Q1

Index.html----------

```
<!DOCTYPE html>
<html>
<head>
  <title>Web Server</title>
</head>
<body>
  <h1>Welcome to server page!</h1>
  <form action="/" method="post">
    <input type="text" name="message" />
    <button type="submit">Submit</button>
  </form>
</body>
</html>
```

Server.js----------

```
const http = require('http');
const fs = require('fs');

const port = 8000;

function serveStaticFile(res, filename, contentType) {
  fs.readFile(filename, (err, data) => {
    if (err) {
      res.writeHead(500, { 'Content-Type': 'text/plain' });
      res.end('Internal Server Error');
    } else {
      res.writeHead(200, { 'Content-Type': contentType });
      res.end(data);
    }
  });
}

const server = http.createServer((req, res) => {
  if (req.method === 'GET') {
    if (req.url === '/') {

      serveStaticFile(res, './index.html', 'text/html');
    }
  }

 else if (req.method === 'POST') {
    if (req.url === '/') {
```

```
    let body = '';
    req.on('data', (chunk) => {
      body += chunk;
    });

    req.on('end', () => {


      res.writeHead(200, { 'Content-Type': 'text/plain' });
      res.end('Hello, this is from POST request : ' + body);
    });
  } else {

    res.writeHead(404, { 'Content-Type': 'text/plain' });
    res.end('404 Not Found');
  }
 } else {

  res.writeHead(101, { 'Content-Type': 'text/plain' });
  res.end('101 Not Implemented');
 }
});

server.listen(port, () => {
 console.log(`Server is running on http://localhost:${port}`);
});
```

Q2

Index.html---------

```
<!-- Get Complete Source Code from Pabbly.com -->
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <meta http-equiv="X-UA-Compatible" content="ie=edge">
   <title>Document</title>
</head>
<body>
  <h1>Hello nodejs!</h1>
</body>
</html>
```

Index2.html------------------------

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
   <h1>HTML page using AJAX call.</h1>
   <button id="ajaxButton">Click</button>
   <div id="ajaxResponse"></div>
 </body>
 <script>
```

```
 document.getElementById("ajaxButton").addEventListener("click", () => {
      fetch("/gethello").then(response=>response.text()).then(data => {
        document.getElementById("ajaxResponse").innerHTML = data;
   }).catch(error => {
    console.error(error)
   });
 });
 </script>
</html>
server.js-------------------------
const express = require("express")
const app = express()
const fs = require("fs")

app.get("/",(req,res)=>{
   fs.readFile("./index2_1.html","utf8",(err,data)=>{
      res.send(data)
   })
})
app.get("/gethello",(req,res)=>{
   fs.readFile("./index2.html","utf8",(err,data)=>{
      res.send(data)
   })

})

app.listen(8000,()=>{
   console.log(`Server is running on http://localhost:8000`)
})
```

Q3---

Server.js

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

var r1 = readline.createInterface(process.stdin, process.stdout);
r1.setPrompt("You==>");
r1.prompt();
r1.on('line', function(message) {
   console.log('Bot ==> '+ reply(message));
   //console.log('Bot ==>  '+ message);

   r1.prompt();
}).on('close',function(){  //chaining events.
   process.exit(0);
});

function reply(message)
{
   this.Bot_Age = 25;
                  this.Bot_Name = "name1";
                  this.Bot_University = "VNSGU";
                  this.Bot_Country = "India";

                  message= message.toLowerCase()

                  if(message.indexOf("hi") > -1 ||
                        message.indexOf("hello") > -1 ||
                        message.indexOf("welcome") > -1 )
```

```
        {
            return "Hi!";
        }
        else if(message.indexOf("age") > -1 &&
            message.indexOf("your"))
        {
            return "I'm " + this.Bot_Age;
        }
        else if (message.indexOf("how") > -1 &&
            message.indexOf("are") &&
            message.indexOf("you"))
        {
            return "I'm fine ^_^"
        }
        else if(message.indexOf("where") > -1
            && message.indexOf("live") &&
            message.indexOf("you"))
        {
            return "I live in " + this.Bot_Country;
        }
        return "Sorry, I didn't get it :( ";

}
```

Q4

Index.html--------------------------

```
<!DOCTYPE html>
<html>
<head>
   <title>Chatbot Module</title>
</head>
<body>
   <h1>Chatbot Module</h1>

   <input type="text" id="messageInput" placeholder="Type your message">
   <button onclick="sendMessage()">Send</button>

   <div id="chatLog"></div>

   <script>
      const socket = new WebSocket('ws://localhost:8080');
      const messageInput = document.getElementById('messageInput');
      const chatLog = document.getElementById('chatLog');

      socket.addEventListener('open', (event) => {
         logMessage('Connected to server');
      });

      socket.addEventListener('message', (event) => {
         logMessage('Chatbot says: ' + event.data);
      });

      function sendMessage() {
```

```
            const message = messageInput.value;
            socket.send(message);
            logMessage('You: ' + message);
            messageInput.value = '';
        }

        function logMessage(message) {
            const messageElement = document.createElement('p');
            messageElement.textContent = message;
            chatLog.appendChild(messageElement);
        }
    </script>
</body>

server.js----------
const http = require('http');
const WebSocket = require('ws');

// Create HTTP server
const server = http.createServer((req, res) => {
    res.writeHead(200, {'Content-Type': 'text/plain'});
    res.end('Chatbot server is running\n');
});

// Create WebSocket server
const wss = new WebSocket.Server({ server });

wss.on('connection', (ws) => {
    console.log('New client connected');
```

```
    ws.on('message', (message) => {
        console.log(`Received: ${message}`);
        ws.send(` ${message}`);
    });

    ws.on('close', () => {
        console.log('Client disconnected');
    });
});

server.listen(8080, () => {
    console.log('Server is running on port http://localhost:8080');
});
```

Q5

Server.js

```
var fs = require('fs')
var zlib = require('zlib')



fs.createReadStream('./test.txt')
   .pipe(zlib.createGzip())
   .pipe(fs.createWriteStream('./Ziped_file/test.txt.gz'));



console.log('File compressed..!!');
```

Q6

Server.js----------

```
var fs = require('fs')
var unzip = require('zlib')



fs.createReadStream('../Q5/Ziped_file/test.txt.gz')
   .pipe(unzip.createGunzip())
   .pipe(fs.createWriteStream('./Unziped_file/test.txt'));



console.log('File Decompressed..!!');
```

Q7

```
var fs = require('fs/promises')

function readFile(fpath)
{
   return new Promise(function(success,fail)
   {
      fs.unlink(fpath,(err,data) =>
      {
         if(err)
            fail(err)
         else
            success(data)
      })
   })
}

readFile('./Q6/Unzipedtest.txt').then((data)=>{
   console.log(data)
}).catch((err)=>{
   console.log(err)
})
```

Q8

```
const fetch = (...args) => import('node-fetch').then(({default: fetch}) => fetch(...args));

async function asyncajaxawait()
{
  const res = await fetch('https://www.google.com/')
  console.log(res);
}

asyncajaxawait();
```

Q9

```
const mysql = require('mysql');



const dbConfig = {
  host: 'localhost',
  user: 'root',
  password: '',
  database: 'empdb',
};



function connectToDatabase(config) {
  return new Promise((resolve, reject) => {
    const connection = mysql.createConnection(config);
    connection.connect((err) => {
      if (err) {
        reject(err);
      } else {
        resolve(connection);
      }
    });
  });
}



function insertEmployee(connection, employee) {
  return new Promise((resolve, reject) => {
    connection.query('INSERT INTO emp SET ?', employee, (err, result) => {
```

```
      if (err) {
        reject(err);
      } else {
        resolve(result);
      }
    });
  });
}

function getAllEmployees(connection) {
  return new Promise((resolve, reject) => {
    connection.query('SELECT * FROM emp', (err, results) => {
      if (err) {
        reject(err);
      } else {
        resolve(results);
      }
    });
  });
}

async function main() {
  try {

    const connection = await connectToDatabase(dbConfig);
    console.log('Connected to the database!');
```

```
    const newEmployee = {
      name: 'Test Abc',
      salary: 60000,
    };
    const insertResult = await insertEmployee(connection, newEmployee);
    console.log('New employee added with ID:', insertResult.insertId);



    const allEmployees = await getAllEmployees(connection);
    console.log('All employees:', allEmployees);



    connection.end();
    console.log('Database connection closed.');
  } catch (error) {
    console.error('Error:', error);
  }
}



main();
```

Q10

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

const PORT = 8000;

const server = http.createServer((req, res) => {

  res.setHeader('Content-Type', 'text/plain');



  if (req.url === '/') {
    res.end('Hello, this is the homepage!');
  } else if (req.url === '/about') {
    res.end('This is the about page.');
  } else {
    res.statusCode = 404;
    res.end('Page not found.');
  }
});

server.listen(PORT, () => {
  console.log(`Server running on port http://localhost:${PORT}`);
});
```

Q11

```
<!DOCTYPE html>
<html>
 <head>
  <title>Live Cricket Score</title>
 </head>
 <body>
  <h1>Live Cricket Score</h1>
  <table>
   <tr>
    <th>Player Name</th>
    <th>Score (Runs)</th>
   </tr>
   <tr id="player1">
    <td>Player 1</td>
    <td id="player1-score">0</td>
   </tr>
   <tr id="player2">
    <td>Player 2</td>
    <td id="player2-score">0</td>
   </tr>
  </table>

  <script>
   const player1ScoreElement = document.getElementById('player1-score');
   const player2ScoreElement = document.getElementById('player2-score');

   // Create a WebSocket connection
   const ws = new WebSocket('ws://localhost:8080');
```

```
    // Handle messages from the server
    ws.onmessage = event => {
      const { player1, player2 } = JSON.parse(event.data);
      player1ScoreElement.textContent = player1;
      player2ScoreElement.textContent = player2;
    };
  </script>
 </body>
</html>
```

server.js

```
const WebSocket = require('ws');


const wss = new WebSocket.Server({ port: 8080 });


let score = {
 player1: 22,
 player2: 50
};

// Broadcast the score to all clients
const broadcastScore = () => {
 wss.clients.forEach(client => {
   if (client.readyState === WebSocket.OPEN) {
     client.send(JSON.stringify(score));
```

```
    }
  });
};


wss.on('connection', ws => {
  // Send the current score to the newly connected client
  ws.send(JSON.stringify(score));

  // Handle messages from clients
  ws.on('message', message => {
    try {
      const playersData = JSON.parse(message);
      playersData.forEach(({ player, runs }) => {
        score[player] = runs;
      });
      broadcastScore();
    } catch (error) {
      console.error('Invalid message:', message);
    }
  });
});
```