1. What is backend development and how is it different from frontend development?
2. Mention three backend programming languages.
3. What is a server? Explain its role in backend development.
4. Define an API and explain its purpose.
5. What is a database and why is it important in backend systems?
6. List two differences between SQL and NoSQL databases.
7. What is a backend framework? Mention one example.
8. Explain what HTTP is and why it is important.
9. Mention two responsibilities of a backend developer.
10. What does CRUD stand for? Explain each word.
11. What is authentication and why is it important for backend applications?
12. What is the difference between a GET request and a POST request?

Answer to section 1

What is backend development and how is it different from frontend development?
Backend development refers to the server-side of web development, focusing on the logic, database integration, and server configuration. It's different from frontend development, which focuses on the client-side, dealing with user interface, user experience, and client-side logic.

Three backend programming languages:
1. Python
2. Java
3. Node.js (JavaScript)

What is a server? Explain its role in backend development.
A server is a computer or system that provides services, data, or resources to other computers or devices over a network. In backend development, a server hosts the application, handles requests, and provides data to clients.

Define an API and explain its purpose.
An API (Application Programming Interface) is a set of defined rules and protocols that enables different applications to communicate with each other. Its purpose is to provide a standardized interface for data exchange and integration.

What is a database and why is it important in backend systems?
A database is a collection of organized data stored in a way that allows for efficient retrieval and manipulation. It's crucial in backend systems for storing, managing, and retrieving data.

Two differences between SQL and NoSQL databases:
1. SQL databases are relational, while NoSQL databases are non-relational/document-based.
2. SQL databases have fixed schemas, while NoSQL databases have dynamic or flexible schemas.

What is a backend framework? Mention one example.

A backend framework is a software framework that provides a structure for building web applications, simplifying tasks like routing, templating, and database integration. Example: Express.js (Node.js)

Explain what HTTP is and why it is important.
HTTP (Hypertext Transfer Protocol) is a protocol for transferring data over the web. It's essential for communication between clients and servers, enabling data exchange and web functionality.

Two responsibilities of a backend developer:
1. Designing and implementing server-side logic
2. Integrating with databases and APIs

What does CRUD stand for? Explain each word.
CRUD stands for:
- Create: Creating new data or resources
- Read: Retrieving existing data or resources
- Update: Modifying existing data or resources
- Delete: Deleting data or resources

What is authentication and why is it important for backend applications?
Authentication is the process of verifying user identity, typically through credentials like username and password. It's crucial for securing applications, controlling access, and protecting user data.

What is the difference between a GET request and a POST request?
A GET request retrieves data from a server, while a POST request sends data to a server for processing or storage. GET requests are typically used for fetching data, while POST requests are used for creating or updating data.

Section B:
JavaScript variables & Datatypes

What is a variable in JavaScript?
A variable is a container that stores a value, allowing you to reference and manipulate it in your code.

Three ways to declare variables in JavaScript:
1. `var x = 10;`
2. `let y = 20;`
3. `const z = 30;`

Difference between let and const:
- `let` declares a variable that can be reassigned.
- `const` declares a constant variable that cannot be reassigned.

Seven primitive datatypes in JavaScript:

1. String
2. Number
3. Boolean
4. Null
5. Undefined
6. Symbol
7. BigInt

Datatype of the value: true
Boolean

Datatype of a JavaScript array
Object

Example of a string in JavaScript
`"hello world"`

Example of a number in JavaScript
`42`

Output of this expression? typeof "123"
"string"

Difference between null and undefined:
- `null` represents a deliberate absence of value, often assigned explicitly.
- `undefined` represents an uninitialized or non-existent variable/property, often the default state.

Section C:
Coding.

1. Declare a variable using let and assign your name to it.
```
let name = "John Doe";
```
2. Write JavaScript code to add two numbers and log the result.
```
let num1 = 10;
let num2 = 20;
let sum = num1 + num2;
console.log(sum); // Output: 30
```
3. Create an object called student with properties: name, age, and department.
```
let student = {
  name: "Jane Doe",
  age: 20,
  department: "Computer Science"
};
```
4. Write a JavaScript function called greet() that prints "Hello World".
```
function greet() {
  console.log("Hello World");
}
```

```
greet(); // Output: Hello World
```
5. Write a program that checks if a number is even or odd.
```
let num = 10;
if (num % 2 === 0) {
  console.log("Even");
} else {
  console.log("Odd");
}
```
6. Create an array of 5 colors and print the first color.
```
let colors = ["Red", "Green", "Blue", "Yellow", "Purple"];
console.log(colors[0]); // Output: Red
```
7. Write a function that returns the square of a number passed into it.
```
function square(num) {
  return num * num;
}
console.log(square(5)); // Output: 25
```
8. Write a small code snippet that converts a string to a number.
```
let value = "42";
let numValue = Number(value);
console.log(numValue); // Output: 42
// or
let numValue2 = parseInt(value);
console.log(numValue2); // Output: 42
```