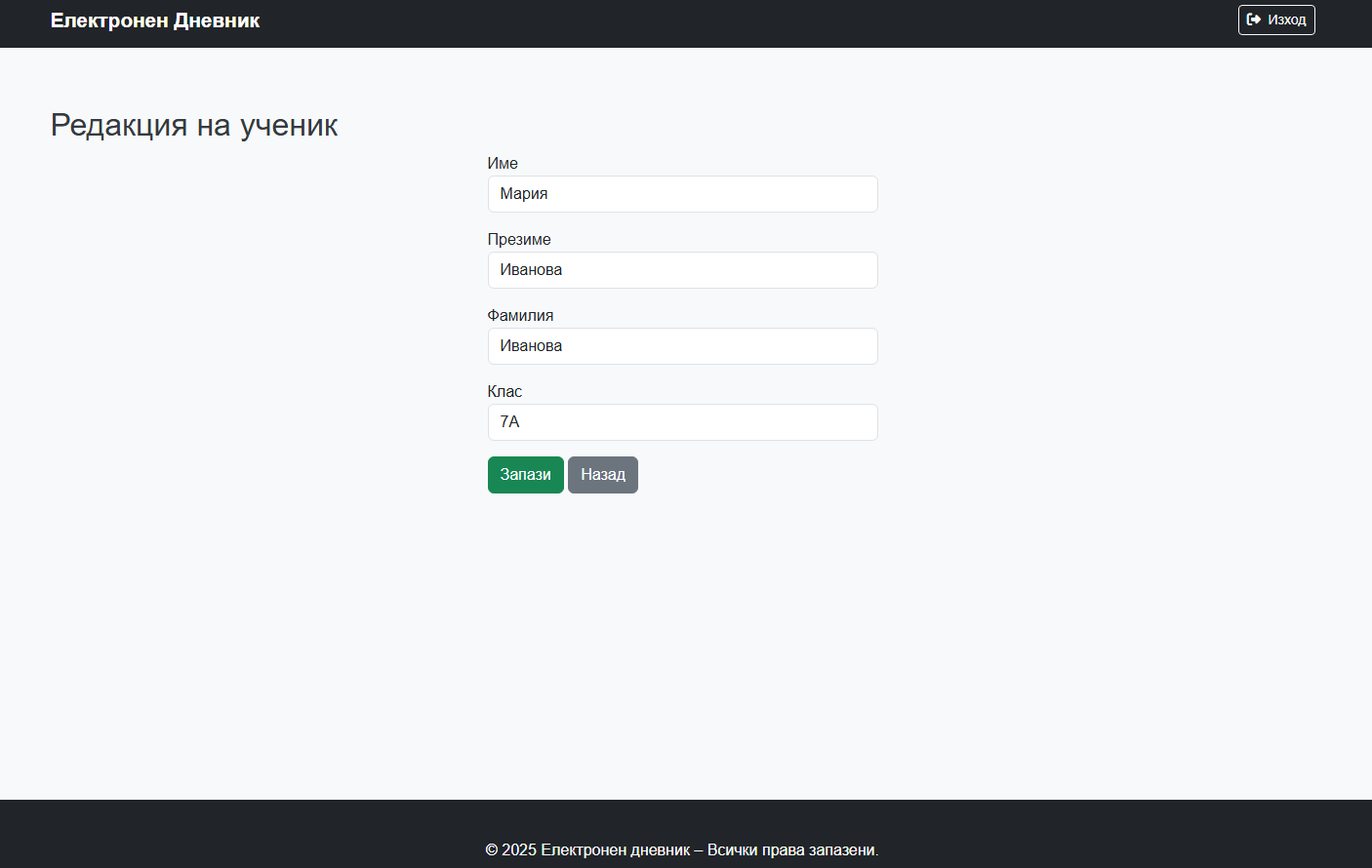
1. **Редактиране на преподавател**

****

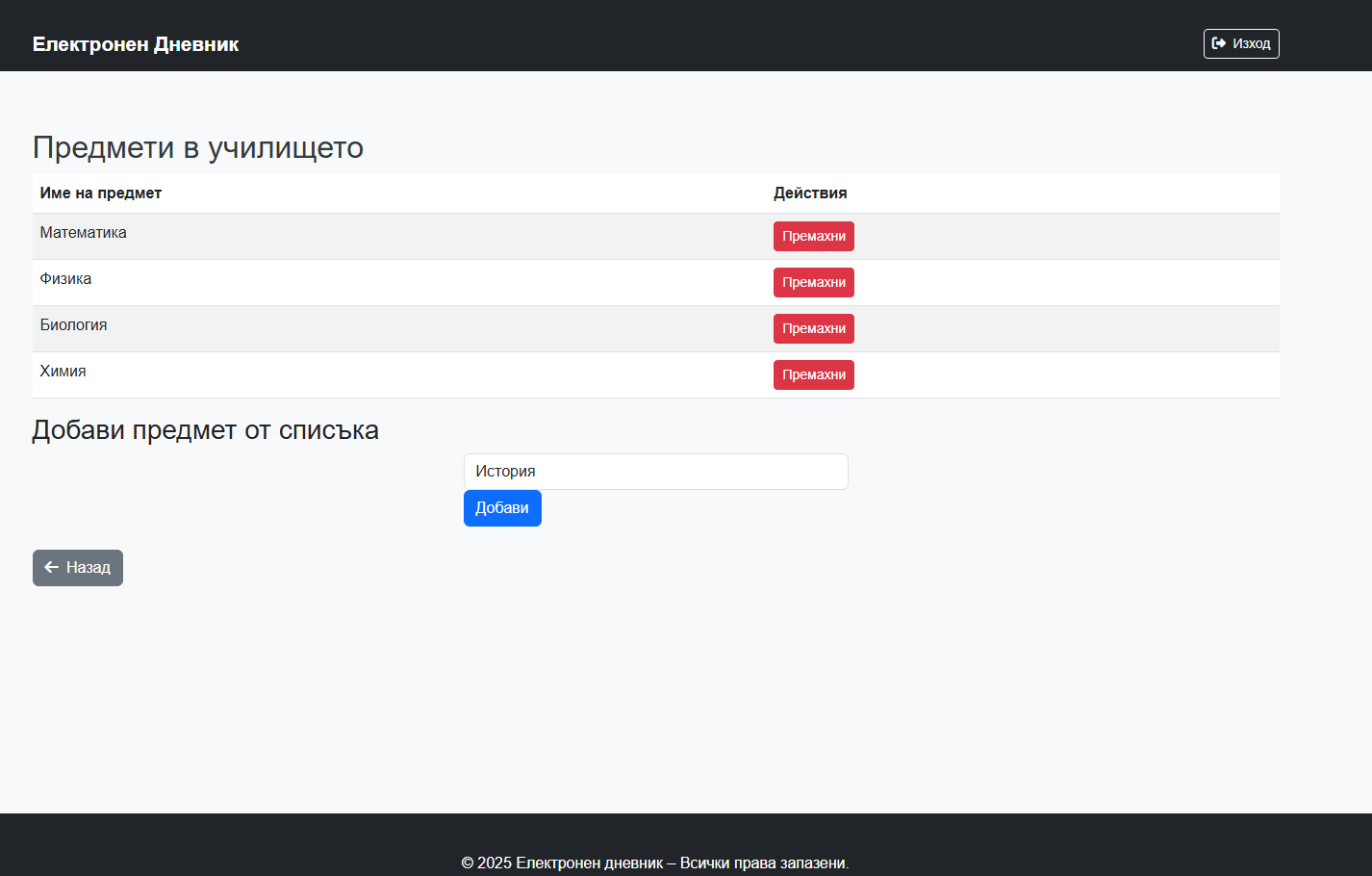
1. **Списък с ученици**

****

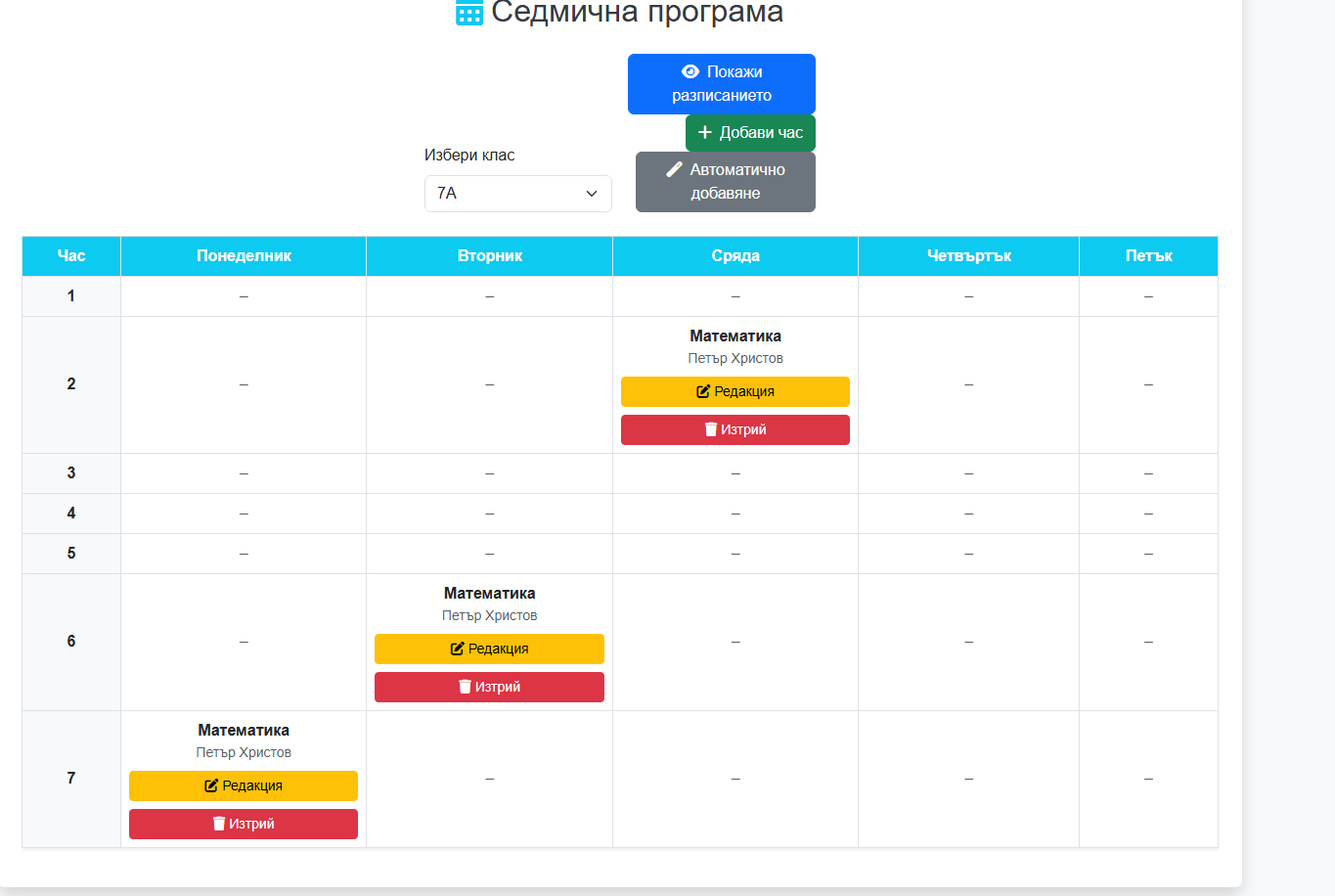
1. **Редактиране на ученик**

****

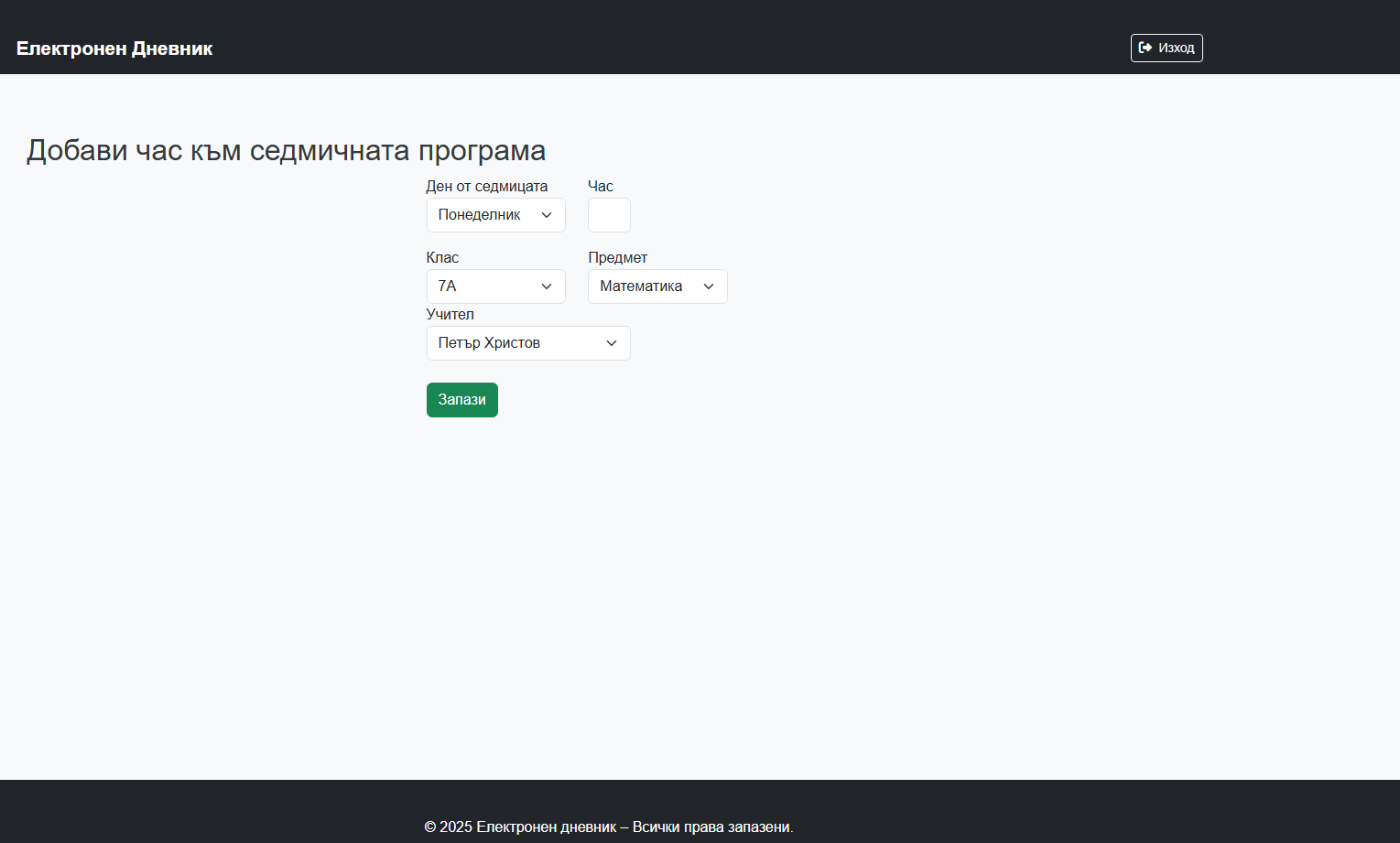
1. **Страница с предмети**

****

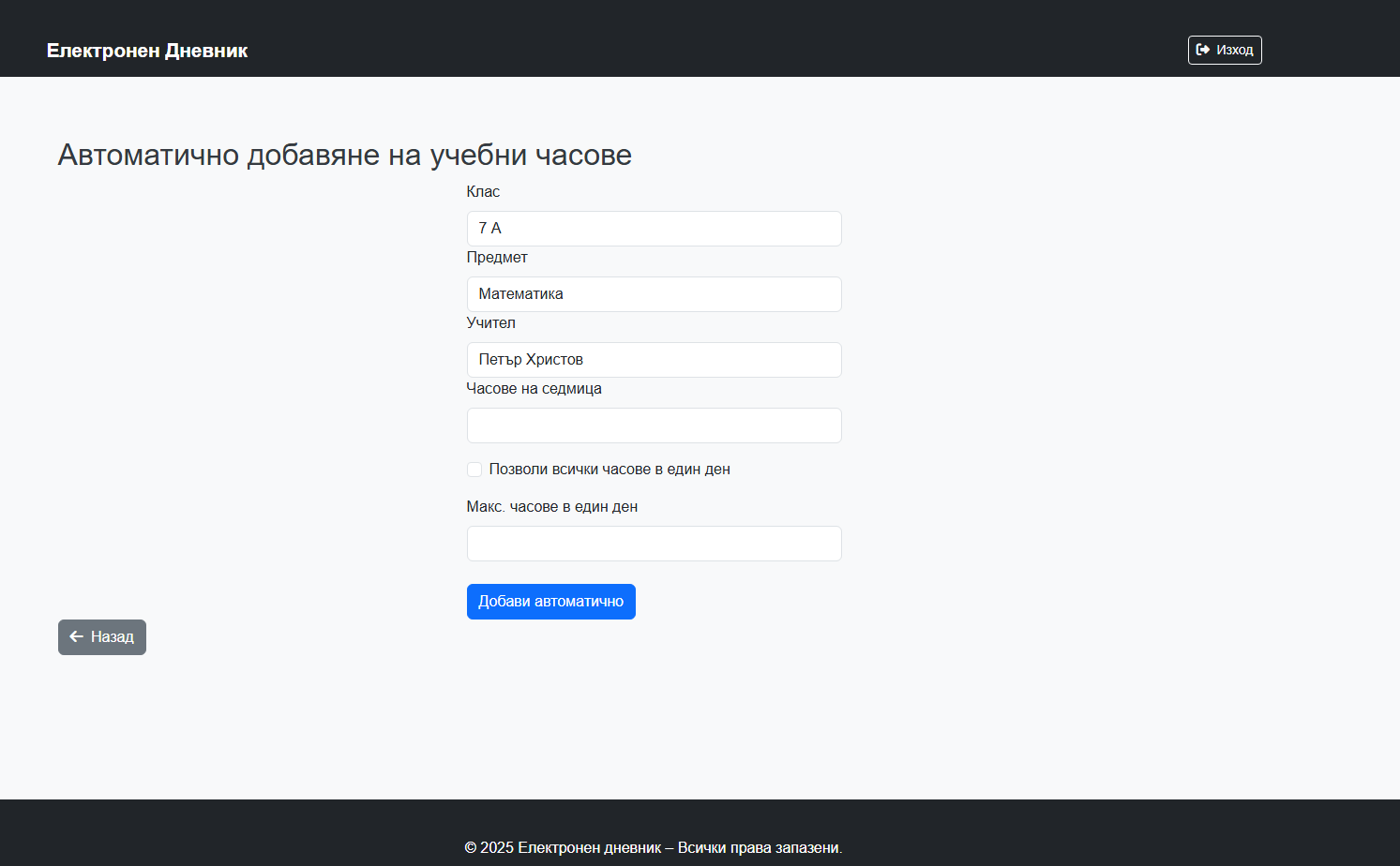
1. **Страница с учебни програми**

****

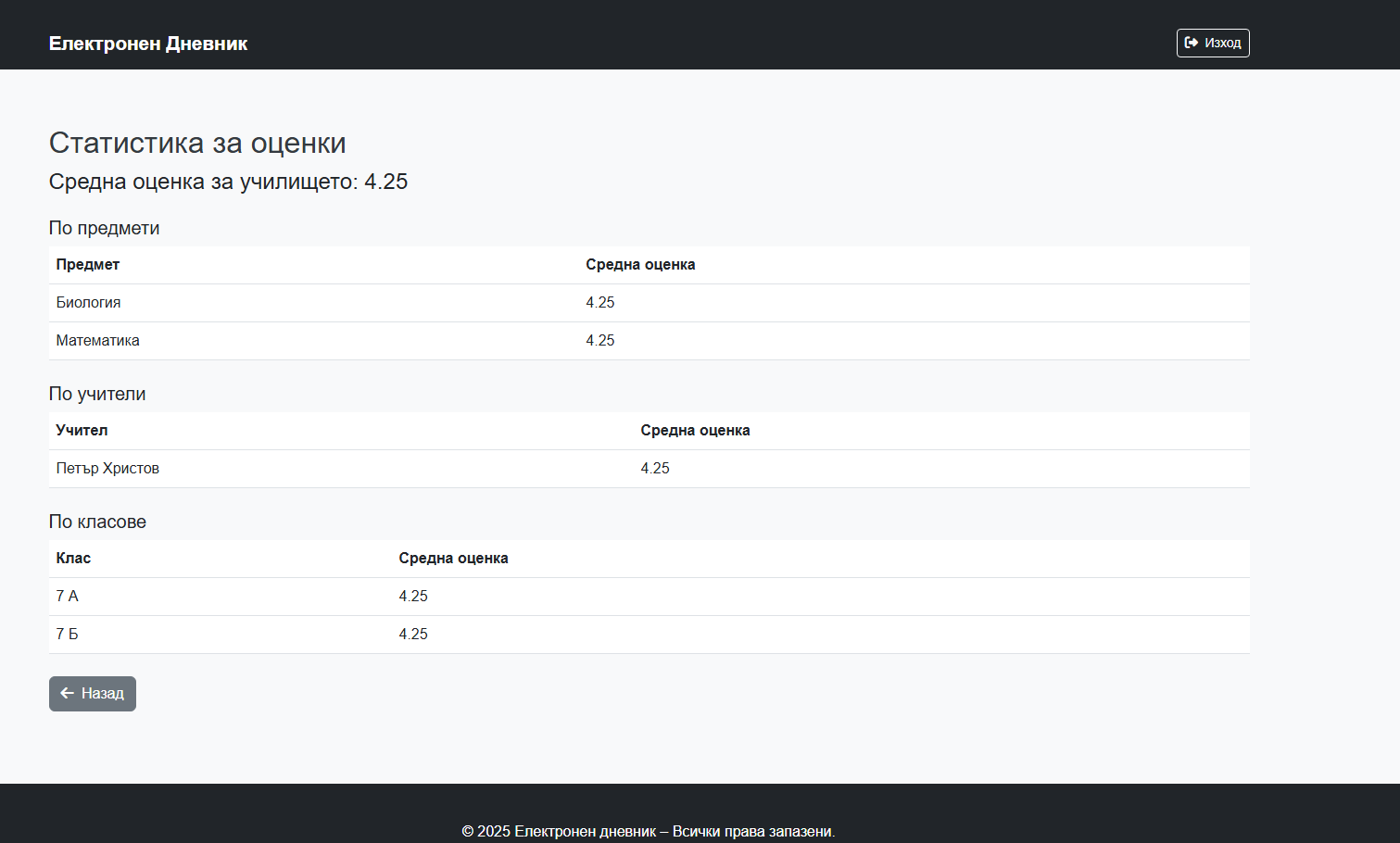
1. **Добавяне на час**

****

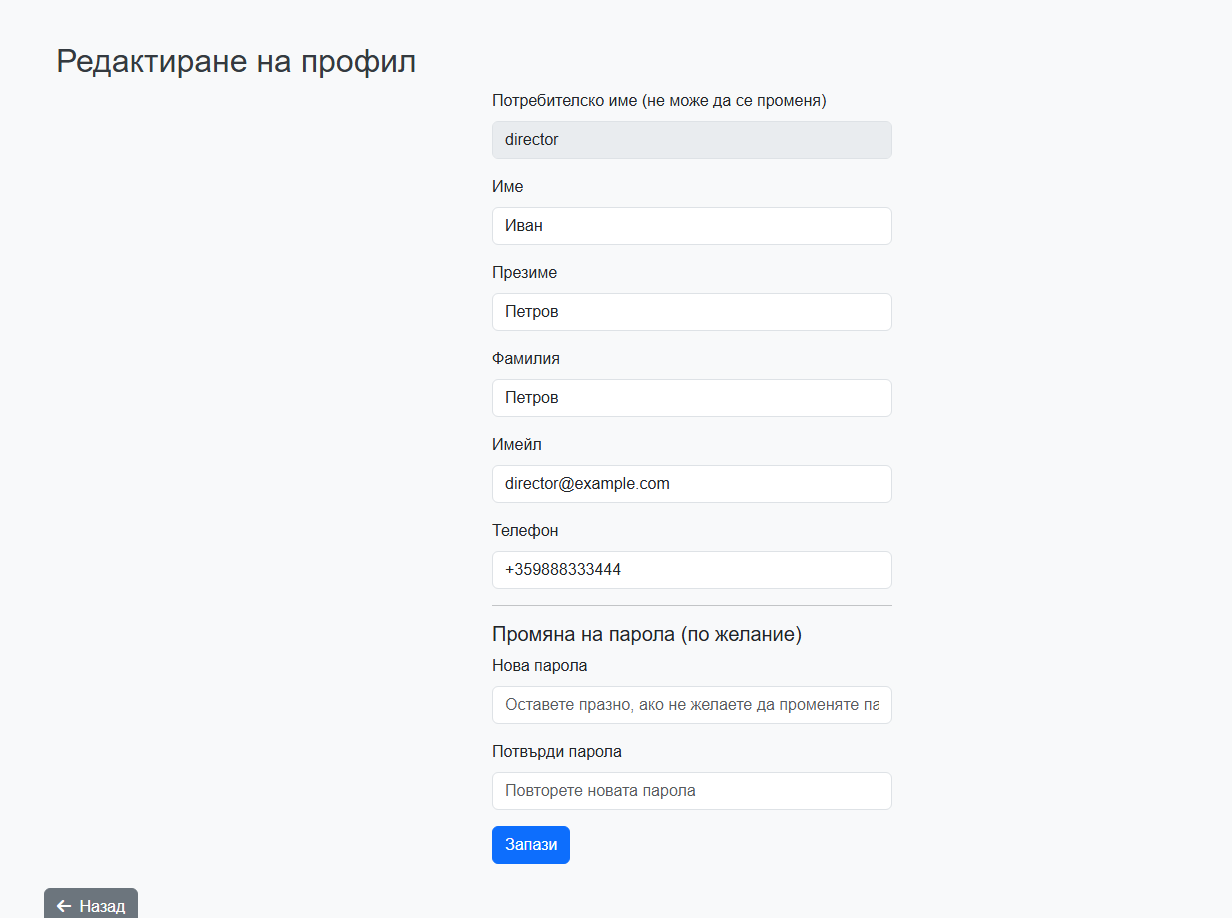
1. **Автоматично добавяне на час**

****

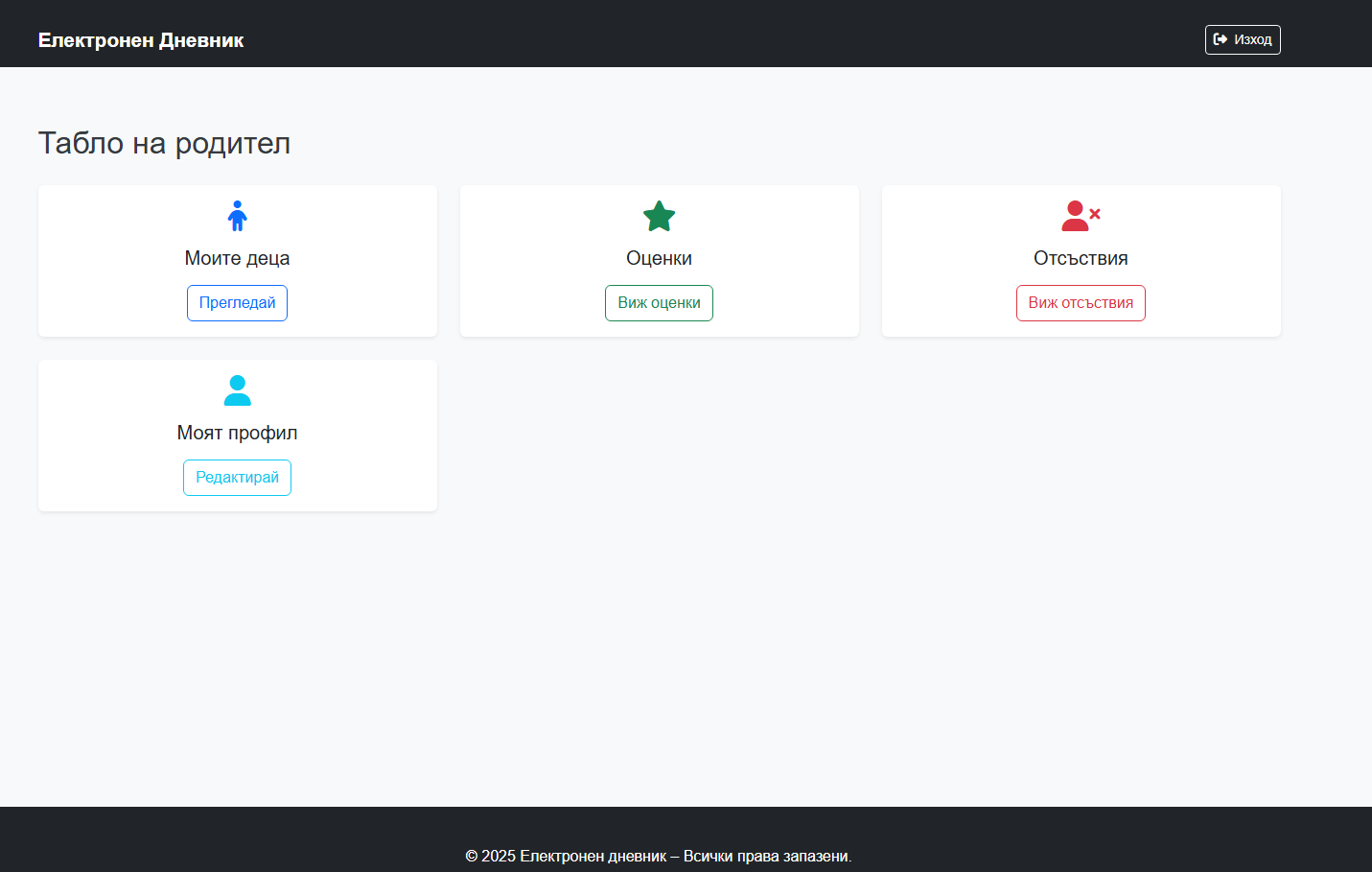
1. **Статистика на училището**

****

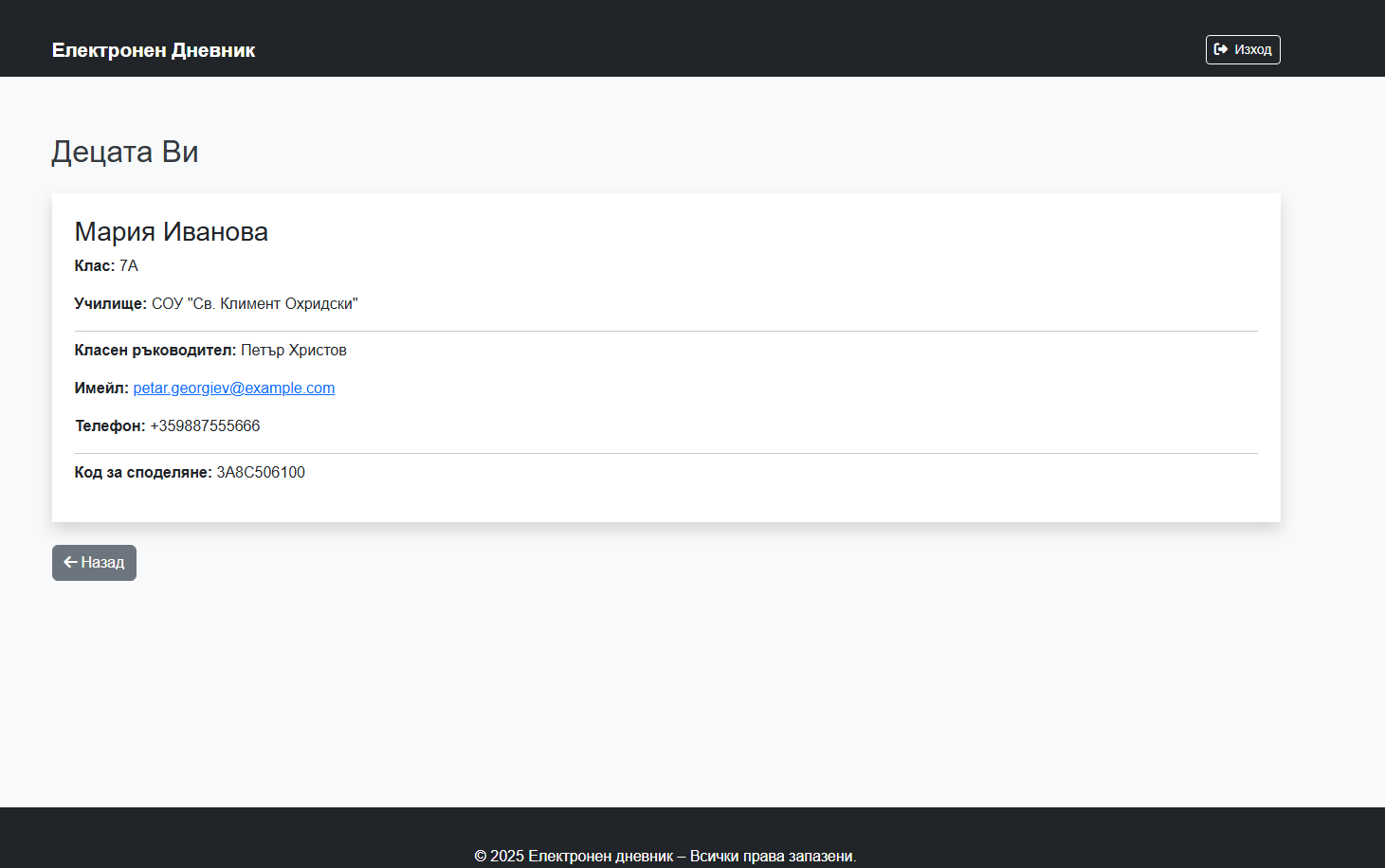
1. **Редактиране на профил на директора**

****

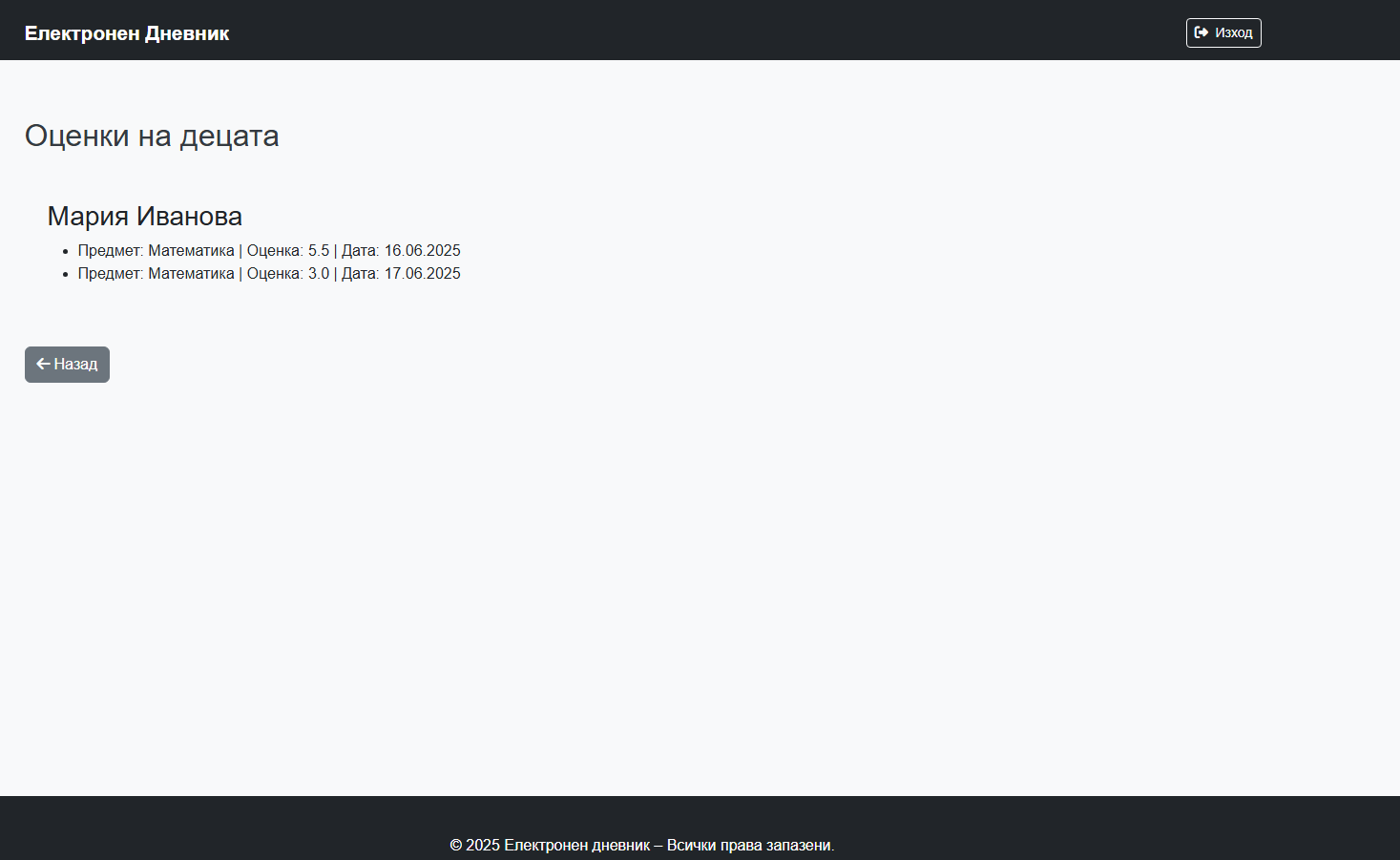
1. **Страници при влизане като родител**
2. **Основна страница**

****

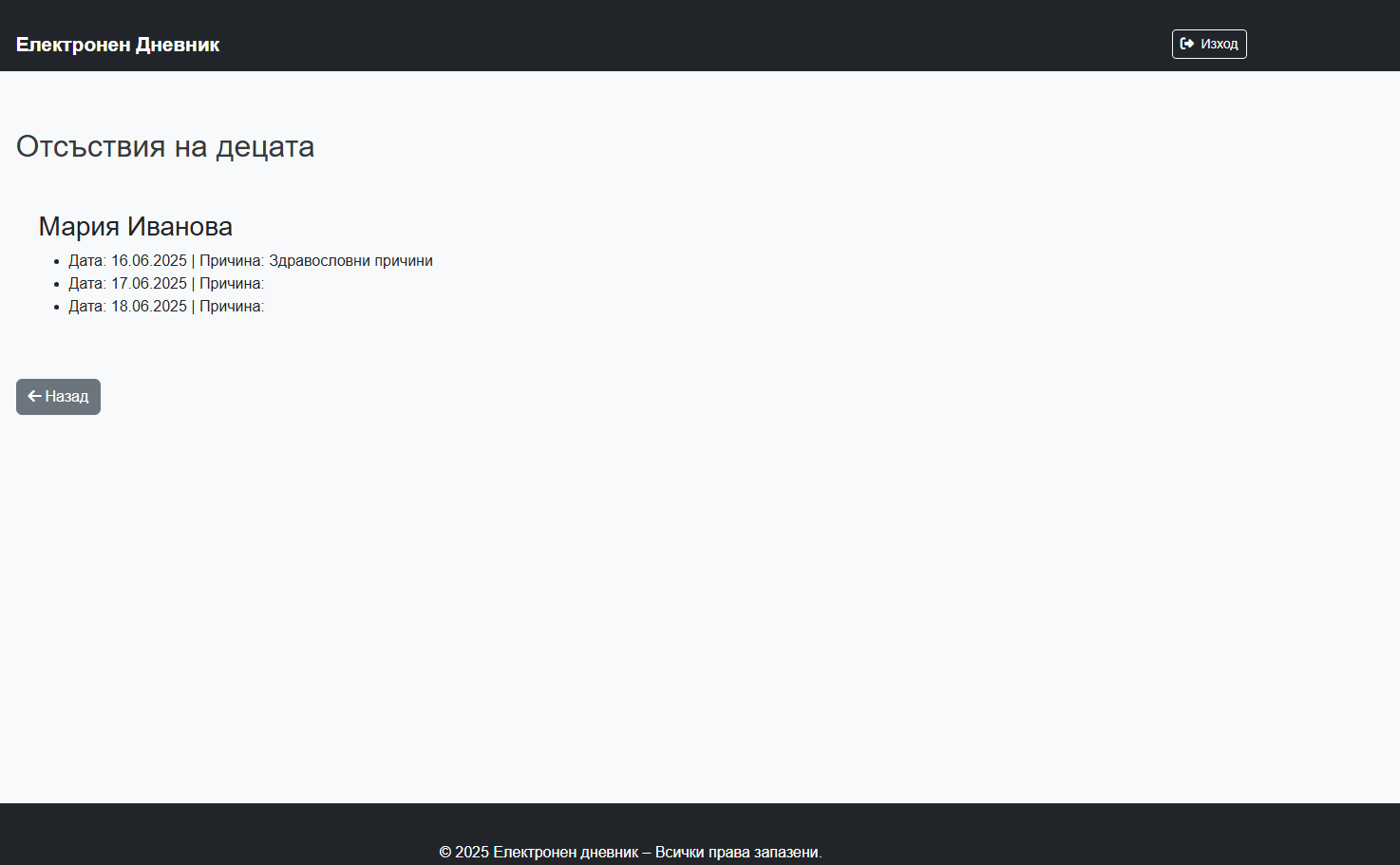
1. **Страница с деца на родителя**

****

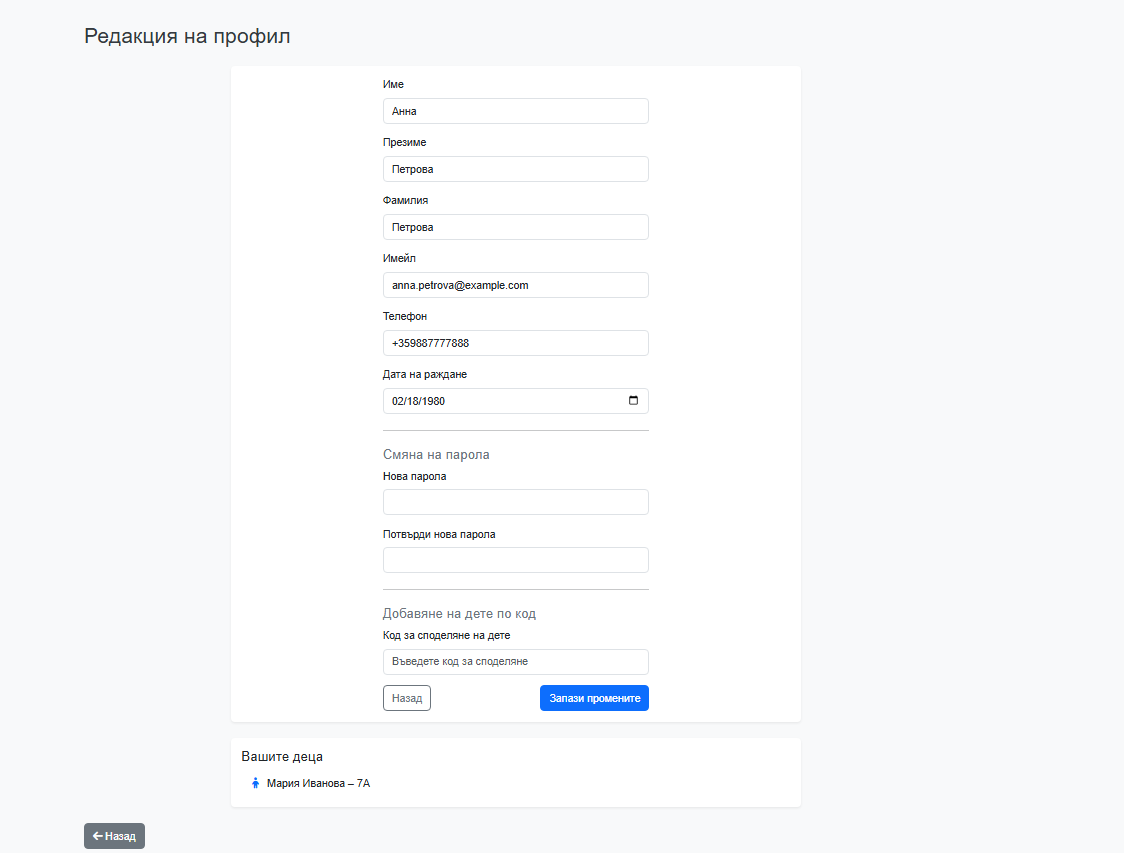
1. **Страница с оценки на децата на родителя**

****

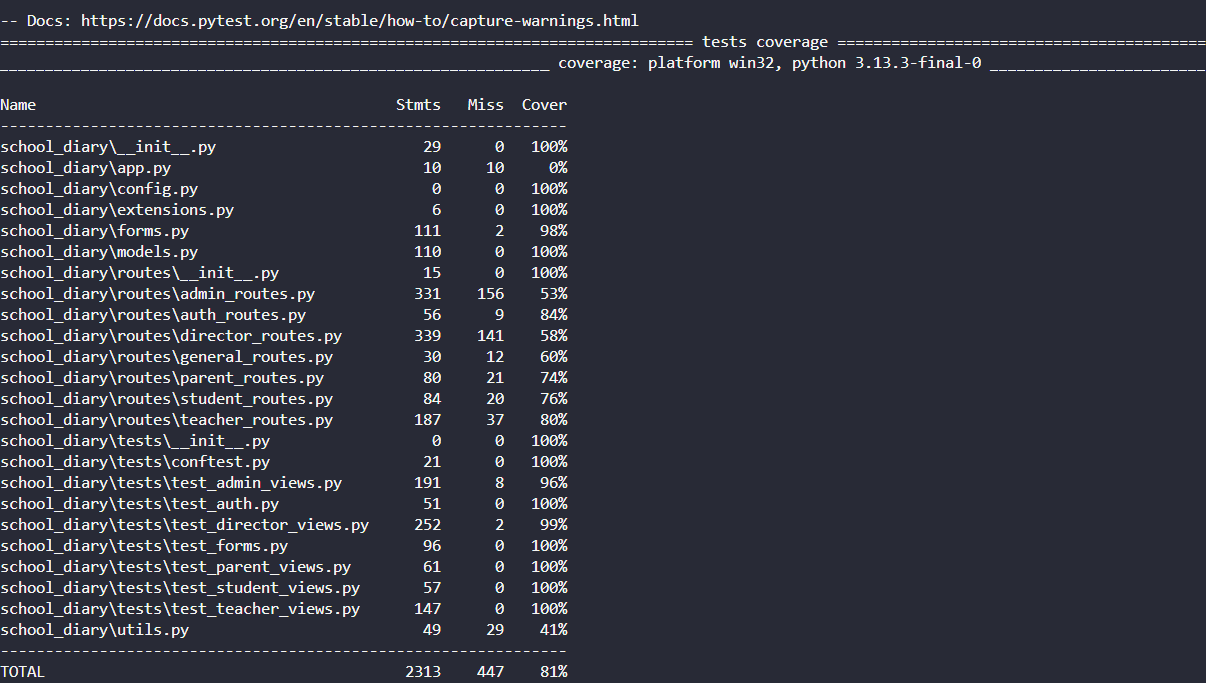
1. **Страница с отсъствия на децата на родителя**

****

1. **Страница за редактиране на профила на родителя**

****

**Резултати от Unit Tests**

****