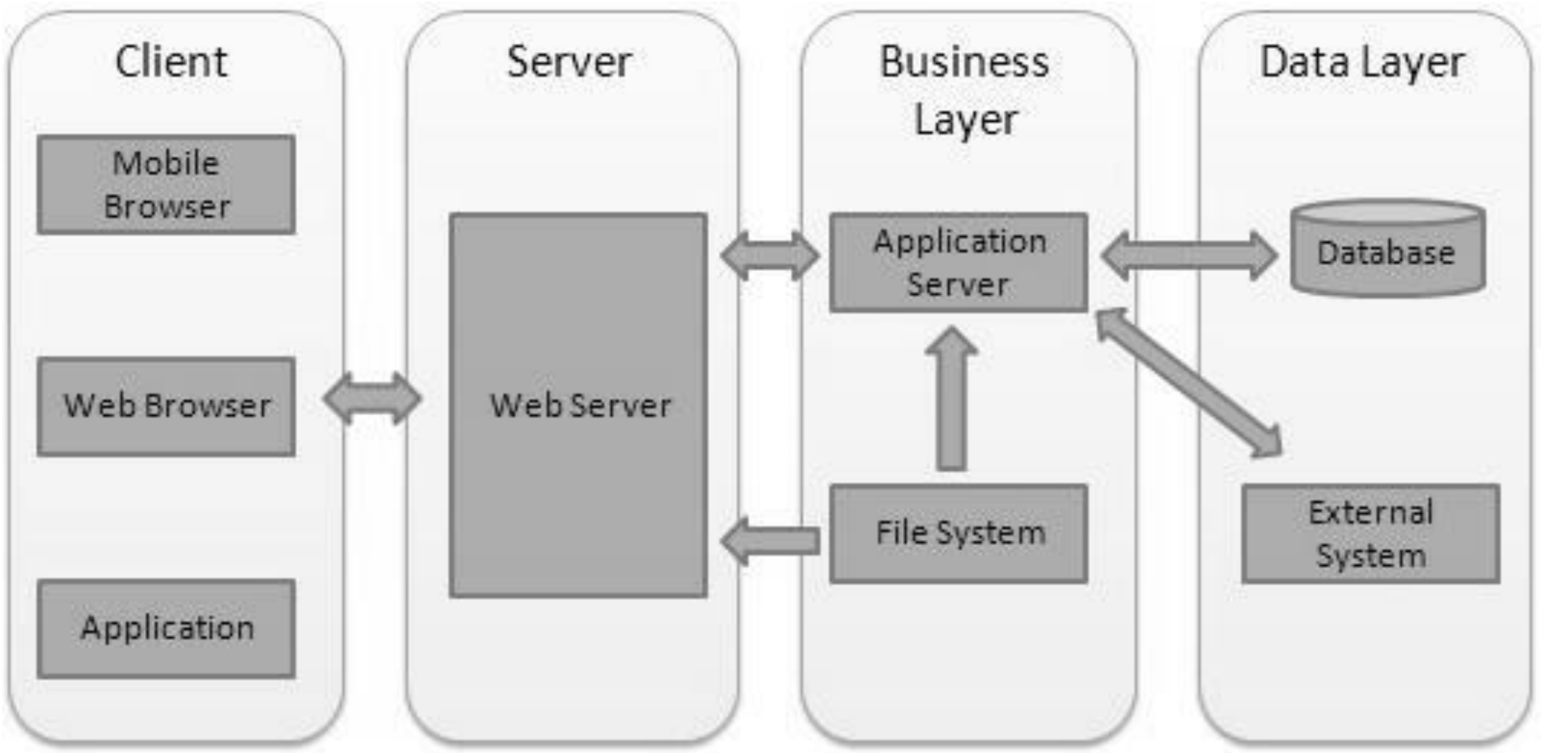
Web Server

What is Web Server?

- A Web Server is a software application (& hardware) which handles HTTP requests sent by the HTTP clients (like web browsers), and returns HTTP Response (i.e. web pages) in response to the clients' requests.
- Web servers usually respond with html documents along with images, style sheets, and scripts.
- Most of the web servers redirect to an application server which performs the specific tasks like getting data from the database, perform complex logic etc. and then sends a result back to the HTTP client through a Web server.
- Popular web servers – Apache Tomcat, Microsoft IIS.

What is Web Server?

Typical multi-layer / multi-tier architecture



Node Web Module

- Node.js provides capabilities to create your own web server which handles HTTP requests asynchronously.

- Node.js has a 'built-in module' called http, which allows Node.js to transfer data over HTTP (Hyper Text Transfer Protocol).

- To include the HTTP module, use the require() method:

```
var http = require('http’);
```

- The HTTP module creates an HTTP server that listens to server port and gives a response back to the client.

Node Web Module

- Use the createServer() method to create an HTTP server.
- The function passed into the http.createServer() method, will be executed when someone tries to access the browser on the configured port using http protocol.
- In the res.writeHead() method,
  - → 1st parameter is the HTTP status code (200 indicates OK),
  - → 2nd parameter is an object containing the response headers.

```
var http = require('http');

//create a server object
http.createServer(function (req, res) {
    res.writeHead(200, { 'Content-Type': 'text/html' });
    res.write("Hello World!"); //write a response to the client
    res.end(); //end the response
}).listen(8080); //the server object listens on port 8080

console.log('Node.js web server is running on port 8080...');
```

Handle HTTP requests

- The http.createServer() method includes request and response parameters supplied by Node.js.
- The request object is used to get information about the current HTTP request e.g., url, request header, and data.
- The response object is used to send a response for a current HTTP request.

```
Run in browser:

http://localhost:8080/
http://localhost:8080/about
http://localhost:8080/contact
http://localhost:8080/abcd (this is invalid request)
```

Handle HTTP requests

```
var http = require('http');

var server = http.createServer(function (req, res) {
    if (req.url == '/') {
        res.writeHead(200, { 'Content-Type': 'text/html' });
        res.write('<html><body><h2>This is home page.</h2></body></html>');
        res.end();

    }
    else if (req.url == "/about") {
        res.writeHead(200, { 'Content-Type': 'text/html' });
        res.write('<html><body><h2>This is about us page.</h2></body></html>');
        res.end();
    }
    else if (req.url == "/contact") {
        res.writeHead(200, { 'Content-Type': 'text/html' });
        res.write('<html><body><h2>This is contact us page.</h2></body></html>');
        res.end();
    }
    else {
        res.writeHead(400, 'Invalid request' });
        res.end('<html><body><h2>Invalid Request!</h2></body></html>');
    }
});
server.listen(8080);
console.log('Node.js web server is running on port 8080...');
```

Request URL

- The callback function passed into the http.createServer() has a req parameter that represents the request from the client.
- This object has a property called 'url' which holds the part of the url that comes after the domain & port.

```
var http = require('http');

//create a server object
http.createServer(function (req, res) {
    res.writeHead(200, { 'Content-Type': 'text/html' });
    res.write(req.url);
    res.end();
}).listen(8080);

console.log('Node.js web server is running on port 8080...');
```

Run in browser: http://localhost:8080/**bmcc**

Parse query string

- The url module splits the query string into multiple readable parts.

```
var http = require('http');
var url = require('url');

http.createServer(function (req, res) {
    res.writeHead(200, { 'Content-Type': 'text/html' });
    var q = url.parse(req.url, true).query;
    res.write("Month: " + q.month);
    res.write("<br>");
    res.write("Year: " + q.year);
    res.end();
}).listen(8080);
```

Run in browser: `http://localhost:8080/?year=2024&month=January`

Sending JSON Response

- The following example demonstrates how to serve JSON response from the Node.js web server.

```
var http = require('http');

var server = http.createServer(function (req, res) {
    if (req.url == '/data') {
        res.writeHead(200, { 'Content-Type': 'application/json' });
        res.write(JSON.stringify({ message: "Hello World" }));
        res.end();
    }
});
server.listen(8080);
console.log('Node.js web server is running on port 8080...');
```

Sending JSON Response

- The following example demonstrates how to serve JSON response from the Node.js web server.

```
// json object
var jsonData = {
    "subjects": [
        { "name": "Node.js", "marks": "22" },
        { "name": "Java", "marks": "20" },
        { "name": "PHP", "marks": "21" }
    ]
};

res.writeHead(200, { 'Content-Type': 'application/json' });
for (i = 0; i < jsonData.subjects.length; i++) {
    res.write(jsonData.subjects[i].name);
    res.write('\t');
    res.write(jsonData.subjects[i].marks);
    res.write('\n');
}
```

Home Work

- What is a web server? Explain the steps to create a web server using Node.js (with code snippet)

- Explain below concepts pertaining to Node.js web server:
  - → Handling HTTP requests
  - → Handling query string
  - → Sending HTTP response
  - → Sending JSON response

  (with code snippet)

Write Node.js application(s) to perform the required processing and display the result on a web page.

- Accept a number as a query string parameter and display the table.
- Accept word as a query string parameter and display whether it is palindrome or not.
- Accept word as a query string parameter and display it in the reverse order.
- Accept word as a query string parameter and display vowel count.
- Accept 2 numbers and an operation as query string parameters and display the result by performing the corresponding operation.