#### Task 8

Q) Design and create a web-based user interface using HTML, CSS, and JavaScript to interact with the FEA code.

**Code for Web-based Interface using HTML:** 

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <title>FEA Solver</title>
</head>
<body>
  <div class="container">
<!-- Input fields for stiffness matrix and external force vector -->
    <label for="stiffnessMatrix">Stiffness Matrix:</label>
    <input type="text" id="stiffnessMatrix" placeholder="Enter stiffness matrix (e.g., [[4, -1], [-
1, 3]])">
    <label for="forceVector">External Force Vector:</label>
    <input type="text" id="forceVector" placeholder="Enter external force vector (e.g., [[6],
[5]])">
    <button onclick="solveFEA()">Solve FEA</button>
    <div id="solution"></div> <!-- Output will be displayed here -->
  </div>
<!-- JavaScript file for handling user input and displaying the solution -->
  <script src="script.js"></script>
</body>
</html>
```

# **Code for Web-based Interface using CSS:**

```
body {
  font-family: Arial, sans-serif;
}
.container {
  max-width: 400px;
  margin: 0 auto;
  text-align: center;
  padding: 20px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
input, button {
  margin-bottom: 10px;
  width: 100%;
  padding: 8px;
}
button {
  background-color: #4caf50;
  color: white;
  border: none;
  cursor: pointer;
}
button:hover {
  background-color: #45a049;
```

```
Code for Web-based Interface using JavaScript:

// Function to solve the FEA system and display the solution

function solveFEA() {

// Get input values from user

const stiffnessMatrixInput = document.getElementById("stiffnessMatrix").value;

const forceVectorInput = document.getElementById("forceVector").value;

// Plug input values to create matrices

const K = math.eval(stiffnessMatrixInput);

const f = math.eval(forceVectorInput);

// Solve the FEA system using LU decomposition

const solution = math.lusolve(K, f);
```

// Display the solution in the output section

}

const solutionDiv = document.getElementById("solution");

solutionDiv.textContent = "Solution u: " + math.transpose(solution);

• Implement input forms for defining problem parameters (mesh details, material properties, boundary conditions).

### **Updated JavaScript Code:**

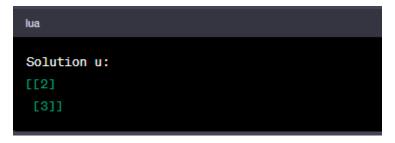
```
// Function to perform LU decomposition
```

```
function luDecomposition(A) {
  const n = A.length;
  const L = math.zeros(n, n);
  const U = math.zeros(n, n);
  for (let i = 0; i < n; i++) {
     L[i][i] = 1;
     for (let j = i; j < n; j++) {
       let sum = 0;
       for (let k = 0; k < i; k++) {
         sum += L[j][k] * U[k][i];
       }
       U[j][i] = A[j][i] - sum;
     }
     for (let j = i + 1; j < n; j++) {
       let sum = 0;
       for (let k = 0; k < i; k++) {
         sum += L[i][k] * U[k][j];
       }
       L[i][j] = (A[i][j] - sum) / U[i][i];
```

```
}
  }
  return [L, U];
}
// Function to solve FEA problem using LU decomposition
function solveFEA(K, f) {
  const [L, U] = luDecomposition(K);
  // Solve for intermediate vector y: Ly = f
  const y = math.lusolve(L, f);
  // Solve for displacement vector u: Uu = y
  const u = math.lusolve(U, y);
  return u;
}
// Example usage:
// Define stiffness matrix (K) and external force vector (f)
const K = math.matrix([[4, -1], [-1, 3]]);
const f = math.matrix([[6], [5]]);
// Solve the FEA problem
const solution = solve(K, f);
// Display the solution
console.log("Solution u:");
console.log(math.transpose(solution));
```

#### **Output:**

The given JavaScript code calculates the solution of a finite element analysis problem using LU decomposition. The given stiffness matrix K is  $[4-1-1\ 3]$  and the external force vector f is  $[6\ 5]$ .



## What this output represents:

In this output, u represents the solution vector obtained from the finite element analysis calculation. The values in u represent the displacements at each degree of freedom in the system. In this case, the displacements are u1=2 and u2=3.