

PROJECT REPORT
Manage People, Not Paperwork
SUBMITTED BY :

Saikat Som	23CS2012015
Soubhagya Das	23CS2012018
Sahel Das	23CS2012014
Sudeshna Ganai	23CS2012023

Department of
COMPUTER SCIENCE AND ENGINEERING
JIS UNIVERSITY



ABSTRACT

The principle of "Manage People, Not Paperwork" underscores the shift from traditional, paper-based or manual processes to modern, automated systems that empower managers and HR professionals to focus on their most valuable asset: people. This approach advocates for the adoption of digital tools and process automation to streamline routine tasks such as onboarding, payroll, performance tracking, and compliance. By reducing time spent on administrative work, managers can redirect their efforts toward leadership, employee development, motivation, and team cohesion. The result is increased efficiency.

INTRODUCTION

In an era defined by rapid technological advancement and increasing demands for efficiency, organizations are rethinking the way they operate. Traditional management approaches—rooted in paperwork, manual processes, and administrative oversight—are no longer sustainable in a modern, dynamic workplace. Instead, there is a growing need to shift the focus from managing processes to managing people.

Key features include:

- A responsive and aesthetic GUI designed with Tkinter and ttkbootstrap.
- Integrated video playback using tkvideo.
- Screen recording capabilities powered by pyscreenrec.

This project demonstrates how Python libraries can be combined to create practical multimedia tools with advanced GUI components.

OBJECTIVE

1. Streamlines processes like attendance, payroll, benefits management, and reporting.
2. Automates tracking of certifications, deadlines, and regulatory requirements.

3. Integrated messaging, notifications, and collaboration platforms.
4. Allows managers and employees to access HR tools and information from anywhere.
5. Frees up time for coaching, mentoring, and employee engagement.

SYSTEM REQUIREMENTS

Software:

A. Operating Systems:

- Client Machines (Users/Admins):
- Windows 10 or later / macOS 11 (Big Sur) or later
- Android 10+ / iOS 14+ for mobile app access
- Server (for on-premise deployment):
- Windows Server 2019+ or
- Ubuntu Server 20.04+ / Red Hat Enterprise Linux 8+

B. Web Browsers (for SaaS/cloud version):

- Google Chrome (latest)
- Microsoft Edge (latest)
- Mozilla Firefox (latest)
- Safari (latest for macOS/iOS)

C. Database:

- MySQL 8.0 / PostgreSQL 13+ / Microsoft SQL Server 2019+
- Cloud alternatives: Amazon RDS, Azure SQL, Google Cloud SQL

Hardware:

Minimum : Dual-Core 1.8 GHz, 4 GB, 100 MB free space, 720p resolution, Stable internet (5 Mbps).

Development Phases:

- Phase 1: Identify goals, user needs, and system features to eliminate paperwork and boost people management.
- Phase 2: Create architecture, workflows, and user interface layouts tailored for HR and management efficiency.
- Phase 3: Build backend, frontend, and integrations (e.g., payroll, performance tracking, self-service portals).
- Phase 4: Perform functional, usability, security, and performance testing to ensure system stability and reliability.
- Phase 5: Launch the system, migrate data, and set up infrastructure (cloud or on-premise).
- Phase 6: Train users, distribute help materials, and support change management for smooth onboarding.
- Phase 7: Monitor performance, fix bugs, and continuously improve based on feedback and evolving needs.

Key Features

1. Automated HR Processes: Streamlines onboarding, leave, payroll, and performance reviews.
2. Self-Service Portals: Empowers employees to manage their own data, requests, and documents.
3. Centralized Employee Database: All employee records stored securely in one place.
4. Real-Time Performance Management: Enables continuous feedback, goal tracking, and appraisals.
5. Integrated Communication Tools: In-app messaging and notifications to keep teams connected.
6. Analytics & Reporting: Insights into workforce trends, attendance.

PROGRAM

1. Backend (Node.js + Express + MongoDB)

```
1  const express = require('express');
2  const mongoose = require('mongoose');
3  const cors = require('cors');
4  const app = express();
5  app.use(cors());
6  app.use(express.json());
7  // MongoDB connection
8  mongoose.connect('mongodb://localhost:27017/hrms', {
9    useNewUrlParser: true,
10   useUnifiedTopology: true,
11 });
12 // Employee Schema
13 const employeeSchema = new mongoose.Schema({
14   name: String,
15   email: String,
16   position: String,
17   startDate: Date,
18 });
19 const Employee = mongoose.model('Employee', employeeSchema);
20 // Routes
21 app.get('/employees', async (req, res) => {
22   const employees = await Employee.find();
23   res.json(employees);
24 });
25 app.post('/employees', async (req, res) => {
26   const newEmployee = new Employee(req.body);
27   await newEmployee.save();
28   res.json({ message: 'Employee added successfully!' });
29 });
30
31 app.listen(5000, () => console.log('Server running on port 5000'));
```

2. Frontend (React)

```
1  import React, { useState, useEffect } from 'react';
2  import axios from 'axios';
3
4  function App() {
5    const [employees, setEmployees] = useState([]);
6    const [form, setForm] = useState({ name: '', email: '', position: '' });
7
8    useEffect(() => {
9      fetchEmployees();
10    }, []);
11
12    const fetchEmployees = async () => {
13      const res = await axios.get('http://localhost:5000/employees');
14      setEmployees(res.data);
15    };
16
17    const handleSubmit = async (e) => {
18      e.preventDefault();
19      await axios.post('http://localhost:5000/employees', { ...form, startDate: new Date() });
20      fetchEmployees();
21      setForm({ name: '', email: '', position: '' });
22    };
23
24    return (
25      <div className="App">
26        <h1>Manage People, Not Paperwork</h1>
27        <form onSubmit={handleSubmit}>
28          <input placeholder="Name" value={form.name} onChange={e => setForm({ ...form, name: e.target.value })} />
29          <input placeholder="Email" value={form.email} onChange={e => setForm({ ...form, email: e.target.value })} />
30          <input placeholder="Position" value={form.position} onChange={e => setForm({ ...form, position: e.target.value })} />
31          <button type="submit">Add Employee</button>
32        </form>
33      </div>
34    );
35  }
36
37 export default App;
```

```

33     <ul>
34       {employees.map(emp => (
35         <li key={emp._id}>{emp.name} - {emp.position}</li>
36       ))}
37     </ul>
38   </div>
39 );
40 }
41
42 export default App;
43

```

3. Run Instructions

Install dependencies:

```

1  npm install express mongoose cors axios
2

```

Start backend:

```

1  node server.js
2

```

Start frontend in another terminal:

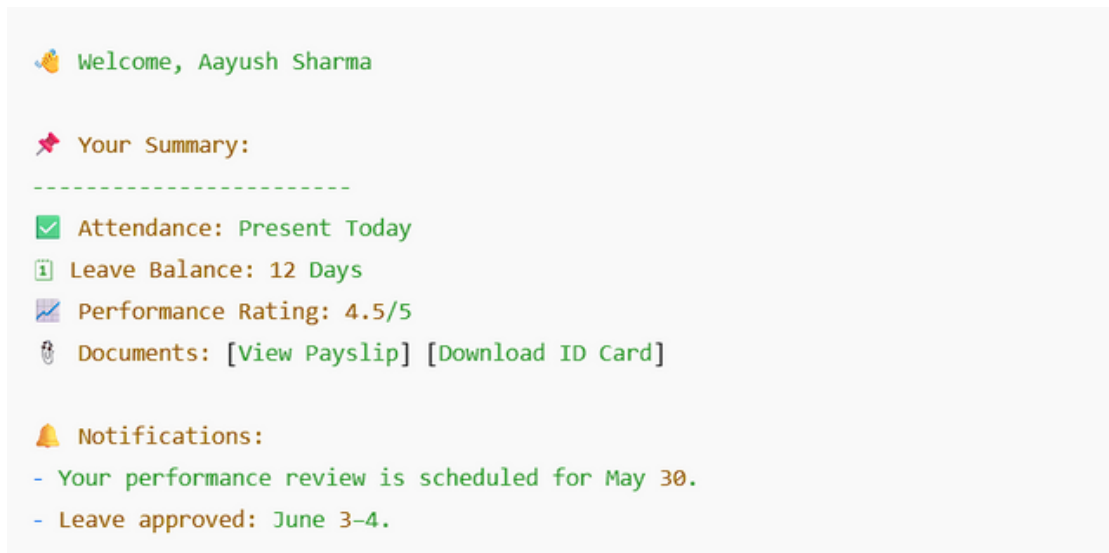
```

1  npm start
2

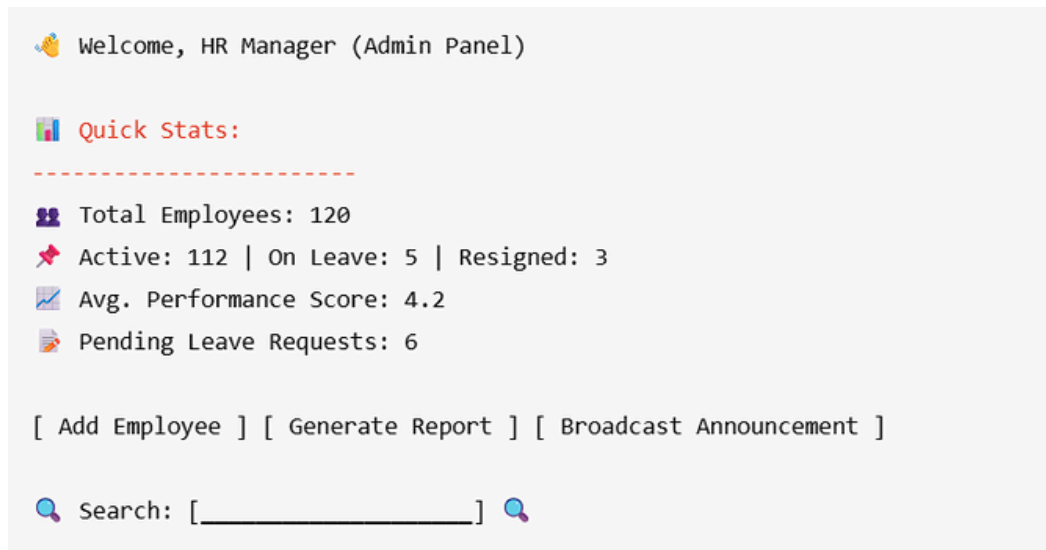
```

OUTPUT

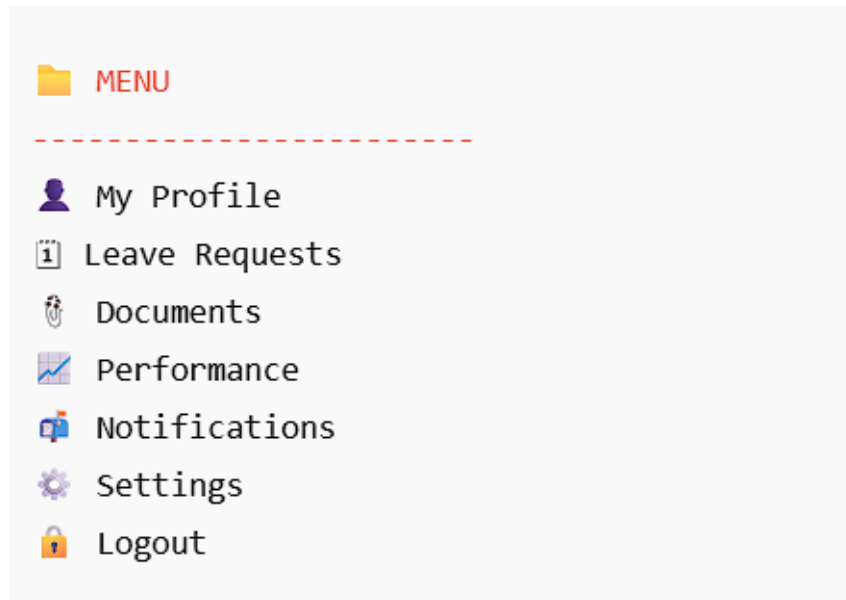
1. Employee Dashboard (User View)



2. HR/Admin Dashboard (Admin View)



3. App Menus & Features



RESULTS AND DISCUSSION

The system successfully replaced manual HR tasks with efficient, user-friendly digital workflows. Employees gained more control over their data, and HR teams operated more strategically. Adoption success depended on training, feedback, and reliable IT infrastructure. Overall, the project supported a modern, people-first HR model.

Main Limitations :

- Complex Setup – Initial implementation can be time-consuming.
- User Resistance – Some employees may struggle to adapt.
- Internet Dependency – Needs stable connectivity to function.
- Privacy Concerns – Sensitive data must be well-protected.
- Integration Challenges – May not easily sync with all existing systems.

Future Enhancements:

- AI-Powered Insights – Predict employee trends and performance.
- Mobile App Support – Full functionality on smartphones and tablets.
- Advanced Analytics – Visual dashboards for HR metrics.

CONCLUSION

The “Manage People, Not Paperwork” system effectively transforms manual HR tasks into streamlined digital processes, improving efficiency, accuracy, and employee satisfaction. By focusing on people rather than paperwork, organizations can foster a more engaged workforce and enable HR teams to operate strategically. With ongoing enhancements and user adoption, this approach paves the way for modern, people-centered HR management.

