FSD Laboratory 06

Aim: Develop a set of REST API using Express and Node.
Objectives:

1. To define HTTP GET and POST operations.
2. To understand and make use of 'REST', 'a REST endpoint', 'API Integration', and 'API Invocation'
3. To understand the use of a REST Client to make POST and GET requests to an API.

Theory:

1. What is REST API?
   REST (Representational State Transfer) API is a set of rules or standards that allow different software systems to communicate over the web, using HTTP methods. It defines how resources (data or functionality) are represented and transferred between a client and a server in a stateless manner. REST APIs often use URLs (Uniform Resource Locators) to identify resources and send/receive data in formats like JSON or XML.

2. Main purpose of REST API.
   The main purpose of a REST API is to enable seamless communication between clients (such as web browsers or mobile apps) and servers over the internet. It allows clients to perform operations like retrieving, creating, updating, or deleting data on the server by interacting with resources through standardized HTTP methods (GET, POST, PUT, DELETE). REST APIs are highly scalable, flexible, and easy to implement, making them ideal for distributed systems and web services.

FAQ:

1. What are HTTP Request types?
   The main HTTP request types (also known as methods) used in REST APIs are:
   GET: Retrieves data from the server (e.g., fetch a list of users).
   POST: Submits new data to the server (e.g., create a new user).
   PUT: Updates existing data on the server (e.g., update user information).
   DELETE: Deletes data from the server (e.g., remove a user).
   PATCH: Partially updates existing data on the server (e.g., modify a specific field of user data).
   HEAD: Similar to GET, but it only retrieves the headers, not the body of the response.
   OPTIONS: Returns the allowed methods for a resource, helping clients understand what actions can be performed.

Output: Screenshots of the output to be attached.