Резюме на дипломния проект

**1.Представяне**

Уважаема г-жо Директор, Уважаема комисия, уважаеми гости (ако има такива / или поименно –ако е само един конкретен гост).

Аз съм Сейхан Молла

Темата на моя дипломен проект е : „Използване на REST API от страна на клиента “.

2.Увод

Динамичните уеб приложения са изключително важни в днешния свят, тъй като позволяват бързо и в реално време обмен на информация с потребителите. Въвеждането на нови технологии, които дават възможност на клиентите да взаимодействат с данни по лесен и гъвкав начин, е основен етап в развитието на софтуерната индустрия. Този дипломен проект се съсредоточава върху използването на REST API от страна на клиента, което е ключов аспект от тези иновации.

3.Описание

Още в началото на работата си по проекта анализирах възможните подходи за реализиране на връзката между клиента и външния REST API. След като се запознах с изискванията на заданието и спецификата на предоставения API, взех решение да разделя приложението на две основни части – backend и frontend, като за backend частта се спрях на Django, а за frontend – на Vue.js и Axios.

REST API (Representational State Transfer Application Programming Interface) е интерфейс за програмиране на приложения, който следва принципите на архитектурния стил REST. REST определя как софтуерните системи трябва да комуникират помежду си, използвайки HTTP протокола. Той предоставя един прост, последователен начин за достъп до данни и функционалност.

4. backend частта

След като анализирах структурата и възможностите на външното REST API на ПГЕЕ – гр. Банско, взех решението да реализирам отделни изгледи във Django backend-а за трите основни типа данни, които са достъпни чрез този API. Всяка от тези функции обработва заявката, изпраща необходимата HTTP заявка към сървъра на ПГЕЕ и връща получения отговор във вид, подходящ за фронтенда.

5. frontend частта

След като backend-ът беше готов и можеше успешно да препраща заявки към външния REST API, се заех с изграждането на потребителския интерфейс. За целта създадох HTML шаблон (info\_main.html), в който дефинирах основната структура на приложението. В този шаблон включих всички необходими елементи за визуализация на разписанието, както и специални контейнери за анимирания часовник, който беше част от изискванията за динамичен интерфейс. във файла info\_main.js реализирах основната логика на Vue компонента. Там дефинирах всички необходими данни и методи, които управляват състоянието на приложението и обработват получените от backend-а данни. Използвах Axios за изпращане на заявки към локалния сървър, като по този начин осигурих асинхронно извличане на информацията за разписанието.

6. ИНТЕГРАЦИЯ С ВЪНШЕН REST API

След проучване на възможните подходи, реших да реализирам така наречения "прокси" слой в backend-а. Това означава, че вместо frontend-ът да се опитва директно да комуникира с външния API, той изпраща заявките си към локалния сървър (Django), който от своя страна препраща тези заявки към външния REST API, получава отговора и го връща обратно към клиента. По този начин всички заявки към външния сървър се изпълняват от backend-а, където CORS ограниченията не важат, а frontend-ът комуникира само с локалния сървър, което е напълно разрешено от браузъра.

7.ТЕХНОЛОГИИ ЗА КЛИЕНТСКАТА ЧАСТ

Клиентската част на приложението е изградена с помощта на модерна JavaScript библиотека Vue.js.

Избрах Vue.js. защото:

Има добра документация и е лесна за интеграция в съществуващи проекти.

Предлага двупосочна свързаност на данните, което улеснява управлението на взаимодействията с потребителския интерфейс.

8.Преди заключението  
Разработването на фронтенд приложение с JavaScript(език за програмиране), заедно с технологии като JSON и AJAX, показа как можем да извлечем данни по асинхронен начин и да актуализираме потребителския интерфейс без необходимост от презареждане на страницата. Този подход увеличава динамиката и интерактивността на приложенията.

Използването на JSO(Java Script Object Notation)N и AJAX(Asynchronous JavaScript and XML) е от съществени значения за разработката на RESTful приложения, предлагайки гъвкаво и интерактивно взаимодействие между клиент и сървър. JSON служи като идеален формат за предаване на данни, а AJAX позволява асинхронно извличане и обработка на информация, придавайки динамичност на уеб приложенията.

9.Заключение

В резултат на всички тези стъпки успях да реализирам уеб приложение, което динамично извлича и визуализира данни от външния REST API на ПГЕЕ – гр. Банско. Приложението работи бързо и надеждно, като предоставя на потребителите актуална информация за дневното и седмичното разписание, представена в модерен и анимиран уеб интерфейс. Чрез използването на прокси слой в backend-а успешно преодолях ограниченията, наложени от CORS политиката на браузъра, а интеграцията на Vue.js и Axios направи възможно създаването на динамичен и лесен за поддръжка frontend.

10.Финал

Благодаря за вниманието!

Ако имате въпроси съм готов да отговоря.