# Защита

Аз и моята съученичка сме разработили уеб приложение на Django, използвайки Java Script и HTML базирано на сайта на нашето училище ПГЕЕ Банско. Нашият проект е електронна библиотека, която може да се намери информация за предмети по специалността „Приложно програмиране“ от 8-ми до 12-ти клас и да се проверят знанията с помощта на тестовете.

Google и Microsoft Teams например имат такава опция за създаване на тестове, но се съдържат някои опции, от които се нуждаем, а липсват и аз съм се заела с тази задача.

Проекта ни е създаден от 2 части – визия и функционалност. Всеки сайт или уеб приложение винаги се състои от отбор, в който едната половина работи по функционалността, а другата по визията. В моят случай работя по функционалността или по-точно, приложение за създаване на тестове, аз съм дала базата да се направи един тест, който не е попълнен, но е направен така, че учител който би искал да го използва, за да направи тест на учениците си например, може по лесен и ефективен начин да го създаде.

\*ТУК ПОКАЗВАМ КАК ИЗГЛЕЖДА ПРИЛОЖЕНИЕТО ЗА СЪЗДАВАНЕ НА ТЕСТОВЕ\*

Използвала съм степен за оценяване на резултата по Таксономията на Блум, с различни степени на трудност.

Има различни начини на оценяване както различни задания на въпроси и за тази цел ми е нужна базата данни MySQL. Също така проекта е създаден с помощта на DFR, един мощен инструмент, от който всъщност може да направи пълноценно приложение с минимални усилия, това е нещо много функционално.

Основната функционалност като почнеме от самото провеждане на тест и стигнеме до редактирането и задаването на нови въпроси и така нататъка, е изведена като софтуер към клиента. Това всъщност е интегрирането на Vue, DRF и Django. Django предоставя шаблоните и сервира на клиента Java Script файлове, но истинската част се върши при клиента. Той изпраща запитване чрез Java Script до сървъра и сървъра с помощта на библиотеката DRF връща данните, сериализатора и случайния избор на въпроси и редактирането става при клиента, защото ако всяка отделна променя я пращам на сърваръа и той през това време задържа ресурси, поддържа сесии, за да помним на кой въпрос ще му пратим редакция, това излишно натоварва както комуникацията между клинет и сървър, така и самия сървър.

По-елегантното и по-удобното решение е моето решение. Всичко да е изнесено при клиента, в началото сървъра се изисква всички въпроси по дадена тема и на место при клиента се генерира по случаен начин тест или се редактира в зависимост от това от кой от режимите сме и всичко тва е концертирано в два Java Script файла, което позволява частта от шабалона и частта, която касае сериализаторите и въпросния Java Script да могат да се прикачат почти без проблем, към което и да е друго приложение и цялата тестова система ще отиде там.

Настройване на JavaScript към проекта:

- JS файла ни, се състои от 3 елемента: данни(data), методи(methods) и направени функции(created: functions)

1. Данните включват дефиниране на оценка, тема, въпроси и предмети

2. Методите включват стартиране на теста, избиране на случаен начин тема, отброяване на времето за решаване под формата на таймер, оценяване на отговорите и приключване на теста

3. Функциите представляват команди на подадените заявки от methods

Накратко моята част от проекта ни електронна библиотека, е да демонстрирам как се вгражда модул в приложение или в моят случай тестова система. От една страна е функционално интегрирана и ние сме работили в екип, за да се напаснат едно с друго, а от друга страна е алгоритмична и функционална. Това е независима система тестове.

Всеки, който има достъп може да променя нивата по Блум, типа на въпросите или формулировката им. Използвали сме Vue и базата данни MySQL, която е обща за цялото приложение. Също така е използван и DRF, който е задължителна част като концепция, защото алтернативите са сложни за реализация.