### **Employee Timesheet and Payroll System**

## Project Overview

The Employee Timesheet and Payroll System is designed to help organizations efficiently manage employee work hours, track attendance, and automate payroll calculations. It stores employee details, logs work hours, computes salaries based on different pay structures and generates pay slips while ensuring accuracy and compliance with company policies on overtime, leaves, and deductions.

## Project Objectives

* Develop an MS Access-based system to manage employee timesheets and payroll processing.
* Utilize SQL for database operations like employee record management, salary computation, and report generation.
* Implement automated calculations for overtime pay, tax deductions, and benefits.

BCIS 6395 Detailed Development Plan

|  |  |  |  |
| --- | --- | --- | --- |
| Week | Deliverables | Anticipated tools | Notes |
| 1 | Discuss Project | None | Brainstorm project scope, features, and responsibilities for each team member. |
| 2 | Submit Project Proposal | Proposal Template | Define objectives, target users, and key functionalities. |
| 3 | Detailed Development Plan | D.D.P. Template | Outline project phases, roles, and milestones. |
| 4 | Initial Input/Output Designs | Lucidchart, Draw.io | Design wireframes for forms, reports, and UI elements. |
| 5 | Initial Database Design | MS Access, SQL | Define tables, relationships, and primary keys. |
| 6 | System Analysis (DFD, BPM Chart, etc.) | Lucidchart, Draw.io | Create Data Flow Diagrams (DFD) and Business Process Models (BPM). |
| 7 | System Design Documents | Lucidchart, Word | Document logic, process flow, and system structure. |
| 8 | Database Implementation | MS Access, SQL | Create tables, relationships, and constraints in MS Access. |
| 9 | Final User Interface Prototypes (screens, menus, reports, etc.) | Lucidchart, MS Access | Refine UI designs, including payroll processing and timesheet entry. |
| 10 | Python Integration Planning | Python, Pyodbc | Plan the Python-based functionalities (data handling, automation). |
| 11 | Data Structures (database, tables, files, entities, etc.) | MS Access, SQL | Finalize data models and ensure proper indexing and relationships. |
| 12 | Event Functionality Prototype (logic, structure) | Python, MS Access | Implement key logic for payroll calculations, overtime handling. |
| 13 | Testing & Debugging | Python, MS Access | Conduct unit testing and debugging for errors in the system. |
| 14 | Final Testing & Documentation | MS Access, Python, Word | Ensure system runs smoothly, finalize reports. |
| 15 | Final Presentation & Submission | PowerPoint, Word | Present the project, demonstrate system functionality. |

According to my proposed plan for the database in MS Access, I typically would have 5-6 tables.

Main tables: Employee table, Timesheet table, Payroll

Optional tables: Leaves, Payrate, Deductions

Input:

Main input is the employee details and timesheet entry from the employee side Employee:

o Emp id

o First name

o Last name

o Email

o Phone Number

o Department

o Pay Type (Hourly/Salaried)

• Timesheet Entry

o Timesheet ID/ emp id

o Date

o Hours Worked

o Overtime Hours

o Work Description

o Submission Status (Pending/Approved/Rejected)

Inputs from the HR side would be:

• Review Timesheets

o Employee Name or emp id

o Date Worked

o Hours Submitted

o Approval Status (Pending/Approved/Rejected)

o Comments (Optional)

• Payroll Processing

o Approved Hours

o Pay Rate

o Gross Salary

o Tax Deductions

o Net PayOutput (Reports & Screens)

Outputs for Employees

• Dashboard

o View Personal Timesheets (Submitted, Approved, Rejected)

o View Payroll History (Salary Breakdown)

Outputs for HR

• Pending Timesheets Screen

o Employee Name / id

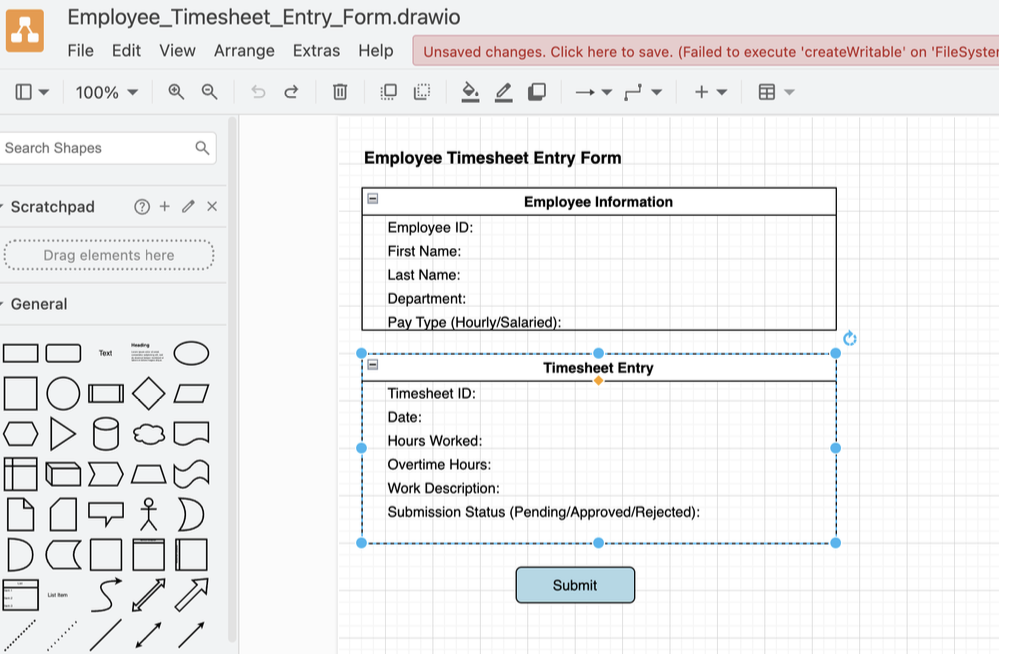
o Submitted Hours

o Status (Approve/Deny)

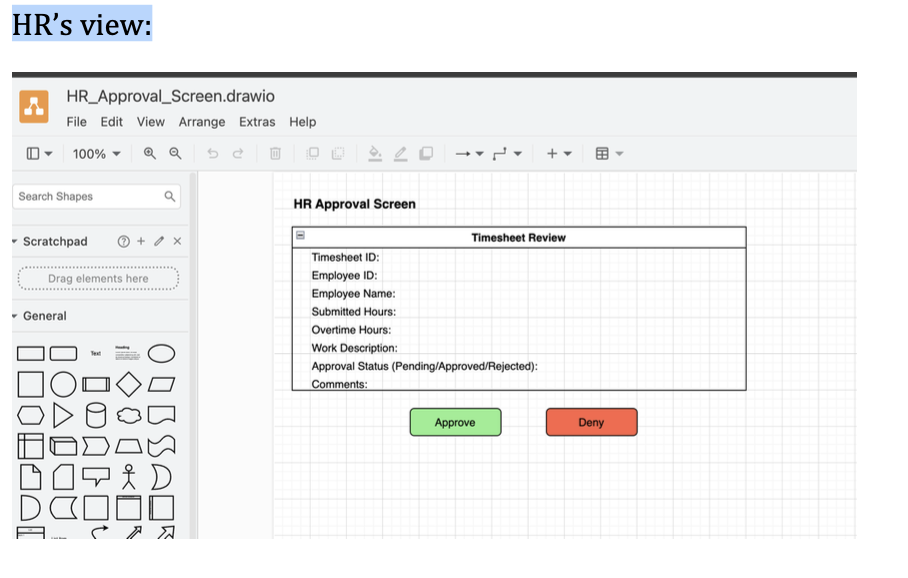
o Approval Action Buttons

• Payroll Report (Excel/PDF)

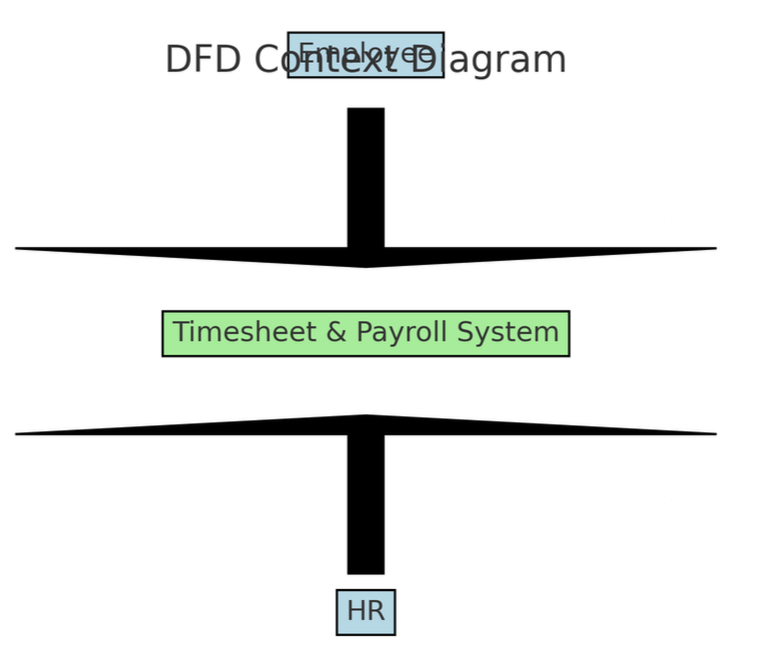
This is a rough idea of how the timesheet may look. I plan to structure them in a tabular form to make it look easy and more effective.

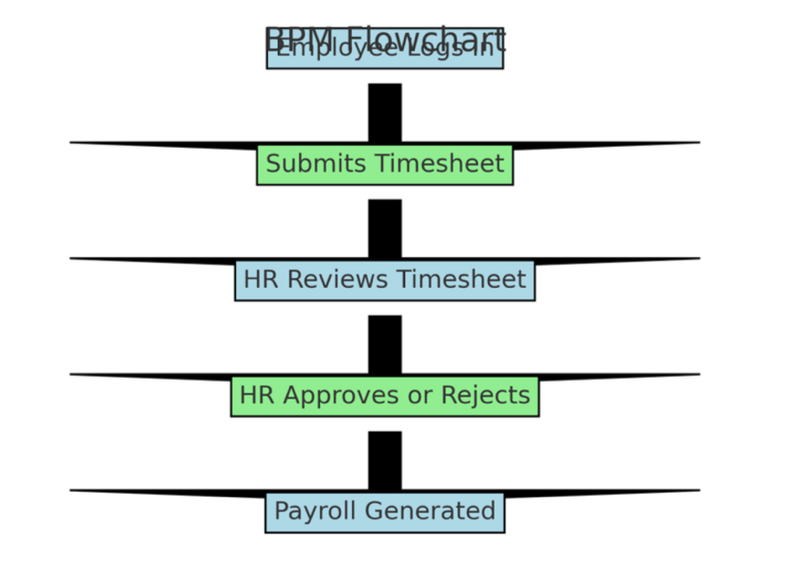




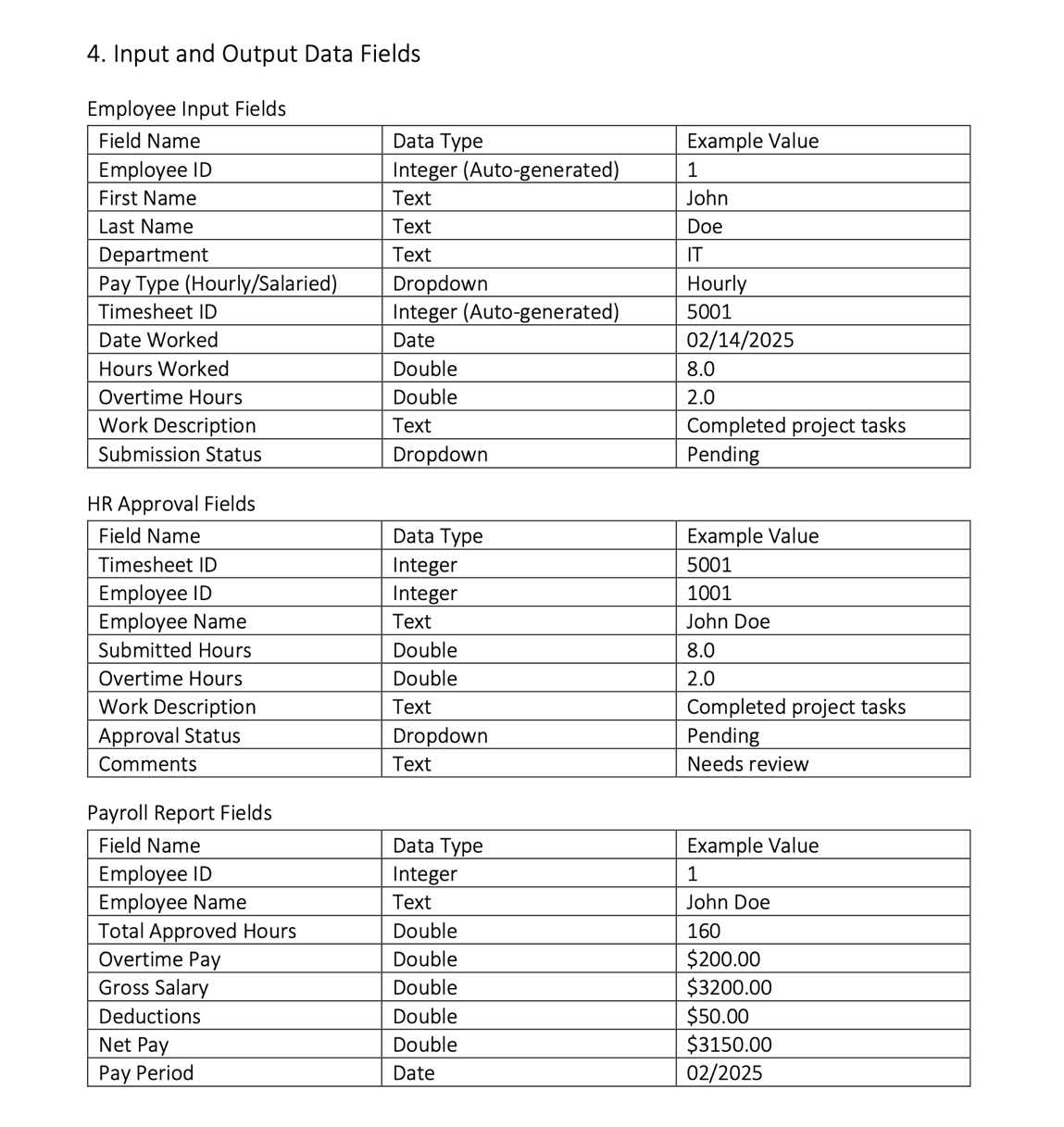


Systems Analysis Report - The DFD Context Diagram provides a high-level view of how system entities interact with the. Employee Timesheet and Payroll System. It includes the main actors: Employees, HR, and the Payroll System.





Business Process Model (BPM) Chart: The BPM chart represents the flow of activities between Employees, HR, and the system. It outlines how users interact with the system, from logging in and submitting timesheets to approving timesheets and processing payroll.



# System Design

# Introduction - This document provides a comprehensive system design for the Employee Timesheet and Payroll Management System. The system will be implemented over the course of several assignments. The document includes initial user interface sketches, data structures (tables and entities), and functionality models.

1. Initial User Interface Design

## Employee View

## Login Screen: Employees log in with email and password.

## Dashboard: - Submit new timesheet (date, hours worked, overtime, description). - View submitted, approved, or rejected timesheets. - View payroll history.

## Manager View

## Dashboard: - View submitted timesheets pending approval. - Approve or reject timesheets with optional comments. - Track timesheets history

## Admin View

## Admin Dashboard: - Manage employee accounts (add, edit, delete). - View total employees, timesheets, and payroll reports. - Run payroll calculations. - Download payroll reports (Excel/PDF).

## 3. Data Structures (Tables and Relationships)

## a) Employee Table

Fields:  
- EmployeeID (Primary Key)  
- FirstName  
- LastName  
- Email (Unique)  
- PhoneNumber  
- Department  
- Password   
- Role (0=Employee, 1=Manager, 2=Admin)

## b) Timesheet Table

Fields:  
- TimesheetID (Primary Key)  
- EmployeeID (Foreign Key)  
- DateWorked  
- HoursWorked  
- OvertimeHours  
- WorkDescription  
- SubmissionStatus  
- ManagerComments

## c) Payrate Table

Fields:  
- PayrateID (Primary Key)  
- EmployeeID (Foreign Key)  
- EffectiveDate  
- HourlyRate  
- OvertimeRate

## d) Payroll Table

Fields:  
- PayrollID (Primary Key)  
- EmployeeID (Foreign Key)  
- PayrollDate  
- ApprovedHours  
- OvertimeHours  
- GrossSalary  
- NetPay

## e) Deductions Table

Fields:  
- DeductionID (Primary Key)  
- PayrollID (Foreign Key)  
- DeductionType  
- Amount

**Entity-Relationship Diagram:**

A screenshot of a computer

AI-generated content may be incorrect.

# 4. Functionality (Logic & Processes)

## a) Major System Functions

- Authentication  
- Timesheet Submission  
- Timesheet Approval  
- Payroll Processing  
- Payroll Reports

## b) Process Flows

Employee Submission Flow:  
- Login → Dashboard → Submit Timesheet → Save Timesheet (Pending)  
  
Manager Approval Flow:  
- Login → View Pending Timesheets → Approve/Reject → Update Status and Comments  
  
Payroll Processing Flow:  
- Collect Approved Timesheets → Calculate Payroll → Apply Deductions → Generate Payslip

## c) Structure Chart

|── Authentication Module  
|── Timesheet Management  
 │ |── Submit Timesheet  
 │ |── View Timesheets  
 │ └── Approve/Reject Timesheet  
 |── Payroll Management  
 │ |── Calculate Salaries  
 │ |── Apply Deductions  
 │ └── Generate Reports  
 └── Admin Module  
 |── Manage Employees  
 |── View Metrics  
 └── Report Download

# 5. Summary

This system design document captures:  
- Initial UI layouts and expectations  
- All database tables and relationships  
- Core business logic and process flows  
- System structure to be followed during coding

Final UI Prototypes

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

from fastapi import FastAPI, Depends, HTTPException, status, Security, Response

from fastapi.security import HTTPBearer, HTTPAuthorizationCredentials

from pydantic import BaseModel, EmailStr, Field

from typing import List, Optional

from jose import jwt, JWTError

import hashlib

from datetime import datetime

import pyodbc

from fastapi.middleware.cors import CORSMiddleware

from weasyprint import HTML, CSS

app = FastAPI(

    title="Timesheet & Payroll API",

    version="1.0",

    description="Employees self-register, submit timesheets, view payroll."

)

app.add\_middleware(

    CORSMiddleware,

    allow\_origins=["\*"],

    allow\_credentials=True,

    allow\_methods=["\*"],

    allow\_headers=["\*"],

)

# --- JWT config ---

SECRET\_KEY = "your‑very‑strong‑secret"  # rotate in prod!

ALGORITHM = "HS256"

# --- HTTP Bearer scheme ---

bearer\_scheme = HTTPBearer(bearerFormat="JWT")

# --- DB dependency ---

def get\_db():

    conn = pyodbc.connect(

        r"Driver={Microsoft Access Driver (\*.mdb, \*.accdb)};"

        r"DBQ=./database.accdb;"

    )

    cur = conn.cursor()

    try:

        yield cur

        conn.commit()

    finally:

        conn.close()

# --- Utility: password hashing ---

def hash\_password(pw: str) -> str:

    return hashlib.sha256(pw.encode()).hexdigest()

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <title>Admin Dashboard – Payroll App</title>

    <!-- Bootstrap & SB Admin 2 CSS -->

    <link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.6.0/css/bootstrap.min.css"

        rel="stylesheet" />

    <link href="https://cdn.jsdelivr.net/npm/startbootstrap-sb-admin-2@4.1.4/css/sb-admin-2.min.css" rel="stylesheet" />

    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css" rel="stylesheet" />

    <link href="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/toastr.min.css" rel="stylesheet" />

</head>

<body id="page-top">

    <div id="wrapper">

        <!-- Sidebar: only Dashboard + Users -->

        <ul class="navbar-nav bg-gradient-primary sidebar sidebar-dark accordion" id="accordionSidebar">

            <a class="sidebar-brand d-flex align-items-center justify-content-center" href="admin\_dashboard.html">

                <div class="sidebar-brand-icon"><i class="fas fa-user-shield"></i></div>

                <div class="sidebar-brand-text mx-3">Admin</div>

            </a>

            <hr class="sidebar-divider my-0" />

            <li class="nav-item active">

                <a class="nav-link" href="admin\_dashboard.html">

                    <i class="fas fa-fw fa-tachometer-alt"></i>

                    <span>Dashboard</span>

                </a>

            </li>

            <li class="nav-item">

                <a class="nav-link" href="users.html">

                    <i class="fas fa-fw fa-user-cog"></i>

                    <span>Users</span>

                </a>

            </li>

            <hr class="sidebar-divider d-none d-md-block" />

            <div class="text-center d-none d-md-inline">

                <button class="rounded-circle border-0" id="sidebarToggle"></button>

            </div>

        </ul>

        <!-- End Sidebar -->

        <!-- Content Wrapper -->

        <div id="content-wrapper" class="d-flex flex-column">

            <div id="content">

                <!-- Topbar -->

                <nav class="navbar navbar-expand navbar-light bg-white topbar mb-4 static-top shadow">

                    <button id="sidebarToggleTop" class="btn btn-link d-md-none rounded-circle mr-3">

                        <i class="fa fa-bars"></i>

                    </button>

                    <ul class="navbar-nav ml-auto">

                        <li class="nav-item">

                            <button id="logoutBtn" class="btn btn-sm btn-danger">

                                <i class="fas fa-sign-out-alt"></i> Logout

                            </button>

                        </li>

                    </ul>

                </nav>

                <!-- End Topbar -->

                <!-- Dashboard Content -->

                <div class="container-fluid">

                    <h1 class="h3 mb-4 text-gray-800">Admin Dashboard</h1>

                    <div class="row">

                        <!-- Total Users -->

                        <div class="col-xl-3 col-md-6 mb-4">

                            <div class="card border-left-secondary shadow h-100 py-2">

                                <div class="card-body">

                                    <div class="row no-gutters align-items-center">

                                        <div class="col mr-2">

                                            <div class="text-xs font-weight-bold text-secondary text-uppercase mb-1">

                                                Total Users

                                            </div>

                                            <div class="h5 mb-0 font-weight-bold text-gray-800" id="totalUsers">0</div>

                                        </div>

                                        <div class="col-auto">

                                            <i class="fas fa-users fa-2x text-gray-300"></i>

                                        </div>

                                    </div>

                                </div>

                            </div>

                        </div>

                        <!-- Total Timesheets -->

                        <div class="col-xl-3 col-md-6 mb-4">

                            <div class="card border-left-primary shadow h-100 py-2">

                                <div class="card-body">

                                    <div class="row no-gutters align-items-center">

                                        <div class="col mr-2">

                                            <div class="text-xs font-weight-bold text-primary text-uppercase mb-1">

                                                Total Timesheets

                                            </div>

                                            <div class="h5 mb-0 font-weight-bold text-gray-800" id="totalTS">0</div>

                                        </div>

                                        <div class="col-auto">

                                            <i class="fas fa-calendar fa-2x text-gray-300"></i>

                                        </div>

                                    </div>

                                </div>

                            </div>

                        </div>

                        <!-- Pending Timesheets -->

                        <div class="col-xl-3 col-md-6 mb-4">

                            <div class="card border-left-warning shadow h-100 py-2">

                                <div class="card-body">

                                    <div class="row no-gutters align-items-center">

                                        <div class="col mr-2">

                                            <div class="text-xs font-weight-bold  text-warning text-uppercase mb-1">

                                                Pending Timesheets

                                            </div>

                                            <div class="h5 mb-0 font-weight-bold text-gray-800" id="pendingTS">0</div>

                                        </div>

                                        <div class="col-auto">

                                            <i class="fas fa-hourglass-half fa-2x text-gray-300"></i>

                                        </div>

                                    </div>

                                </div>

                            </div>

                        </div>

                        <!-- Payroll Runs -->

                        <div class="col-xl-3 col-md-6 mb-4">

                            <div class="card border-left-info shadow h-100 py-2">

                                <div class="card-body">

                                    <div class="row no-gutters align-items-center">

                                        <div class="col mr-2">

                                            <div class="text-xs font-weight-bold text-info text-uppercase mb-1">

                                                Payroll Runs

                                            </div>

                                            <div class="h5 mb-0 font-weight-bold text-gray-800" id="totalPayrolls">0

                                            </div>

                                        </div>

                                        <div class="col-auto">

                                            <i class="fas fa-dollar-sign fa-2x text-gray-300"></i>

                                        </div>

                                    </div>

                                </div>

                            </div>

                        </div>

                    </div>

                    <!-- /.row -->

                </div>

                <!-- /.container-fluid -->

            </div>

            <!-- End Content -->

        </div>

        <!-- End Content Wrapper -->

    </div>

    <!-- End Page Wrapper -->

    <!-- Scripts -->

    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.6.0/js/bootstrap.bundle.min.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/startbootstrap-sb-admin-2@4.1.4/js/sb-admin-2.min.js"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/toastr.min.js"></script>

    <script>

        const API\_URL = 'http://localhost:8001';

        // Attach JWT

        $.ajaxPrefilter((opts, \_, xhr) => {

            const tk = localStorage.getItem('access\_token');

            if (!tk) window.location.href = 'login.html';

            xhr.setRequestHeader('Authorization', 'Bearer ' + tk);

        });

        $(function () {

            $('#logoutBtn').click(() => {

                localStorage.removeItem('access\_token');

                window.location.href = 'login.html';

            });

            // Fetch total users

            $.getJSON(`${API\_URL}/users`)

                .done(u => $('#totalUsers').text(u.length))

                .fail(() => toastr.error('Failed to load users'));

            // Fetch metrics

            $.getJSON(`${API\_URL}/metrics`)

                .done(m => {

                    $('#totalTS').text(m.total\_timesheets);

                    $('#pendingTS').text(m.pending\_timesheets);

                    $('#totalPayrolls').text(m.payroll\_runs);

                })

                .fail(() => toastr.error('Failed to load metrics'));

        });

    </script>

</body>

</html>

**Data Structures:**

## Employee:

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Key type | Description |
| EmployeeID | AUTOINCREMENT | Primary Key | Unique employee identifier |
| FirstName | TEXT(50) |  | Employee’s first name |
| LastName | TEXT(50) |  | Employee’s last name |
| Email | TEXT(100) |  | |  | | --- | | Login email address (unique) |  |  | | --- | |  | |
| PhoneNumber | TEXT(20) |  | |  | | --- | | Contact phone number |  |  | | --- | |  | |
| Department | TEXT(50) |  | |  | | --- | | Department or team |  |  | | --- | |  | |
| Password | TEXT(100) |  | |  | | --- | | SHA-256 hash of the user’s password |  |  | | --- | |  | |
| Role | INTEGER |  | 0=Employee, 1=Manager, 2=Admin |

## **Timesheet:**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Key type | Description |
| TimesheetID | AUTOINCREMENT | Primary Key | |  | | --- | | Unique timesheet entry ID |  |  | | --- | |  | |
| EmployeeID | LONG | Foreign Key | |  | | --- | | → Employee (EmployeeID) |  |  | | --- | |  | |
| DateWorked | DATE |  | |  | | --- | | Date of the work |  |  | | --- | |  | |
| HoursWorked | DOUBLE |  | |  | | --- | | Regular hours worked |  |  | | --- | |  | |
| OvertimeHours | DOUBLE |  | |  | | --- | | Overtime hours worked |  |  | | --- | |  | |
| WorkDescription | TEXT(255) |  | |  | | --- | | Free-text description of the work done |  |  | | --- | |  | |
| SubmissionStatus | TEXT(20) |  | |  | | --- | | “Pending”/“Approved”/“Rejected” |  |  | | --- | |  | |

## **Payrate:**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Key type | Description |
| PayrateID | AUTOINCREMENT | Primary Key | Unique payrate record ID |
| EmployeeID | LONG | Foreign Key | → Employee (EmployeeID) |
| EffectiveDate | DATE |  | Date this rate becomes active |
| HourlyRate | CURRENCY |  | |  | | --- | | Base hourly rate (in local currency) |  |  | | --- | |  | |
| OvertimeRate | CURRENCY |  | Hourly rate applied to overtime hours |

## **Payroll:**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Key type | Description |
| PayrollID | AUTOINCREMENT | Primary Key | Unique payroll run ID |
| EmployeeID | LONG | Foreign Key to | → Employee (EmployeeID) |
| PayrollDate | DATE |  | |  | | --- | | Date of the payroll calculation |  |  | | --- | |  | |
| ApprovedHours | DOUBLE |  | |  | | --- | | Sum of approved regular hours for period |  |  | | --- | |  | |
| OvertimeHours | DOUBLE |  | |  | | --- | | Sum of approved overtime hours for period |  |  | | --- | |  | |
| GrossSalary | CURRENCY |  | |  | | --- | | Calculated gross pay before deductions |  |  | | --- | |  | |
| NetPay | CURRENCY |  | GrossSalary minus total deductions |

## **Deductions:**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Key type | Description |
| DeductionID | AUTOINCREMENT | Primary Key | Unique deduction record ID |
| PayrollID | LONG | Foreign Key | → Payroll (PayrollID) |
| DeductionType | TEXT(50) |  | |  | | --- | | Type of deduction (tax, insurance, etc.) |  |  | | --- | |  | |
| Amount | CURRENCY |  | Deduction amount |

Event Functionality Prototype

Fastapi screenshots:

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

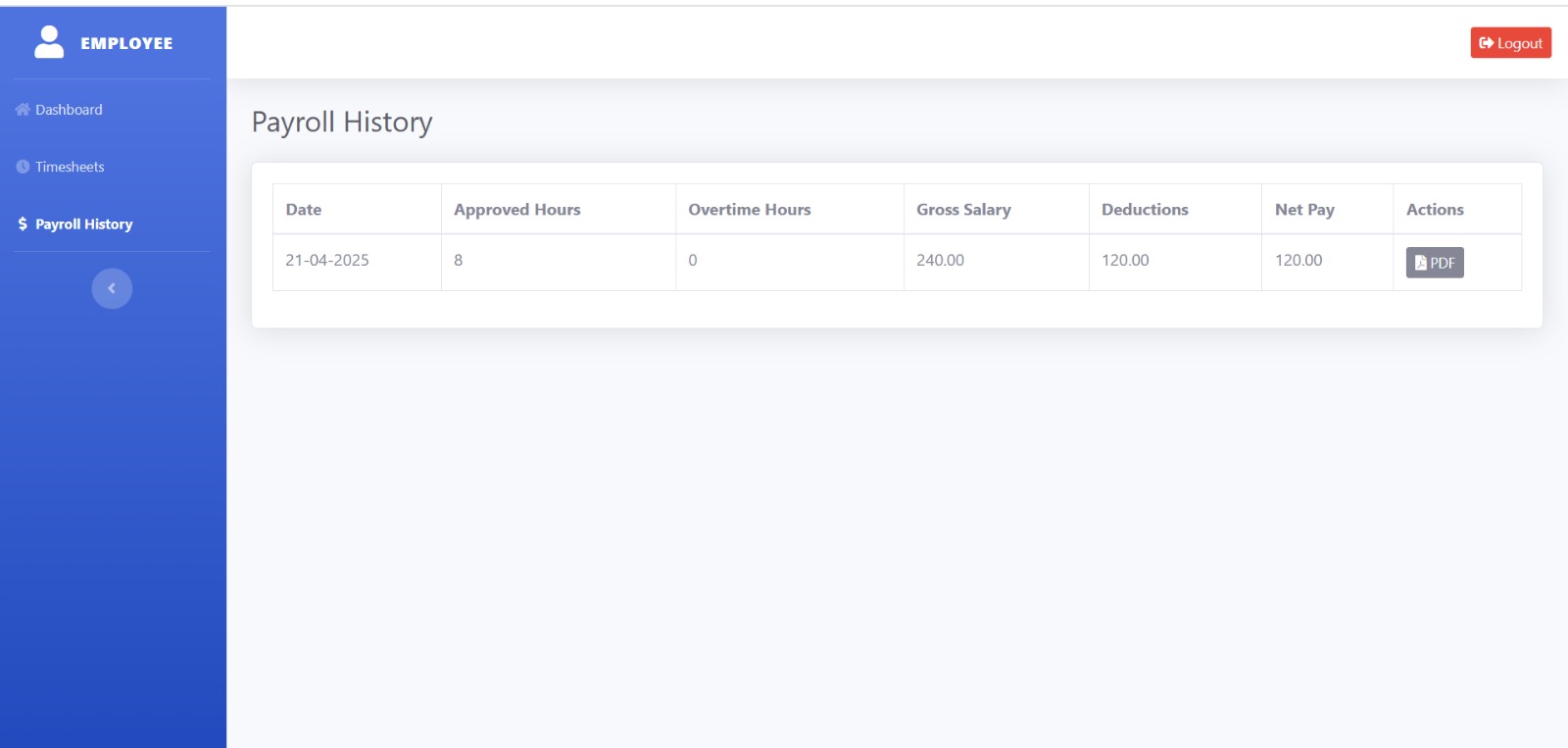
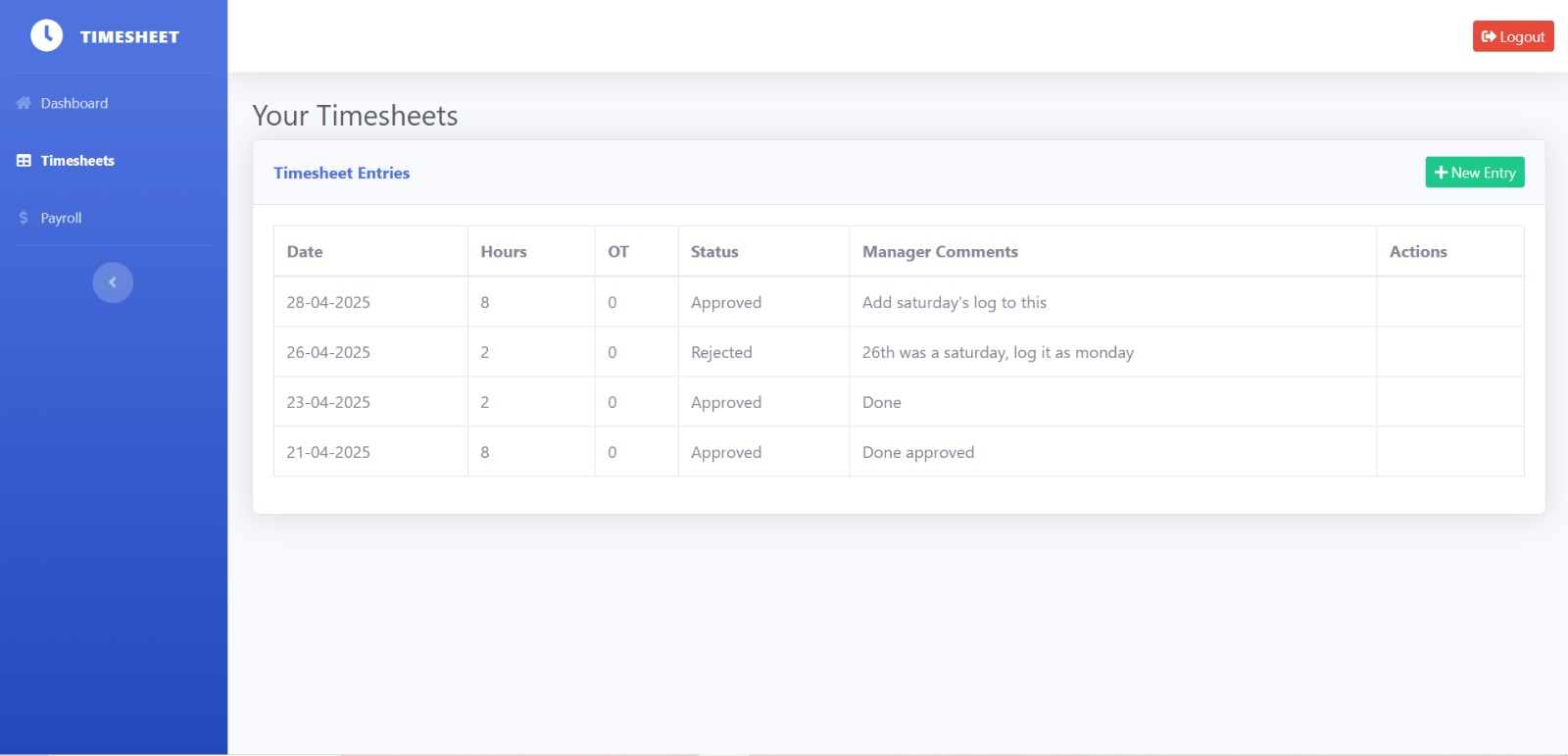
Employee Screenshots:

A screenshot of a login form

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

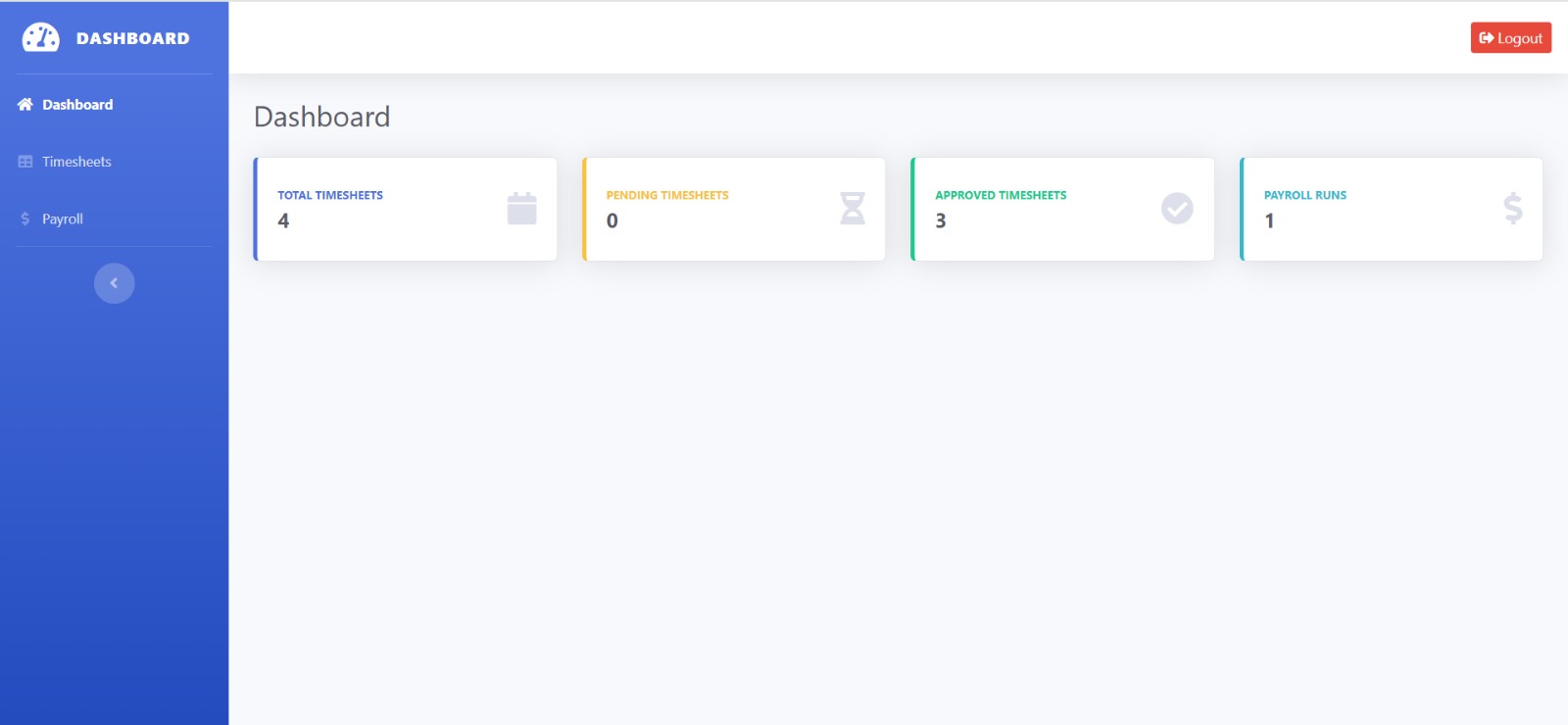


A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a pay slip

AI-generated content may be incorrect.

Manager Screenshots:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Admin Screenshots:

A screenshot of a login form

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## **Conclusion**

The Employee Timesheet and Payroll System successfully meets the project’s objectives of efficiently managing employee work hours, automating payroll calculations, and streamlining administrative processes. By integrating MS Access for database management, SQL for data operations, and Python/FastAPI for automation, the system provides a user-friendly platform for employees, managers, and administrators.

Through systematic development phases — from database design and user interface prototyping to final implementation and testing — the project demonstrates how small organizations can adopt accessible tools to simplify payroll management while maintaining accuracy and compliance.

Although the current system meets the primary functional requirements, future enhancements could include integrating advanced reporting, real-time notifications, and broader API support to scale the system for larger organizations.

Overall, this project provided valuable experience in system analysis, database management, and application development, laying a strong foundation for future, more complex projects.