

WEB PROGRAMMING

ASSIGNMENT-11 (JavaScript)

Student Name: Avyakth Shriram

Register Number: 24BCE5461

Course Slot: TG1

Professor: Dr. Kanniga Devi

LAB 11 – TASK 1 - HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Bank loan approval system</title>
    <style>
      body{
        background-color:blanchedalmond;
      }
      .container{
        height: 60vh;
        width: 40vw;
        margin:auto;
        padding-top:20px;
        background-color:skyblue;
        border-radius: 1.5rem;
      }
      .formm{
        display: flex;
        flex-direction: column;
        gap:10px;
        margin: 20px;
      }
      #error{
        color:red;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <h1 style="text-align:center;margin:auto;height:50px;width:60%;background-
color:yellowgreen;border-radius:1rem;">Check Your Loan Eligibility</h1>
      <div>
        <form class="formm">
          AGE: <input type="text" id="age" class="num"/>
          SALARY: <input type="text" id="salary" class="num"/>
          Credit-Score: <input type="text" id="credit-score" class="num"/>
        </form>
        <div style="text-align:center;gap:20px;">
```

```

        <input type="button" onclick="submitted()" value="submit"
style="background-color:yellowgreen;border:none;height:30px;width:70px;border-
radius:2rem;"/>
        <p id="error"></p>
    </div>
</div>
</div>
<script type="text/javascript" src="Task1.js"></script>
</body>
</html>

```

LAB 11 – TASK 1 - JavaScript:

```

function submitted(){
    var age = Number(document.getElementById("age").value);
    var salary = Number(document.getElementById("salary").value);
    var Cred = Number(document.getElementById("credit-score").value);
    var inputs = document.querySelectorAll(".num");
    if(age < 21 || age>60){
        document.getElementById("error").innerHTML = "The age must be between 21 and
60";
        return;
    }
    else if(salary<50000){
        document.getElementById("error").innerHTML = "The salary must be greater than
50,000";
        return;
    }
    else if (Cred<700 && salary<100000){
        document.getElementById("error").innerHTML = "Either the salary must be greater
than 100000 or the Credit score of the customer must be greater than 700";
        return;
    }else{
        document.getElementById("error").innerHTML = "Document submitted
Successfully!";
    }
    for(let input of inputs){
        if(isNaN(input.value)){
            document.getElementById("error").innerHTML = `${input.id} must be a number`;
            return;
        }
    }
}

```

```
}  
}  
}
```

LAB 11 – TASK 1 - Screenshots:

File /Users/srivani/Downloads/24BCE5407%20(jan%2021)/24BCE5407%20(jan%2021)/LAB-11%20(1)/Task1/Task1.html

Check Your Loan Eligibility

AGE:
26

SALARY:
80000

Credit-Score:
721

submit

Document submitted Successfully!

LAB 11 – TASK 2 – HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <title>SMART TRAFFIC LIGHT SYSTEM</title>
    <link rel="stylesheet" type="text/css" href="Task2.css"/>
  </head>
  <body>
    <div class="container">
      <h1>Smart traffic light system</h1>
      <div class="inner-container">
        <form>
          Time(24hr-format): <input type="text" id="time" />
          Count(vehicles): <input type="text" id="count" />
        </form>
        <input value="submit" type="button" id="submit-btn" onclick="findTime()"/>
        <p id="output">Green Light is On for the next:<span id="req"></span></p>
      </div>
    </div>
    <script type="text/javascript" src="Task2.js"></script>
  </body>
</html>
```

LAB 11 – TASK 2 – JavaScript:

```
function findTime(){
  var time = Number(document.getElementById("time").value);
  var count = document.getElementById("count").value;
  if((time>=7 && time<=10) || (time>=17 && time <=20)){
    if(count<50){
      document.getElementById("req").innerHTML = "60";
    }else{
      document.getElementById("req").innerHTML = "80";
    }
  }
}
```

```
}  
}else if((time>24 || time<1)){  
    document.getElementById("output").innerHTML = "Invalid time input!";  
    return;  
}else{  
    if(count<50){  
        document.getElementById("req").innerHTML = "30";  
    }else{  
        document.getElementById("req").innerHTML = "50";  
    }  
}  
}  
}
```

LAB 11 – TASK 2 – CSS:

```
*{  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
    font-family: roboto;  
}  
  
body{  
    background-color: #1a1a2e;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    height: 100vh;  
    color: #ffffff;  
}  
  
.container{  
    background: #16213e;  
    padding: 2rem;  
    border-radius: 15px;  
    box-shadow: 0 10px 30px rgba(0, 0, 0, 0.5);  
    text-align: center;  
    width: 350px;  
    border: 1px solid #0f3460;  
}
```

```
h1{
  font-size: 1.5rem;
  margin-bottom: 1.5rem;
  color: #4ecca3;
  text-transform: uppercase;
  letter-spacing: 1px;
}
```

```
.inner-container form {
  display: flex;
  flex-direction: column;
  gap: 15px;
  margin-bottom: 20px;
}
```

```
input {
  padding: 10px;
  border-radius: 5px;
  border: none;
  background: #0f3460;
  color: white;
  font-size: 1rem;
  outline: none;
  transition: 0.3s;
}
```

```
#submit-btn {
  width: 100%;
  padding: 12px;
  background-color: #4ecca3;
  color: #1a1a2e;
  border: none;
  border-radius: 5px;
  font-weight: bold;
  cursor: pointer;
  font-size: 1rem;
  transition: transform 0.2s, background-color 0.3s;
}
```

```
#submit-btn:hover {
  background-color: #45b291;
}
```

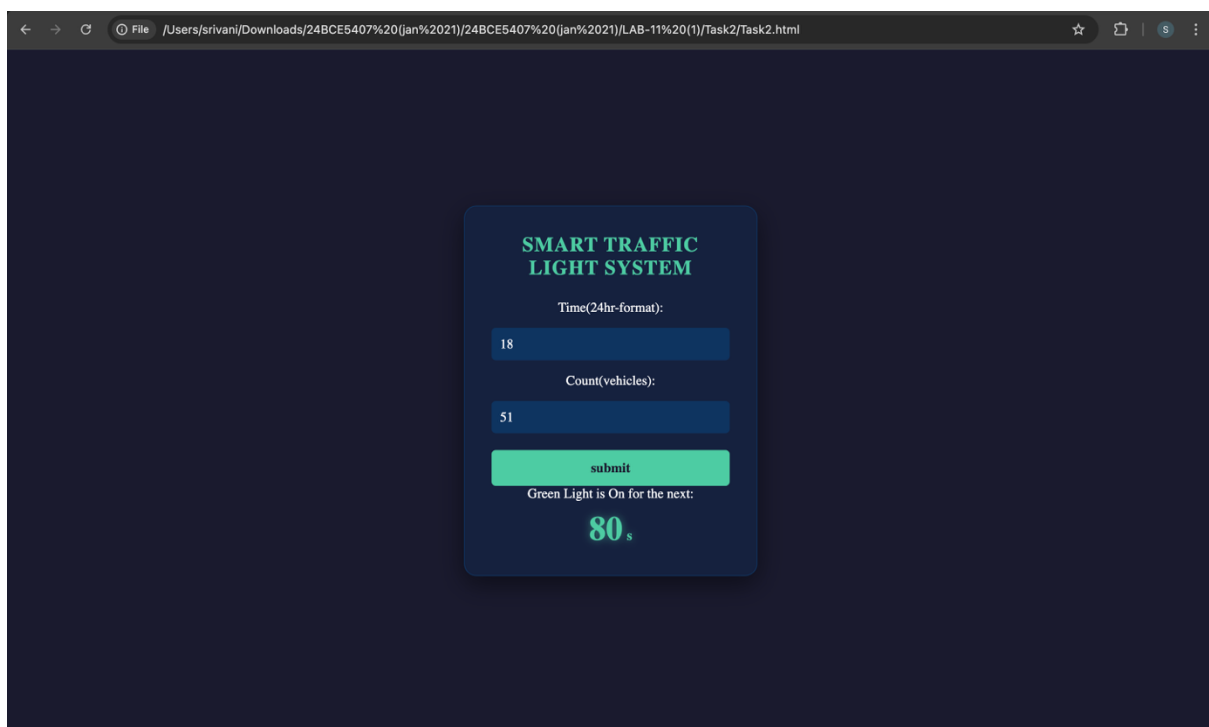
```
.output {
  margin-top: 20px;
  font-size: 1.1rem;
  color: #95a5a6;
}
```

```
}

#req{
  display: block;
  font-size: 2.5rem;
  font-weight: bold;
  color: #4ecca3;
  margin-top: 10px;
  text-shadow: 0 0 10px rgba(78, 204, 163, 0.5);
}

#req:not(:empty)::after{
  content: "s";
  font-size: 1rem;
  margin-left: 5px;
}
```

LAB 11 – TASK 2 – Screenshots:



LAB 11 - TASK3 – HTML:

```
<!DOCTYPE html>
<head>
  <title>Hotel Room Pricing System</title>
  <link rel="stylesheet" type="text/css" href="Task3.css"/>
</head>
<body>
  <div class="container">
    <h1>Hotel Room Pricing System</h1>
    <h3>Select the month:</h3>
    <form>
      <input type="radio" name="month" value="january">January
      <input type="radio" name="month" value="february">February
      <input type="radio" name="month" value="march">March
      <br/>
      <input type="radio" name="month" value="april">April
      <input type="radio" name="month" value="may">May
      <input type="radio" name="month" value="june">June
      <br/>
      <input type="radio" name="month" value="july">July
      <input type="radio" name="month" value="august">August
      <input type="radio" name="month" value="september">September
      <br/>
      <input type="radio" name="month" value="october">October
      <input type="radio" name="month" value="november">November
      <input type="radio" name="month" value="december">December
      <br>
    </form>
    <h3>Select the room type:</h3>
    <form>
      <input type="radio" name="room" value="standard">Standard
      <input type="radio" name="room" value="deluxe">Deluxe
      <input type="radio" name="room" value="suite">Suite
    </form>
    <input value="submit" type="button" id="submit-btn" onclick="findPrice()"/>
    <p id="output"></p>
  </div>
  <script type="text/javascript" src="Task3.js"></script>
</body>
```

LAB 11 – TASK 3 – JavaScript:

```
function findPrice() {
    const month = document.querySelector('input[name="month"]:checked').value;
    const room = document.querySelector('input[name="room"]:checked').value;
    const output = document.getElementById("output");
    if (!month || !room) {
        output.innerHTML = "Please select both a month and a room type.";
        return;
    }

    let price = 0;
    if (month === "june" || month === "july" || month === "august" || month ===
"december") {

        if (room === "standard") {
            price = 5000;
        } else if (room === "deluxe") {
            price = 8000;
        } else {
            price = 12000;
        }

    }

    else {

        if (room === "standard") {
            price = 3000;
        } else if (room === "deluxe") {
            price = 5000;
        } else {
            price = 8000;
        }

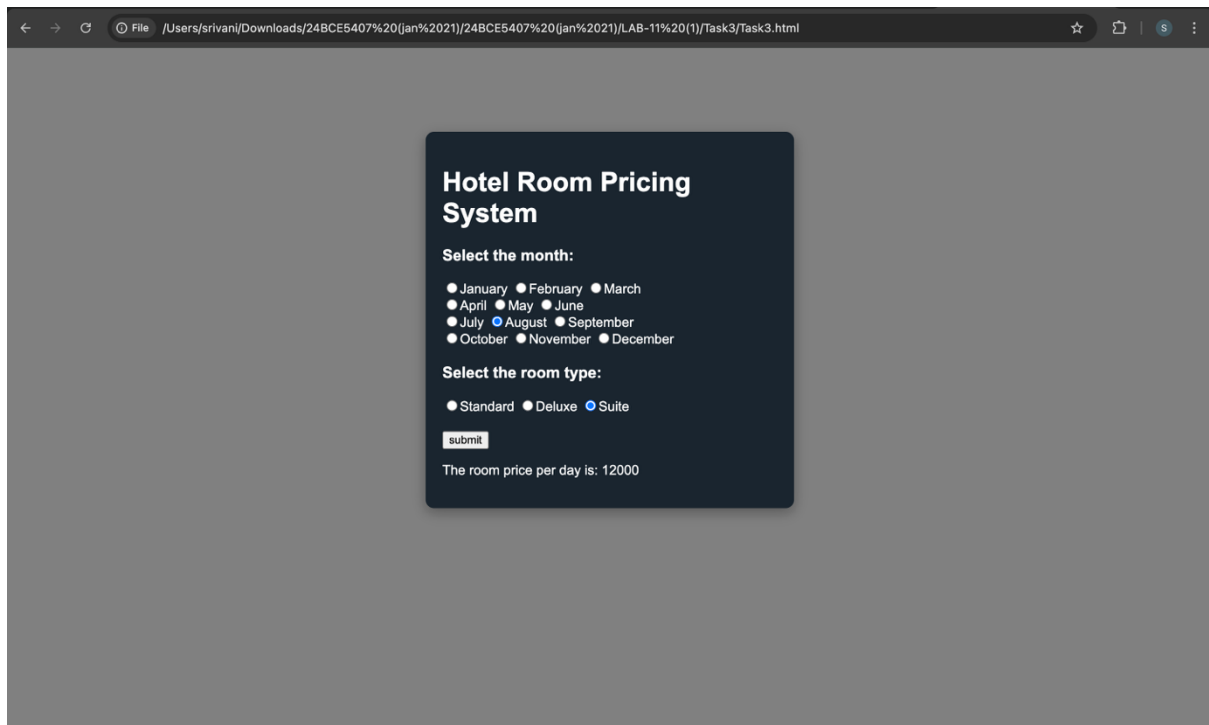
    }

    output.innerHTML = `The room price per day is: ${price}`;
}
```

LAB 11 – TASK 3 – CSS:

```
body{
  background-color:gray;
}
.container{
  background-color: #1a252f;
  width: 400px;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 5px 15px rgba(0,0,0,0.3);
  color: white;
  font-family: Arial, sans-serif;
  margin: 100px auto;
}
form{
  margin-bottom: 20px;
  text-align: left;
}
```

LAB 11 – TASK 3 – Screenshots:



LAB 11 – TASK 4 – HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Movie Ticket Calculator</title>
  <link rel="stylesheet" href="Task4.css">
</head>
<body>
  <div class="container">
    <h2>Movie Ticket Price</h2>
    <label for="age">Enter Age:</label>
    <input type="number" id="age" placeholder="e.g. 25">
    <label for="time">Show Time (24hr format):</label>
    <input type="number" id="time" placeholder="e.g. 14 for 2 PM" min="0" max="23">
    <button onclick="calculatePrice()">Calculate Price</button>
    <div id="result" class="result"></div>
  </div>
```

```
<script src="Task4.js"></script>
</body>
</html>
```

LAB 11 – TASK 4 – JavaScript:

```
function calculatePrice() {
  const age = parseInt(document.getElementById('age').value);
  const time = parseInt(document.getElementById('time').value);
  const resultDiv = document.getElementById('result');
  let price = 0;
  if (age < 12) {
    price = 100;
  } else if (age >= 60) {
    price = 120;
  } else {
    price = 200;
  }
  if (time < 17) {
    price = price * 0.5;
  }
  if (isNaN(age) || isNaN(time)) {
    resultDiv.innerHTML = "Please enter valid numbers.";
  } else {
    resultDiv.innerHTML = `Final Ticket Price: ₹${price}`;
  }
}
```

LAB 11 – TASK 4 – CSS:

```
body {
  font-family: Arial, sans-serif;
  display: flex;
  justify-content: center;
```

```
    align-items: center;
    height: 100vh;
    background-color: #f4f4f9;
}

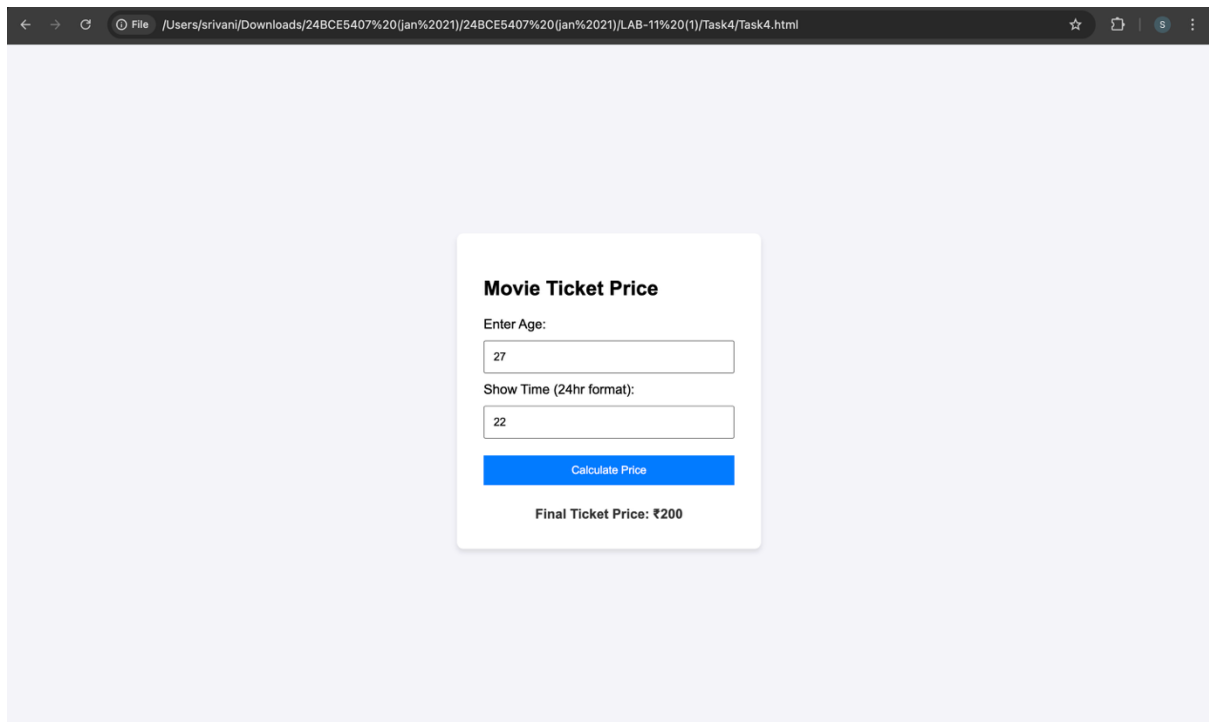
.container {
    background: white;
    padding: 2rem;
    border-radius: 8px;
    box-shadow: 0 4px 6px rgba(0,0,0,0.1);
    width: 300px;
}

input, button {
    width: 100%;
    padding: 10px;
    margin: 10px 0;
    box-sizing: border-box;
}

button {
    background-color: #007bff;
    color: white;
    border: none;
    cursor: pointer;
}

.result {
    margin-top: 15px;
    font-weight: bold;
    text-align: center;
    color: #333;
}
```

LAB 11 – TASK 4 – Screenshots:



LAB 11 – TASK 5 – HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Number Pyramid Generator</title>
    <link rel="stylesheet" href="Task5.css">
  </head>
  <body>
    <div class="container">
      <div class="card">
        <h3>Number Pyramid Generator</h3>
        <input type="number" id="levels" placeholder="Enter n (e.g., 5)">
        <button onclick="generatePyramid()">Generate</button>
        <p id="pyramidResult"></p>
      </div>
    </div>
    <script type="text/javascript" src="Task5.js"></script>
  </body>
</html>
```

LAB 11 – TASK 5 – JavaScript:

```
function generatePyramid() {  
  const n = parseInt(document.getElementById('levels').value);  
  let currentNumber = 1;  
  let output = "";  
  for (let i = 1; i <= n; i++) {  
    for (let j = 1; j <= i; j++) {  
      output += currentNumber + " ";  
      currentNumber++;  
    }  
    output += "\n";  
  }  
  
  document.getElementById('pyramidResult').innerText = output;  
}
```

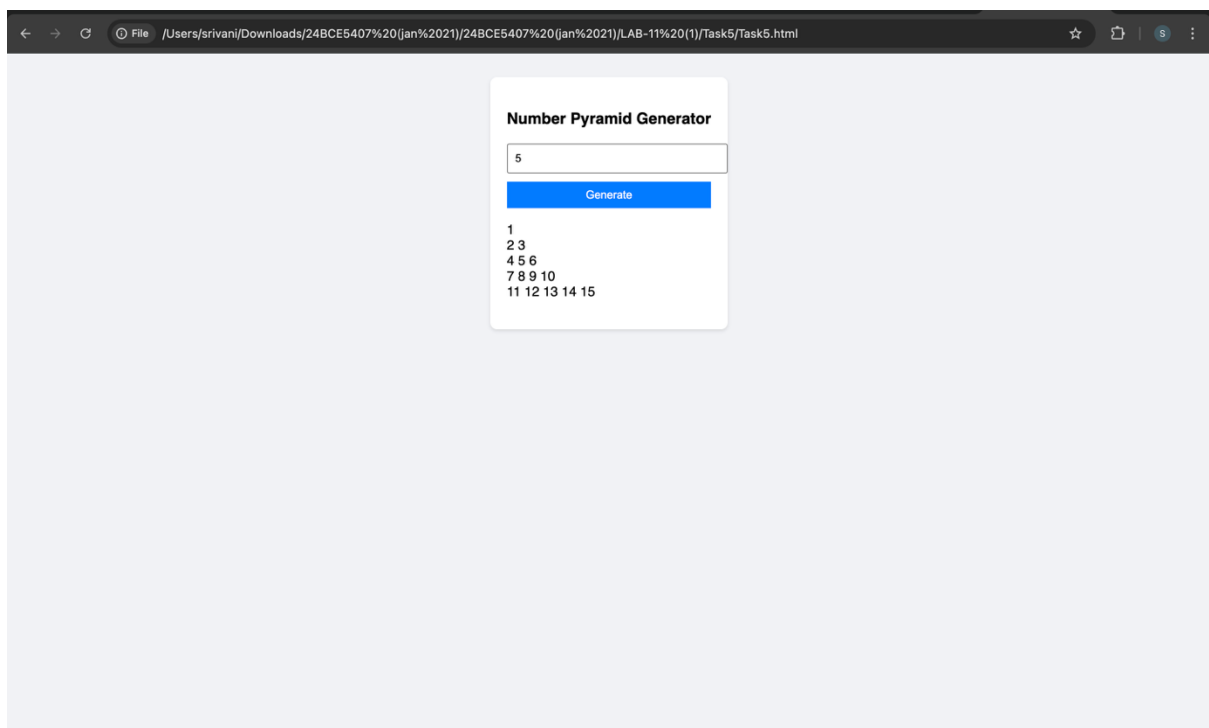
LAB 11 – TASK 5 – CSS:

```
body { font-family: sans-serif; background-color: #f0f2f5; padding: 20px; }  
.card {  
  background: white;  
  padding: 20px;  
  border-radius: 8px;  
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);  
  max-width: 400px;  
}  
input, button {  
  display: block;  
  width: 100%;  
  margin: 10px 0;  
  padding: 8px;  
}  
button {  
  background: #007bff;
```



```
color: white;
border: none;
cursor: pointer;
}
pre{
  background: #eee;
  padding: 10px;
  border-radius: 4px;
}
.container{
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}
```

LAB 11 – TASK 5 – Screenshots:



LAB 11 - TASK 6 – JavaScript:

```
let currentCount = 1;
let maxVehicles = 0;
let totalTax = 0;
let log = "Log:\n";

function startCollection() {
    maxVehicles = parseInt(document.getElementById('totalVehicles').value);
    if (maxVehicles > 0) {
        document.getElementById('setupStep').classList.add('hidden');
        document.getElementById('inputStep').classList.remove('hidden');
        updateLabel();
    }
}

function updateLabel() {
    document.getElementById('currentVehicleLabel').innerText = "Vehicle " +
    currentCount + " of " + maxVehicles;
}

function processNext() {
    const type = document.getElementById('vType').value;
    const isVIP = document.getElementById('isVIP').checked;
    let fee = 0;

    if (!isVIP) {
        if (type === "car") fee = 50;
        else if (type === "truck") fee = 100;
        else if (type === "bike") fee = 30;
    }

    totalTax += fee;
    log += "Vehicle " + currentCount + " (" + type + "): ₹" + fee + (isVIP ? " [VIP]" : "") + "\n";

    if (currentCount < maxVehicles) {
        currentCount++;
        updateLabel();
        document.getElementById('isVIP').checked = false; // Reset checkbox for next
    } else {
        finishCollection();
    }
}
```

```

    }
}

function finishCollection() {
    document.getElementById('inputStep').classList.add('hidden');
    log += "-----\nTotal Collected: ₹" + totalTax;
    document.getElementById('resultDisplay').innerText = log;
}

```

LAB 11 – TASK 6 – CSS:

```

body {

    font-family: sans-serif;
    display: flex;
    justify-content: center;
    padding-top: 50px;
    background-color: gray;
}

.box {
    background-color: rgb(210, 78, 100);
    padding: 25px;
    border-radius: 8px;
    width: 320px;
    text-align: center;
    box-shadow: 0 4px 6px rgba(0,0,0,0.1);
}

.hidden { display: none; }

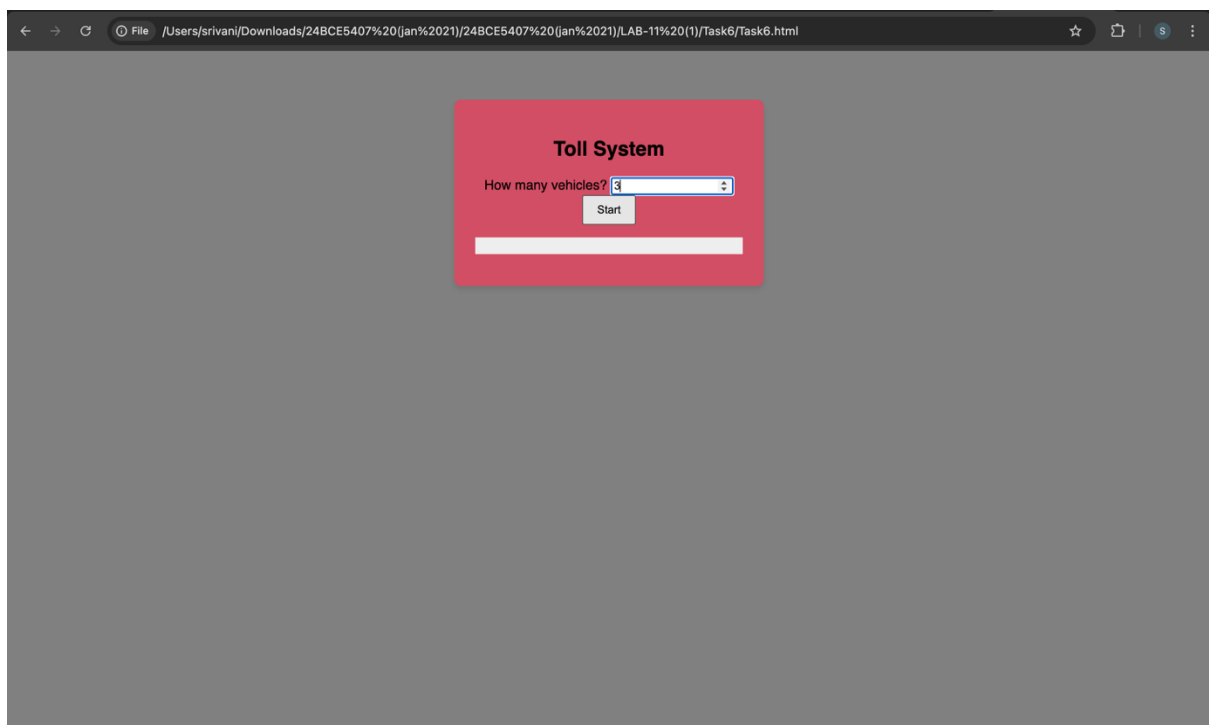
#resultDisplay {
    text-align: left;
    background: #eee;
    padding: 10px;
    margin-top: 15px;
    white-space: pre-wrap;
}

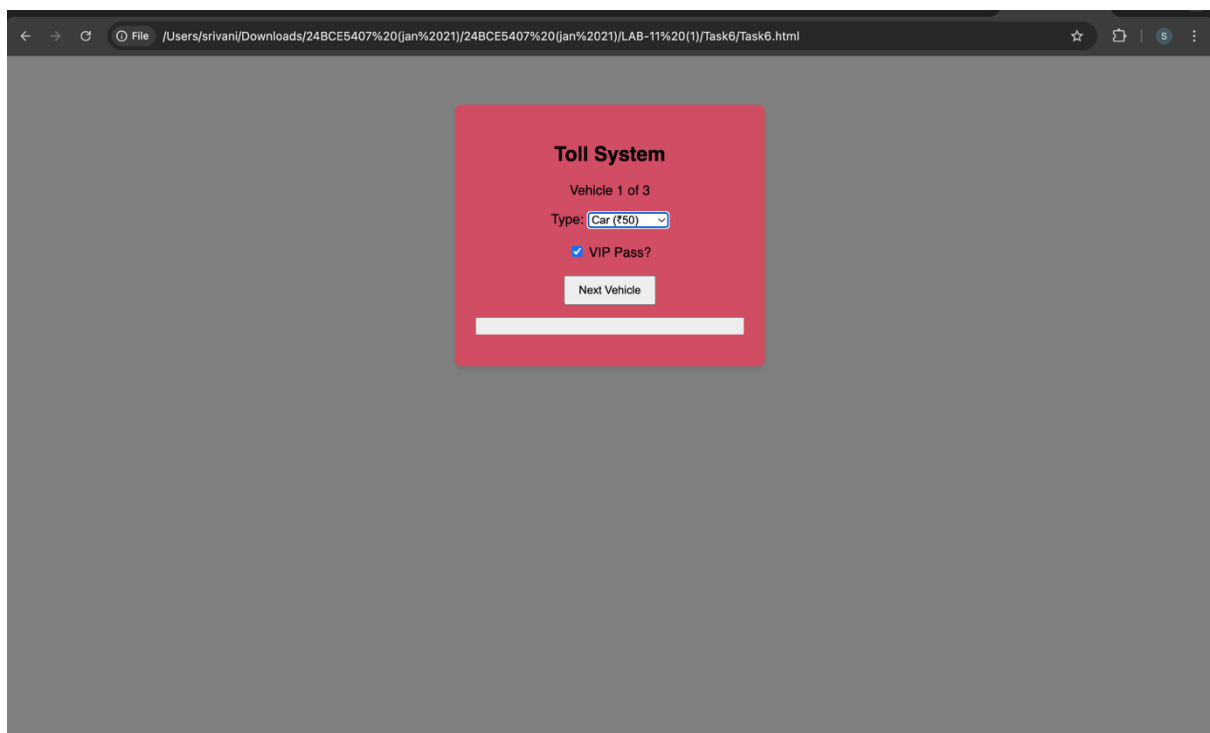
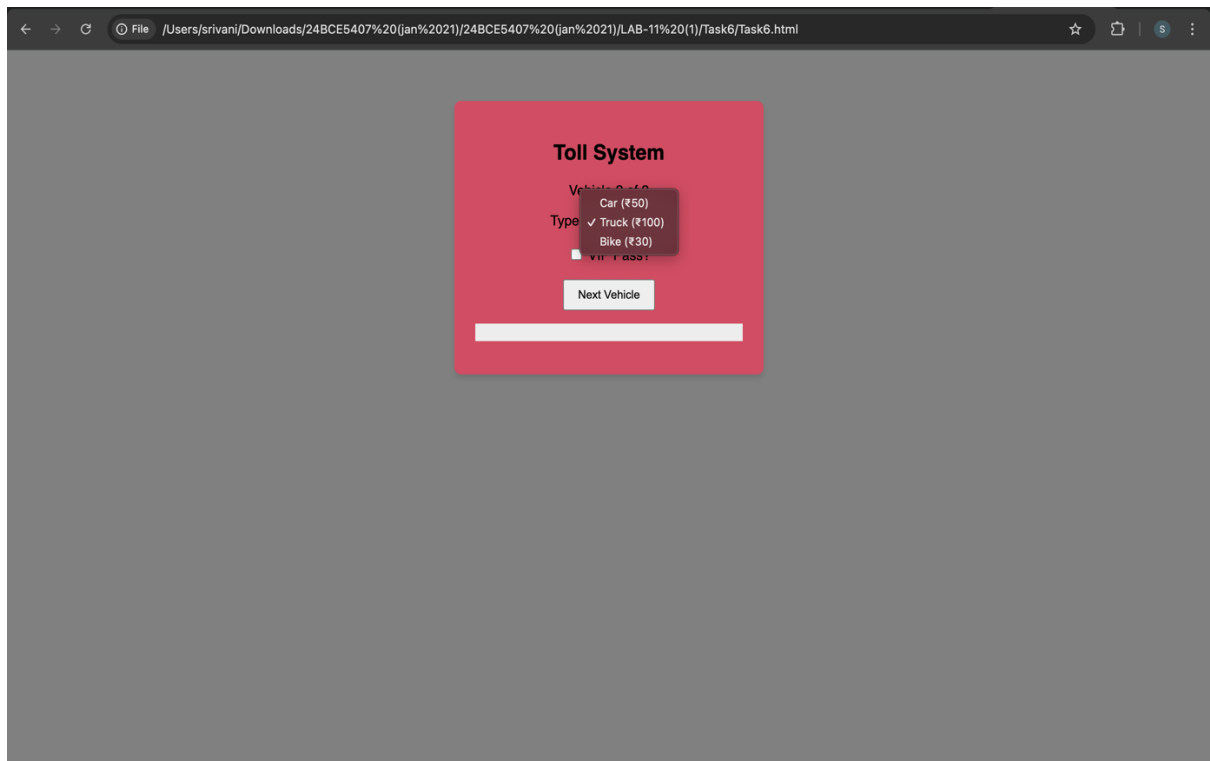
button {
    padding: 8px 16px;
}

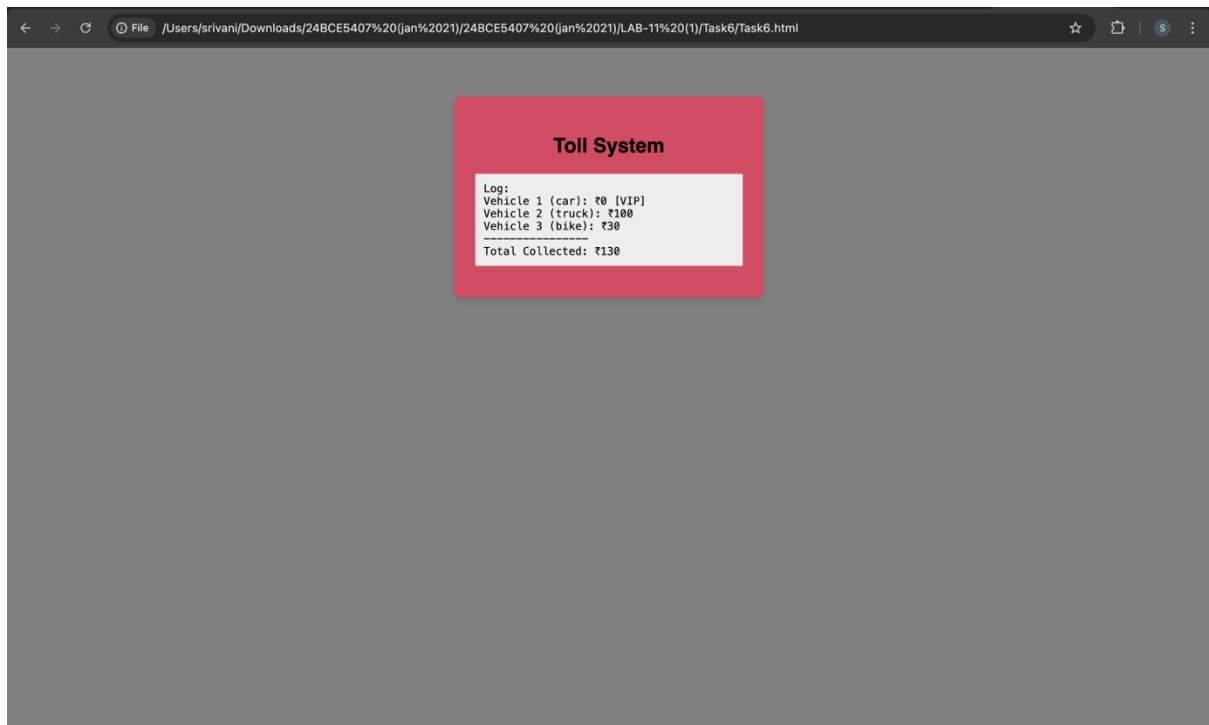
```

```
cursor: pointer;  
}
```

LAB 11 – TASK 6 – Screenshots:







LAB 11 – TASK 7 – HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Smart Hospital Emergency System</title>
  <link rel="stylesheet" href="Task7.css">
</head>
<body>
  <div class="container">
    <h2>Smart Emergency Terminal</h2>

    <div id="setupStep">
      <label>Number of Patients to Process:</label>
      <input type="number" id="patientTotal" value="1" min="1">
      <button class="btn btn-primary" onclick="initializeSystem()">Initialize
System</button>
    </div>
```

```

<div id="inputStep" class="hidden">
  <div class="badge" id="patientCounter">Patient 1 of N</div>

  <div class="form-grid">
    <input type="number" id="age" placeholder="Age">
    <input type="number" id="temp" placeholder="Temp (°C)" step="0.1">
    <input type="number" id="heartRate" placeholder="Heart Rate (BPM)">
    <input type="number" id="oxygen" placeholder="Oxygen Level (%)">

    <select id="injuryType">
      <option value="accident">Accident</option>
      <option value="burn">Burn</option>
      <option value="fracture">Fracture</option>
      <option value="fever">Fever</option>
      <option value="other">Other</option>
    </select>

    <select id="consciousness">
      <option value="conscious">Conscious</option>
      <option value="unconscious">Unconscious</option>
    </select>
  </div>

  <button class="btn btn-success" onclick="processPatient()">Classify &
Next</button>
</div>

<div id="reportStep" class="hidden">
  <pre id="resultDisplay"></pre>
  <div id="emergencyAlert" class="alert hidden">EMERGENCY ALERT: Critical
Threshold Exceeded!</div>
  <button class="btn btn-gray" onclick="location.reload()">New Session</button>
</div>
</div>
<script src="Task7.js"></script>
</body>
</html>

```

LAB 11 – TASK 7 – JavaScript:

```

let currentPatient = 1;
let totalToProcess = 0;

```



```

let criticalCount = 0;
let report = "--- HOSPITAL EMERGENCY REPORT ---\n\n";

function initializeSystem() {
    totalToProcess = parseInt(document.getElementById('patientTotal').value);
    if (totalToProcess > 0) {
        document.getElementById('setupStep').classList.add('hidden');
        document.getElementById('inputStep').classList.remove('hidden');
        updateCounter();
    }
}

function updateCounter() {
    document.getElementById('patientCounter').innerText = "PATIENT " + currentPatient +
    " OF " + totalToProcess;
}

function processPatient() {
    const age = parseInt(document.getElementById('age').value);
    const oxygen = parseInt(document.getElementById('oxygen').value);
    const conscious = document.getElementById('consciousness').value;
    const injury = document.getElementById('injuryType').value;

    let priority = "Low";
    let dept = "General Medicine";
    let wait = "2 Hours";

    if (oxygen < 90 || conscious === "unconscious") {
        priority = "Critical";
        dept = "ICU";
        wait = "Immediate";
        criticalCount++;
    } else if (injury === "accident" || injury === "burn") {
        priority = "High";
        dept = "Emergency Trauma";
        wait = "15 Mins";
    } else if (injury === "fracture") {
        priority = "Moderate";
        dept = "Orthopedics";
        wait = "45 Mins";
    }

    if ((age > 65 || age < 5) && priority !== "Critical") {
        priority = "High (Upgraded)";
        wait = "10 Mins";
    }
}

```

```

report += `Patient ${currentPatient}: [${priority}] -> ${dept} (Wait: ${wait})\n`;

if (currentPatient < totalToProcess) {
    currentPatient++;
    updateCounter();
    clearForm();
} else {
    showFinalReport();
}
}

function clearForm() {
    document.getElementById('age').value = "";
    document.getElementById('oxygen').value = "";
}

function showFinalReport() {
    document.getElementById('inputStep').classList.add('hidden');
    document.getElementById('reportStep').classList.remove('hidden');

    let criticalPercent = (criticalCount / totalToProcess) * 100;
    report += `\nStatistics: ${criticalPercent.toFixed(1)}% Critical cases.`;

    document.getElementById('resultDisplay').innerText = report;

    if (criticalPercent >= 30) {
        document.getElementById('emergencyAlert').classList.remove('hidden');
    }
}

```

LAB 11 – TASK 7 – CSS:

```

body {
    background-color: #f5f5dc;
    font-family: 'Segoe UI', sans-serif;
    display: flex;
    justify-content: center;
    padding-top: 40px;
}

.container {

```

```
background: white;
width: 450px;
padding: 30px;
border-radius: 12px;
box-shadow: 0 10px 30px rgba(0,0,0,0.1);
text-align: center;
}

.form-grid {
display: grid;
grid-template-columns: 1fr 1fr;
gap: 15px;
margin: 20px 0;
}

input, select {
padding: 10px;
border: 2px solid #e0e0e0;
border-radius: 6px;
}

.btn {
width: 100%;
padding: 12px;
border: none;
border-radius: 6px;
font-weight: bold;
cursor: pointer;
margin-top: 10px;
}

.btn-primary { background: #7f8c8d; color: white; }
.btn-success { background: #2d5a27; color: white; }
.btn-gray { background: #bdc3c7; color: #333; }

.badge {
background: #f5f5dc;
color: #7f8c8d;
padding: 5px 15px;
border-radius: 20px;
font-weight: bold;
}

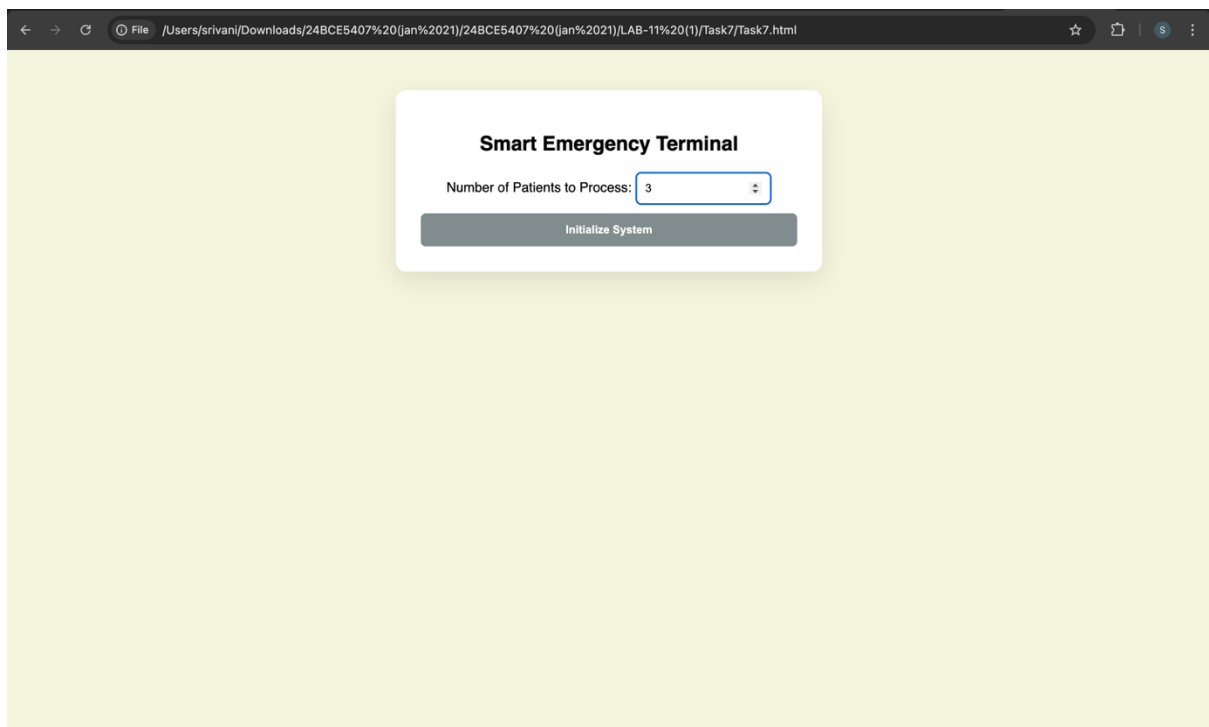
#resultDisplay {
text-align: left;
background: #2c3e50;
color: #ecf0f1;
}
```

```
padding: 15px;
border-radius: 6px;
white-space: pre-wrap;
}

.alert{
background: #e74c3c;
color: white;
padding: 10px;
margin-top: 10px;
border-radius: 6px;
animation: blink 1s infinite;
}

.hidden{
display: none;
}
```

LAB 11 – TASK 7 – Screenshots:



File

/Users/srivani/Downloads/24BCE5407%20(jan%2021)/24BCE5407%20(jan%2021)/LAB-11%20(1)/Task7/Task7.html

☆

S

Smart Emergency Terminal

PATIENT 1 OF 3

33

34

90

112

Other

Unconscious

Classify & Next

File

/Users/srivani/Downloads/24BCE5407%20(jan%2021)/24BCE5407%20(jan%2021)/LAB-11%20(1)/Task7/Task7.html

☆

S

Smart Emergency Terminal

HOSPITAL EMERGENCY REPORT

Patient 1: [Low] -> General Medicine (Wait: 2 Hours)
Patient 2: [High] -> Emergency Trauma (Wait: 15 Mins)
Patient 3: [Critical] -> ICU (Wait: Immediate)
Statistics: 33.3% Critical cases.

EMERGENCY ALERT: Critical Threshold Exceeded!

New Session

THE END