**Backend Course Outlay**

API

Introduction to APIs

Uses of APIs

How does API Work?

**Types of APIs**

Web APIs

SOAP APIs

Remote APIs

Database APIs

Library or Framework APIs

GraphQL APIs

Operating System APIs

**API Methods**

GET

POST

PUT

DELETE

PATCH

**User Flow in API Interaction**

1.Authentication

Request

Processing

Response

Handling

**Parameters in APIs**

QUERY

PATH

BODY

HEADERS

**Fetching Data (GET Request with Query Parameters)**

**Submitting Data (POST Request with Body Parameters)**

**REST APIs**

**What are the key principles of RESTful APIs?**

Resources

HTTP Methods:

Statelessness

Representation

**RESTful API Design Best Practices**

Use Nouns for Resource URIs:

**Use HTTP Status Codes Appropriately:**

200 OK: Request succeeded.

201 Created: Resource successfully created.

204 No Content: Request succeeded, no content to return.

400 Bad Request: Malformed request syntax.

401 Unauthorized: Authentication required.

404 Not Found: Resource not found.

500 Internal Server Error: Generic server error.

**Support Filtering, Sorting, and Pagination:**

Use JSON for Responses:

Provide API Versioning:

Secure the API:

**Tools for Building RESTful APIs**

Frameworks:

Testing Tools:

**WHAT IS A CRUD OPERATIONS ?**

.Create

.Read

Update:

Delete:

**METHODS OF REST API**

PREREQUISITES

SET UP YOUR DEVELOPMENT ENVIRONMENT

**CREATING A BASIC SERVER**

1.Initialize a Project:

Open your terminal.

Run npm init -y to create a package.json file.

2.Install Express:

Run npm install express to install Express.js.

3.Create Server File:

Create a file named server.js.

4.Write Server Code:

Open server.js

and write Server Code

**STEP 1: TEST THE GET METHOD**

1.Select GET from the drop-down list.

2.Enter the URL: localhost:8080/api/customers.

3.Click Send.

4.If successful, you'll see a list of all customers.

**STEP 2: ADD A NEW CUSTOMER**

1.Select POST from the drop-down list.

2.Enter the URL for the POST method.

3.Click on Body, select raw, then JSON.

4.Type the customer's name in the text area.

5.Click Send.

6.If successful, the response will include the new customer's name and ID.

**STEP 3: UPDATE A CUSTOMER**

1.Select PUT from the drop-down list.

2.Enter the URL along with the customer ID you want to update.

3.In the Body, type the new customer name.

4.Click Send.

5.If successful, the response will show the updated customer name and ID

**STEP 4: DELETE A CUSTOMER**

1.Select DELETE from the drop-down list.

2.Enter the URL along with the customer ID you want to delete.

3.Click Send.

4.If successful, the response will show the details of the deleted customer

**FINAL STEP: VERIFY DELETION WITH GET**

1.Send a GET request to localhost:8080/api/customers.

2.The response will show the updated customer list without the deleted entry

**THANK YOU**