Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

AY: 2025-26

Class:	TE	Semester:	V
Course Code:	CSC502	Course Name:	Web Computing

Name of Student:	Yash Nilesh Kasare
Roll No. :	26
Assignment No.:	01
Title of Assignment:	Fundamental Web Technologies
Date of Submission:	04/08/2025
Date of Correction:	04/08/2025

Evaluation

Performance Indicator	Max. Marks	Marks Obtained
Completeness	5	4
Demonstrated Knowledge	3	2
Legibility	2	2
Total	10	8

Performance Indicator	Exceed Expectations (EE)	Meet Expectations (ME)	Below Expectations (BE)
Completeness	5	3-4	1-2
Demonstrated Knowledge Legibility	3	2	1
Legibility	2	1	0

Checked by

Name of Faculty : Ms. Kshitija Gharat

Signature : Gharat

Date : 4/8/25

, Nilesh Kasare
(AI/DS)

Web Computing.

1. Explain how DNS works and the process it follows to resolve domain names to IP addresses.

The Domain Name System (DNS) translates human readable domain names (like www.google.com) into IP addresses (like 142.250.64.78) that computers use to identify each other on a network.

Step by Step process of DNS Resolution :,

Let's say you type www.google.com into your browser here's what happens :

1) Check Browser Cache.

Your browser first checks if it already has the IP address of the domain saved

If yes → It directly uses it (fastest step)

2) Check OS cache

if not browser, the OS checks its DNS cache.

3) Ask the Recursive DNS Resolver (usually ISP)

your system sends a query to a recursive resolver (usually provided by your ISP or set manually like Google DNS 8.8.8.8)

4) Recursive Resolver checks cache.

If It has the IP cached, it returns it.

If not, it continues the search.

5) Query Root DNS Search.

the resolver asks the root DNS server (eg. " ")

where to find info for .com domains.

6.) Query TLD DNS server.

the resolver then asks the .com TLD server "where is google.com?"

TCD server replies with the authorative name server for google.com.

7.) Query Authorative DNS server.

Resolver can ask the authorative server ' What is the IP of www.google.com ?'

This server gives the actual IP address like 142.250.64.78

8.) Return to Client.

The Recursive resolver sends the IP back to your system.

Your system gives it to your browser., browser makes the connection to that IP and loads the website.

Compare the two protocols used for formatting and transmitting the message over the internet with respect to encryption, authentication, integrity and application.

The HTTP (Hypertest Transfer Protocol) and HTTPS (hypertext secure or HTTP over SSL|TLS) are two commonly used internet communication protocols.

Aspect	HTTP	HTTPS
① Encryption	Not encrypted data is transmitted in plain text, making it vulnerable to caves dropping.	encrypted using SSL/TLS. protects data from interception during transmission.
② Authentication.	No mechanism to verify the identify of website.	Uses digital certificate to authenticate the websites.
③ Integrity	Data can be modified or corrupted during transmission without detection.	ensures data integrity, any tampering is detected and rejected.

(4) Application	used for general browsing where security is not of concern.	used for secure application like banking online shopping, logic forms & data transfer.

Q.3. Identify and explain the different methods to retreive the resource, create new resource, update the resource, modify the resource and delete the resource while using the REST style.

In REST (Representational State Transfer) we use specific HTTP methods to perform actions on resources (like databases, entities, files, or user profiles). Here's how each method works

1) GET - Retreiver a resource
   purpose : fetch data from the server
   usage : does not modify anything
   Idempotent : Yes

example :
GET/users → returns a list of users
GET/users/101 → returns users with ID 101

2.) POST - create a new resource

purpose : send the data to the server to create something new.

usage : submits data, and server generates an ID for the new resource.

Idempotent : No.

example :
POST /users with body.'

3.) PUT - update (replace) a resource

purpose : replace an entire resource

usage : the resource is overwritter entirely

Idempotent : Yes

example :
PUT /users/ 101 with body.

4.) PATCH - modify (partial update)

purpose : partially update a resource (only for specific fields)

usage : changes only the fields provided.

Idempotent : usually yes.

example :
PATCH /users/101 with body.

5.) DELETE - remove a resource.

purpose : permanently deletes a resource

usage : removes the specified data from server

Idempotent : Yes

example :
DELETE /users/ 101