**Теоретична част**

**I. Въведение в REST API**

1. **Определение на REST API**
   * Какво е REST (Representational State Transfer)?
   * Какво представлява API (Application Programming Interface)?
   * Основни характеристики на REST API.
2. **История и развитие**
   * Произход на REST: Разработено от Рой Фийлдегиб (Roy Fielding) през 2000 г.
   * Развитие на REST архитектурата и нейното внедряване в съвременната разработка на приложения.

**II. Основни принципи на REST**

1. **Архитектурните ограничения на REST**
   * **Client-Server**: Разделение на клиент и сървър.
   * **Stateless**: Безсъстояние на заявките – няма съхранение на информация за предишните взаимодействия.
   * **Cacheable**: Подобряване на производителността чрез кеширане на отговори.
   * **Layered System**: Архитектура, която позволява управление в различни слоеве.
   * **Uniform Interface**: Стандартен интерфейс за комуникация между клиент и сървър.
2. **HTTP методи**
   * **GET**: Извличане на данни.
   * **POST**: Изпращане на данни за създаване на ресурс.
   * **PUT**: Подновяване на съществуващ ресурс.
   * **DELETE**: Изтриване на ресурс.
   * **PATCH**: Частично обновление на ресурс.

**III. Структура на URL и ресурси**

1. **URL структура на RESTful API**
   * Как се формират ендпойнтите.
   * Примери за URL структури и какво представляват те, например:
     + https://api.example.com/users (свързано с ресурс "потребители").
2. **Идентификация на ресурси**
   * Определяне и разлика между ресурси и представяния (representations).
   * Примери за представяния на ресурси (JSON, XML).

**IV. Формати на данни**

1. **JSON (JavaScript Object Notation)**
   * Какво е JSON и защо е предпочитан за REST API?
   * Структура на JSON (обекти и масиви).
   * Примери за JSON данни.
2. **XML (eXtensible Markup Language)**
   * Какво е XML, как се структурира?
   * Сравнение между JSON и XML – предимства и недостатъци на всеки формат.

**V. Сигурност на REST API**

1. **Методи за удостоверяване и оторизация**
   * Основни методи за удостоверяване (Basic Authentication, API Tokens, OAuth 2.0).
   * Разлика между удостоверяване и оторизация.
2. **HTTPS**
   * Какво е HTTPS и защо е важно за сигурност?
   * Как HTTPS защитава данните при комуникация.
3. **CORS (Cross-Origin Resource Sharing)**
   * Обяснение на CORS и как то решава проблема с междудомейнните заявки.
   * Конфигуриране на CORS на сървър.

**VI. Документация на API**

1. **Значение на документацията**
   * Как да се чете документация на REST API (например, Swagger, OpenAPI).
   * Как правилната документация улеснява разработчиците.
2. **Примери и ресурси за изучаване**
   * Линкове към популярни API с докладна документация.

**VII. Приложение на REST API**

1. **Примери за приложения, които използват REST API**
   * Социални медии (Facebook API, Twitter API), метеорологични приложения (OpenWeatherMap), търсачки (Google Maps API).
   * Каквато функционалност предлагат, и примери за заявките, които могат да се правят.
2. **Клиентско приложение**
   * Как клиентска част има нужда и използва REST API за взаимодействие с бекенда.

**Заключение на теоретичната част**

* Обобщение на ключовите точки от теоретичната част.
* Важността на REST API за съвременната разработка на приложения.