# 🌐 Web Architecture Quiz – React + Flask + HTTP + APIs

## Quiz Questions

### Section 1 – Concepts

1. What is the role of React in your project?  
 a) Rendering HTML templates from Flask  
 b) Handling UI and forms in the browser  
 c) Storing data in MongoDB  
 d) Training ML models

2. What is the role of Flask in your project?  
 a) Rendering and styling your table UI  
 b) Acting as a JSON API layer between React, the database, and Notion  
 c) Managing Wi-Fi connections  
 d) Replacing MongoDB

3. Why is WTForms not needed in your architecture?

4. True or False: In your system, React and Flask both render the UI.

### Section 2 – HTTP + Networking

5. What is an HTTP request?  
 a) A JavaScript function  
 b) A letter-like message sent from a client to a server, structured with method, headers, and optional body  
 c) A way to style web pages  
 d) A database query

6. When React sends a fetch() call to Flask, what network layers carry that request? Order them:  
 - TCP  
 - HTTP  
 - IP  
 - Physical medium (Wi-Fi/Ethernet)

7. What does it mean when we say “HTTP over TCP/IP”?

8. What role does the browser play in this context?

### Section 3 – APIs

9. What is an API in the context of web development?  
 a) A Python library for ML  
 b) A database table schema  
 c) A set of defined HTTP endpoints with request/response formats  
 d) A styling framework

10. How does Flask serve as both an API provider and an API consumer in your project?

11. Fill in the blanks:  
 - React sends JSON to Flask → Flask \_\_\_\_\_\_ the data, \_\_\_\_\_\_ it to the DB/Notion/ML, then \_\_\_\_\_\_ JSON back to React.

### Section 4 – Application Flow

12. Put the following steps in order for what happens when you enter a task and hit “Submit”:  
 - Flask validates JSON  
 - React makes POST /api/tasks  
 - Browser transmits HTTP request via TCP/IP  
 - Flask responds with confirmation JSON  
 - React updates UI with success message

13. True or False: The physical medium carrying HTTP requests could be Wi-Fi, Ethernet cables, or fiber optics.

### Bonus (Critical Thinking)

14. Why is it considered best practice today to let React handle the UI and Flask only handle APIs, instead of Flask rendering HTML directly?

## Extended Response Questions (HTTP, TCP, IP)

1. Explain how TCP differs from IP.  
 - Include what job each one does in delivering data between computers.  
 - Why do we need both?

2. Describe what happens when React calls fetch("/api/tasks").  
 - Start from the JavaScript call in the browser,  
 - Explain how the browser builds the HTTP request,  
 - How TCP/IP ensures it gets to Flask,  
 - And how Flask reconstructs the request.

3. Why is HTTP called a “stateless protocol”?  
 - What does statelessness mean in practice for a web application?  
 - How do things like cookies or tokens add “state” back in when needed?

4. Draw an analogy between HTTP/TCP/IP and the real world.  
 - Identify what corresponds to the roads, the vehicles, the letter, and the language inside the letter.

5. What are the roles of “port numbers” in TCP/IP communication?  
 - Why might Flask default to port 5000 or 5001?  
 - How does the browser know where to deliver the HTTP request on a given machine?

## Answer Key

Section 1 – Concepts  
1. b  
2. b  
3. WTForms is unnecessary since React handles forms.  
4. False

Section 2 – HTTP + Networking  
5. b  
6. Physical → IP → TCP → HTTP  
7. HTTP rides on top of TCP/IP transport.  
8. Browser = runtime + messenger for React.

Section 3 – APIs  
9. c  
10. Provider to React, consumer of Notion/MongoDB.  
11. Validate, forward, return.

Section 4 – Application Flow  
12. React POST → Browser TCP/IP → Flask validate → Flask respond → React update UI  
13. True

Bonus  
14. Separation of concerns: React UI, Flask API = modern best practice.