#### ****Question 4 :****Write a technical note on RESTful API clearly stating principals it works upon ?

**RESTful API Principles:**

1. **Stateless**: Each request is independent.
2. **Client-Server**: Separation of concerns.
3. **Uniform Interface**: Standard endpoints.
4. **Cacheable**: Responses can be cached.
5. **CRUD Operations**: Mapped to HTTP verbs (GET, POST, PUT, DELETE).

Any supporting illustration/Architecture plan / code snippet will make you close to full rewards.

**Client (Browser/Mobile App)**

**|--> GET /api/users → List Users**

**|--> POST /api/users → Add User**

**|--> PUT /api/users/1 → Update User**

**|--> DELETE /api/users/1 → Remove User**

**Server (Express/Node)**

Extra credit: Do you know anything comparable/better than RESTful API ? Briefly explain.

1. **GraphQL**: More flexible data queries.
2. **gRPC**: Faster binary protocol for microservices.

Q2:You are building a simple RESTful API using Node.js and Express. The API should have two routes:

1. GET /api/users - Fetches a list of all users.

2. POST /api/users - Adds a new user to the list. Your task is to modify the POST route to: ✓ Ensure that the email provided in the request body is valid (it should contain both @ and .). ✓ If the email is valid, the user should be added to the list and a success message should be returned. ✓ If the email is invalid, return an error message saying "Invalid email address".

**app.post("/api/users", (req, res) => {**

**const { name, email } = req.body;**

**const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;**

**if (!emailRegex.test(email)) {**

**return res.status(400).json({ status: "error", message: "Invalid email address" });**

**}**

**const newUser = { id: users.length + 1, name, email };**

**users.push(newUser);**

**res.status(201).json({ status: "success", user: newUser });**

**});**

Q3: My question is why 'header' tags looks different in syntax from react components 'Nav' , 'Article Entry ' and 'article’ which seems to use Curley braces?

1. **Header** is an **HTML tag** → used as <header>
2. **Nav, ArticleEntry, Article** are **React components** → used as <Nav />
3. JSX allows mixing of both, but component names must start with capital letters.

Q4: Briefly explain the line purpose and modify the rendering to conditional based on user login state.

const user = useAuthentication();

return (

<>

{user ? <Nav articles={articles} setArticle={setArticle} /> : <p>Please log in to see articles.</p>}</>

);

Q5: Write a technical note on web application using CRUD operations

**CRUD in Web Apps:**

1. **C**reate → POST
2. **R**ead → GET
3. **U**pdate → PUT/PATCH
4. **D**elete → DELETE

