ALGORITHM:

STEP 1: Start the Program

STEP 2: Create HTML iles: index.html, login.html, applyLeave.html, viewLeave.html

STEP 3: Set up basic structure in each HTML ile (<html>, <head>, <body>)

STEP 4: Create forms for login, applying leave, and viewing leave balance

STEP 5: Set up backend using Node.js and Express

STEP 6: Create REST API routes for user authentication, applying leave, and viewing leave balance

STEP 7: Connect the application to a database (e.g., MySQL/MongoDB) to store user and leave data

STEP 8: Implement JWT for authentication and session management

STEP 9: Create styles.css to style forms, leave balance table, and buttons

STEP 10: Test the low for applying leave and viewing available leave days

STEP 11: End the Program

PROGRAM:

BACKEND (Node.js & Express.js):

server.js:

const express = require('express');

const mongoose = require('mongoose');

const bodyParser = require('body-parser');

const app = express(); // Middleware

app.use(bodyParser.json()); // Database connection mongoose.connect('mongodb://localhost/leave\_management', { useNewUrlParser: true,

useUni iedTopology: true })

.then(() => console.log('Connected to MongoDB...'))

.catch(err => console.error('Could not connect to MongoDB...', err));

// Leave Schema

const leaveSchema = new mongoose.Schema({

user: String, type: String, days: Number, status: String // pending, approved, rejected

});

const Leave = mongoose.model('Leave', leaveSchema);

// Routes app.post('/apply-leave', async (req, res) => { const { user, type, days } = req.body;

const newLeave = new Leave({

user, type, days,

status: 'pending'

});

await newLeave.save(); res.send("Leave application submitted!");

});

app.get('/view-leave/:user', async (req, res) => { const leaves = await Leave. ind({ user: req.params.user }); res.send(leaves);

});

// Start Server

app.listen(3000, () => console.log('Server started on port 3000'));

FRONTEND (HTML/CSS): index.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Leave Management System</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="container">

<h1>Leave Management System</h1>

<div class="apply-leave">

<h2>Apply for Leave</h2>

<form id="leaveForm">

<label for="user">Employee Name:</label>

<input type="text" id="user" name="user" required>

<label for="type">Leave Type:</label>

<select id="type" name="type">

<option value="casual">Casual Leave</option>

<option value="medical">Medical Leave</option>

</select>

<label for="days">Number of Days:</label>

<input type="number" id="days" name="days" required>

<button type="submit">Apply</button>

</form>

</div>

<div class="view-leaves">

<h2>Your Leaves</h2>

<button onclick="viewLeaves()">View Leaves</button>

<div id="leaveDetails"></div>

</div>

</div>

<script src="app.js"></script>

</body>

</html>

CSS (style.css):

body { font-family: Arial, sans-serif; background-color: #f4f4f4;

padding: 20px;

}

.container { width: 50%; margin: auto; background: white; padding: 20px; border-radius: 10px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h1, h2 { text-align: center;

color: #333;

}

form { display: lex;

lex-direction: column;

} input, select { padding: 10px; margin: 5px 0; border-radius: 5px;

border: 1px solid #ddd;

}

button { padding: 10px; background-color: #28a745; color: white; border: none; border-radius: 5px;

cursor: pointer;

}

button:hover { background-color: #218838;

}

FRONTEND JAVASCRIPT (Client-Side Logic): app.js:

// Apply Leave

document.getElementById('leaveForm').addEventListener('submit', async function(e) { e.preventDefault(); const user = document.getElementById('user').value; const type = document.getElementById('type').value; const days = document.getElementById('days').value; const response = await fetch('/apply-leave', { method: 'POST', headers: { 'Content-Type': 'application/json' },

body: JSON.stringify({ user, type, days })

});

const result = await response.text();

alert(result);

});

// View Leaves async function viewLeaves() { const user = prompt('Enter your name:'); const response = await fetch(`/view-leave/${user}`); const leaves = await response.json(); let output = '<h3>Leave Details</h3><ul>'; leaves.forEach(leave => { output += `<li>${leave.type}: ${leave.days} days (${leave.status})</li>`;

});

output += '</ul>'; document.getElementById('leaveDetails').innerHTML = output;

}