Schooled For Success

Онлайн платформа за училища

Version 1.0

Strypes Lab септември- ноември 2022

Ерика Карамучева

**Съдържание:**

*1. Изисквания*

*2. Цели*

*3. Използвани библиотеки и модули*

*4. Архитектура*

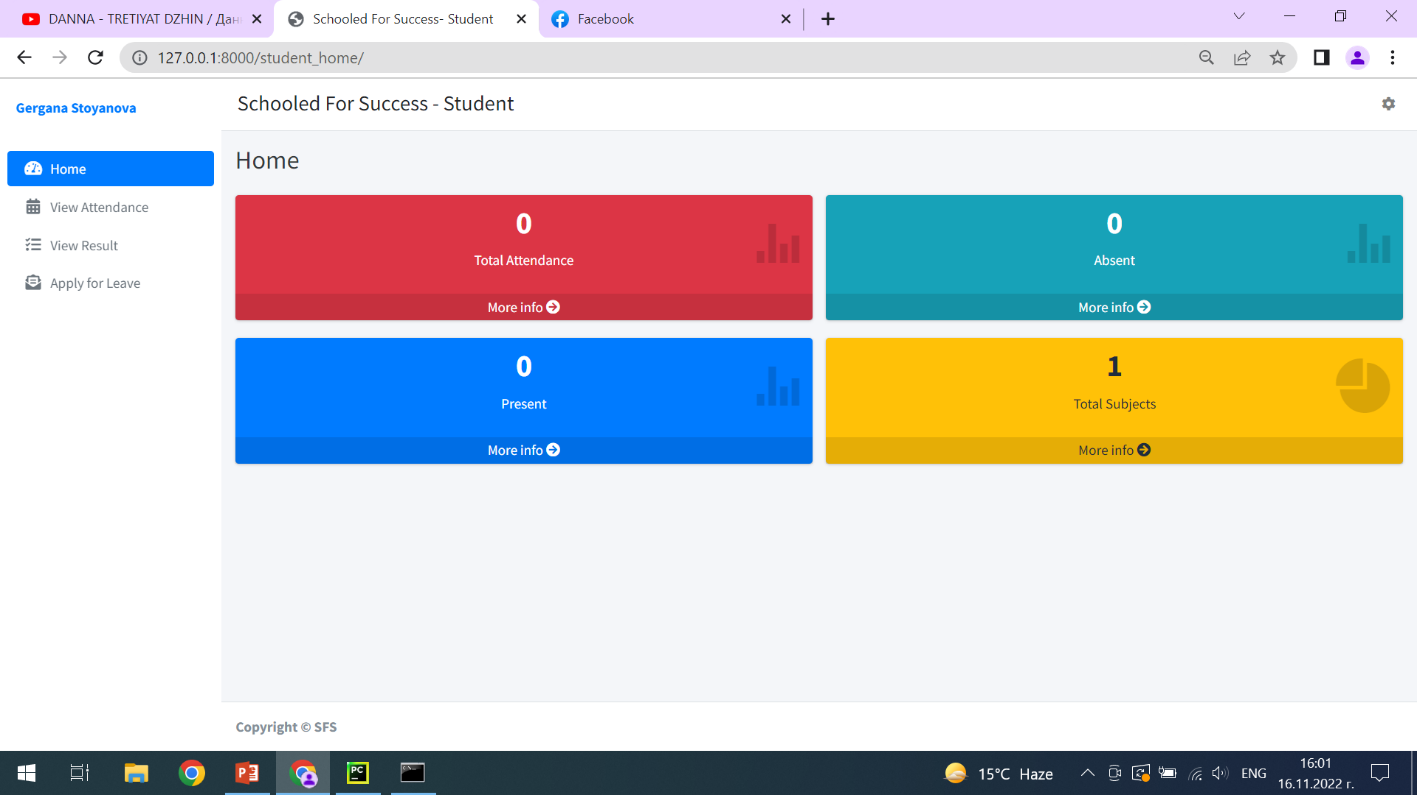
*5. Version control*

* *Изисквания:*
* Python: 3.9.x
* https://www.python.org/downloads/
* Django: 4.1.x
* *Цели:*
* Разработване на онлайн платформа, предназначена за образователни институции. Изпълнява ролята на онлайн дневник, в който се вписват оценките и се проследяват присъствията на студентите. Създадено е и място, в което могат да се публикуват различни съобщения (може да се развие съобразно изискванията на институцията).
* *Използвани библиотеки и модули*
* Requests- за изпращане на HTTP заявки
* django.dispatch- изпращане на заявки
* django.db- създаване на база данни и модели в тях
* django.views.decorators.csrf- защита срещу фалшифициране на заявки между сайтове .
* и други

1. *Архитектура*

* **models.py**- Файл, съдържащ основните модели за приложението.
  + SessionYearModel- съдържа информация за началото и края на академичната година
  + CustomUser- пренаписан User модел, който може да бъде 3 типа- админ, студент, преподавател
  + Adminn- има поле, рефериращо към Customuser
  + Staff- включва поле, рефериращо към админ и адрес
  + Courses- има име на курса
  + Предмет- име на предмета, към кой курс е включен( връзка едно-към-едно), id на преподавателя, който го провежда ( външен ключ)
  + Student- поле, рефериращо към CustomUser, пол, адрес, към кой курс принадлежи, академична година
  + Message- таблица за съобщения; админ, съобщение
  + Attendance-отбелязва, присъствия по предмети; id на предмет, дата на занятие, академична година
  + AttendancReport-присъствие на студенти; студентско id, id по предмет, статус (посетено или не)
  + LeaveReportStudent-пази заявките за отпуск на студентите; студентско id, желана дата за отпуск, причина, действие(прието, отхвърлено, не е извършено такова)
  + LeaveReportTeacher-пази заявките за отпуск на преподавателите; преподавателско id, желана дата за отпуск, причина, действие(прието, отхвърлено, не е извършено такова)
  + StudentsResult- пази оценките, които са получили студентите; id на студент, id на предмет, оценка
* **Urls.py**- задава пътищата за изгледите; описани са пътищата до всяка, от функциите
* **Views.py**- изгледи, които са общи за цялото приложение:
* doLogin()- проверява в базата данни дали user с такива имейл и парола съществува; ако не- показва, че данните са некоректни, ако да- проверява каква роля играе потребителя и го пренасочва към различна начална страница.
* Logout()- връща ни към най, първоначалната страница; в нашия случай- login формата
* **TeacherViews**-изгледи, които контролират преподавателите:
* Staff\_home()- взема общия брой студенти, на които преподава съответния преподавател, броя на предметите, които води, общ брой на провеждани занятия, брой на молби за отпуск
* Staff\_profile/\_update()- на екрана се показват текущите данни, които са запазени за съответния потребител в базата данни. На основата на неговото id можем да променим част от информацията.
* Add\_result\_save()- вземаме от потребителя студента, на когото искаме да въведем оценка, по кой предмет и каква е самата оценка. Опитваме се да въведем данните в базата данни. Показваме на екрана дали заявката е минала или не.
* Staff\_apply\_leave()- от потребителя се взема датата и причината и я записваме в модела LeaveReportTeacher.
* Create\_message\_save(), staff\_take\_attendance()-по подобие на описаните горе методи.
* **StudentViews**-изгледи, контролиращи студентите:
* Student\_home()- извежда на екрана общия брой занятия, брой присъствия, отсъствия и предмети
* Student\_profile\_update()- подобно на staff\_profile\_update()- вземаме id-то на студента и променяме данните, които искаме
* Student\_view\_result()- взема резултатите от базата данни с оценки и ги показва
* View\_attendance()- задава се предмет, академична година и период, в който искаме да проверим присъствията.
* Student\_apply\_leave()- от потребителя се взема датата и причината и я записваме в модела
* Create\_message()-записваме име на създател и съобщение. Записваме в Message модела.
* **AdminViews**- изгледи, контролирани от админ:
* Admin\_home()-извличаме броя студенти, преподаватели, курсове, предмети
* Profile\_update()- на базата на id правим промени на съществуващ запис в базата
* Manage\_subject()- показваме на екрана информацията за всичките, предмети, които сме създали, с възможност за промяна и изтриване; manage\_course(), manage\_session()- подобни
* Delete\_subject()-изтрива предмета от базата данни; delete\_course(), delete\_session()- подобни
* Edit\_subject()- показва информацията, която е записана за съответния предмет. Може да я променим; edit\_course(), edit\_session ()- подобни
* Add\_subject()- създаваме нов обект от типа Предмет. Записваме неговото име, курса, към когото принадлежи и името на преподавателя, който ще води занятията; add\_course(), add\_session ()- подобни
* Create\_message()-за създаване на съобщения
* A**dminTeacherViews**-изгледи, контролирани от админ по отношение на менажиране на преподаватели
* Add\_staff()- подобно на другите функции за добавяне; взема информация от потребителя и създава нова инстанция от тип преподавател
* Manage\_staff()- показва информацията за преподавателския екип; дава възможност за корекция на данните
* Edit\_staff()- коригира данните за даден прподавател
* Delete\_staff()- изтрива обекта
* Staff\_leave\_view()- преглежда молбите за отпуск
* Staff\_leave\_view\_approve()- задава статус 1 на молбите за отпуск- одобрява ги
* Staff\_leave\_view\_reject()- задава статус 2 на молбите за отпуск- отхвърля ги
* **AdminStudentViews**-изгледи, контролирани от админ по отношение на менажиране на студенти
* Add\_student()- подобно на другите функции за добавяне; взема информация от потребителя и създава нова инстанция от тип студент
* Manage\_student()- показва информацията за студентите; дава възможност за корекция на данните
* Edit\_student()- коригира данните за даден студент
* Delete\_student()- изтрива обекта
* Student\_leave\_view()- преглежда молбите за отпуск
* Student\_leave\_view\_approve()- задава статус 1 на молбите за отпуск- одобрява ги
* Student\_leave\_view\_reject()- задава статус 2 на молбите за отпуск- отхвърля ги

1. *Version control*

* За контрол на версиите се използва GitHub:  
  Линк: <https://github.com/ErikaKaramucheva>
* Идеи за бъдещо развитие на проекта:
* Създаване на класни стаи, в които могат да се качват различни файлове
* Създаване на специална секция за тестове и изпити
* Ранг листи, изградени на база успех
* Място за забележки и похвали

