# Резюме на дипломен проект: Разработка на REST API с Django REST Framework

# 1.Представяне

Уважаема г-жо Директор, Уважаема комисия,уважавани гости.

Аз съм Димитър Дурчов

Темата на моя дипломен проект е „Разработка на REST API с Django REST Framework”

# 1.Увод

В съвременния свят на уеб технологиите, REST API е основен стандарт за обмен на данни между различни системи и платформи. Този проект цели създаването на такъв API с помощта на Django REST Framework, който да управлява курсове за деца, демонстрирайки основните принципи и практически приложения.

# 2. Цел и значение

За проекта си реших да създам надежден и гъвкав REST API за управление на учебни курсове. Този API ще позволи на учители и ученици лесно да добавят, редактират, преглеждат и изтриват курсове онлайн. Като начало, разделих цялото нещо на няколко основни части: ресурси, методи и правила за сигурност. Например, използвах HTTP методите – GET за преглеждане, POST за създаване, PUT за обновяване и DELETE за изтриване. Така се гарантира, че всяка команда си има ясно предназначение. За да направя API-то по-надежден, въведох проверки за грешки. Ако някой изпрати невалидна заявка, получавах ясно съобщение за грешка, вместо да се срине системата. Освен това, добавих и защита чрез API ключове, така че само одобрени потребители да могат да използват услугата

# 3. Основи на REST архитектурата

. REST (Representational State Transfer) е един вид начин за създаване на уеб услуги или API-та. Това означава, че с REST можем да правим приложения, които си комуникират през интернет по определени правила. REST е гъвкав, тъй като не изисква сложни инструкции, а просто следва стандартните правила на уеба. Той работи с ресурси, които може да са неща като курсове, ученици или преподаватели, представени в JSON формат, който е лесен за четене и обработка от компютри и хора.

# 4. Технологичен стек

За създаването на моя REST API използвах различни технологии и инструменти, които ми помогнаха да го направя по-лесно и ефективно.

Първо, за програмиране използвах Python

За сървърната част, на която работи API-то, използвах Flask

За базата данни, където съхранявам информацията за курсовете, учителите и учениците, използвах SQLite

За да проверявам дали апликацията работи правилно, използвах Postman

И накрая, за хостване и тестване в облака, понякога използвах Heroku

# 5. Основни компоненти на API-то

Когато говорим за едно API (уеб услуга), има няколко важни компоненти, без които то не може да работи добре.

. Първо е Endpoints или крайни точки. Това са адресите, които използваме, за да комуникираме с API-то. Например, може да има адрес като api/courses, който връща списък с всички курсове. Всеки ендпойнт е като конкретна врата към определена част от данните. След това идва Форматиране на данни. Друг компонент е Аутентикация и сигурност. Това значи, че трябва да има начин да се провери дали потребителят, който иска да използва API-то И накрая е Обработката на грешки. Трябва да има лесно разбирателни съобщения, ако нещо не е наред — например, ако изпратим грешен формат на заявката или се опитваме да изтрием ресурс, който не съществува.

# 6. Безопасност и тестове

Когато създаваме API, е много важно то да е безопасно и надеждно. Това означава, че трябва да предпазим данните и системата да не може да бъде лесно хакната или използвана по лош начин

Първо, за безопасност използваме аутентикация

Освен това, добавяме и авторизация

Друг важен аспект е защита от атаки — това включва проверка за опасни заявки или код, който може да За да сме сигурни, че всичко работи както трябва, използваме тестове. Това са програми или сценарии, които проверяват дали API-то връща правилните резултати и се държи както трябва при различни ситуации навреди

# 7. Приложение и интерфейс

Когато създавам API за управление на учебни курсове, искам то да е удобно за използване от хората. Затова трябва да направя приложение с добър интерфейс. Интерфейсът може да бъде уеб страница или мобилно приложение, в което учители и ученици да имат ясни и лесни за разбиране бутони, менюта и форми. Например, когато искаш да добавиш нов курс, да имаш бутон “Добави курс”, след като натиснеш него, се появява форма, където въвеждаш име, описание и други данни. После натискаш “Запази” и данните се изпращат към API-то.

Така, API-то работи зад сцената, а хората гледат красив и удобен дизайн, който им помага бързо да си вършат задачите. Това прави приложението по-приятно и по-лесно за използване.

# 8. Резултати и перспективи

Когато разкажа за проекта си за REST API-то за управление на учебни курсове, мога да кажа, че съм много доволен от това, което постигнах. Научих много за това как работят уеб услугите и как да ги направя надеждни, сигурни и лесни за използване. В перспектива, такива умения ме карат да се чувствам по-уверен и подготвен за бъдеща кариера в областта на уеб разработката или информационните технологии. Надявам се тези знания да ми помогнат и в училище, и в бъдещи проекти или дори в реална работа.

Общо взето, това беше много полезно и интересно преживяване, и имам големи надежди, че ще продължа да създавам още по-големи и по-добри неща!

# 9.Заключение

В заключение мога да кажа, че създаването на този REST API беше много полезен и ценен опит за мен. Научих как работи една уеб услуга, как да я направя сигурна и надеждна, и как да създам приложение с удобен интерфейс. Това ми помогна да разбера по-добре технологиите, които се използват при реалните уеб проекти.

Вярвам, че така придобитите умения ще ми бъдат полезни в бъдеще, ако реша да се занимавам с програмиране или разработка на уеб приложения. Радвам се, че успях да реализирам тази идея и да науча всичко това, защото съм убеден, че подобни проекти ще ми помогнат да се развия като бъдещ професионалист в технологичната сфера.

.