**Здравей! Аз съм ученик в 12 клас специалност: "Приложно програмиране" пиша дипломен проект на тема: Използване на REST API от страна на клиента. Моля направи ми съдържание за теоретичната част**

**Съдържание на теоретичната част**

1. **Въведение**
   * Обяснение на важността на използването на REST API.
   * Кратко описание на основните цели на дипломния проект.
2. **Основи на REST API**
   * **Какво е API (Application Programming Interface)?**
     + Дефиниция и обяснение на концепцията за API.
     + Разлика между различните видове API (например SOAP, REST, GraphQL и т.н.).
   * **Какво е REST?**
     + Основни принципи на REST архитектурата.
     + Принципи като stateless, client-server, cacheability и т.н.
   * **REST методи**:
     + Обяснение на основните HTTP методи: GET, POST, PUT, DELETE, PATCH.
3. **Структура на REST API**
   * Обяснение на URL маршрутизация и използване на ресурси.
   * Стандартни HTTP статус кодове (200, 400, 404, 500 и т.н.).
   * Обработка на заявки и отговори.
4. **Използване на REST API от клиента**
   * **Как работи клиент-сървър комуникацията?**
     + Как клиентът изпраща заявки към сървъра и получава отговори.
   * **Как се осъществява комуникацията чрез HTTP?**
     + Основни принципи на HTTP заявките.
   * **Заявки и отговори в JSON формат**:
     + Обяснение на формата JSON и защо се използва.
5. **Използване на REST API в клиентски приложения**
   * Примери за използване на REST API в различни типове клиентски приложения:
     + Веб приложения
     + Мобилни приложения
   * Технологии за работа с REST API:
     + JavaScript (Fetch API, Axios)
     + Python (requests библиотека)
     + Java (HttpURLConnection и библиотеки като Retrofit)
   * Примерни случаи на приложение (например използване на REST API за достъп до външни данни като прогноза за времето, новини и т.н.).
6. **Обработка на отговорите от API**
   * Как да се валидират и обработват данните, получени от API (например проверка за грешки, валидност на отговорите).
   * Примери за работа с грешки (например как да се обработват статус кодове като 404, 500).
7. **Сигурност при използване на REST API**
   * Какви са рисковете и как да се защитят заявките към API:
     + Използване на токени за автентикация (например Bearer токен).
     + Защита на данни чрез HTTPS.
     + Rate limiting и защита от атаки.
8. **Теоретични примери**
   * Примерен код за заявка и обработка на отговор в JavaScript (Fetch API).
   * Примерен код за заявка и обработка на отговор в Python (requests библиотека).
9. **Предизвикателства при работа с REST API**
   * Какви могат да бъдат предизвикателствата при работа с API:
     + Лимитации на бързодействието.
     + Ограничения в броя на заявките.
     + Проблеми с интеграцията и различия между различни API.
10. **Заключение**
    * Резюме на основните аспекти на използването на REST API от страна на клиента.
    * Изводи относно приложимостта и ефективността на REST API в съвременните уеб и мобилни приложения.