EMPLOYEE MANAGEMENT SYSTEM

Using Python and SQLite

Submitted by

Amrutha K

Contents

1. INTRODUCTION

The Employee Management System is a command-line based application developed using Python and SQLite. The primary objective of this project is to efficiently manage employee records, attendance, leave requests, and payroll using a centralized database system.

The system supports role-based access control with two types of users:

- **Admin**
- **Employee**

This project demonstrates practical implementation of database concepts, authentication mechanisms, and CRUD operations.

2. SYSTEM REQUIREMENTS

2.1 Software Requirements

- Python 3.x
- SQLite Database
- VS Code / PyCharm
- Windows / Linux / macOS

2.2 Hardware Requirements

- Minimum 4 GB RAM
- Standard Processor

3. FUNCTIONAL REQUIREMENTS

3.1 Admin Functions

- Secure admin login

- Add and manage employee details
- Create employee login credentials
- Manage departments and job roles
- Mark attendance
- Approve or reject leave requests
- Generate payroll

3.2 Employee Functions

- Employee login using ID and password
- View personal profile
- View attendance records
- Apply for leave
- View leave status
- View payroll details

4. DATABASE SCHEMA

4.1 Employee Table

Attribute	**Data Type**	**Description**
employee_id (PK)	INT	Unique employee ID
first_name	VARCHAR	Employee first name
last_name	VARCHAR	Employee last name
email	VARCHAR	Email address
phone	VARCHAR	Contact number
dob	DATE	Date of birth
gender	VARCHAR	Gender

hire_date	DATE	Joining date
department_id (FK)	INT	Department reference
job_id (FK)	INT	Job role reference
salary	DECIMAL	Salary
manager_id (FK)	INT	Reporting manager

4.2 Department Table

department_id (PK)	INT	Unique department ID
department_name	VARCHAR	Department name
location	VARCHAR	Department location

4.3 Job Table

job_id (PK)	INT	Unique job ID
job_title	VARCHAR	Job title
min_salary	DECIMAL	Minimum salary
max_salary	DECIMAL	Maximum salary

4.4 Attendance Table

attendance_id (PK)	INT	Attendance record ID
employee_id (FK)	INT	Employee reference
date	DATE	Attendance date
status	VARCHAR	Present / Absent / Leave

4.5 Leave Table

leave_idUID_id (PK)	INT	Leave record ID
employee_id (FK)	INT	Employee reference
leave_type	VARCHAR	Sick / Casual / Paid
start_date	DATE	Leave start date

end_date	DATE	Leave end date
status	VARCHAR	Approved / Pending / Rejected

4.6 Payroll Table

payroll_id (PK)	INT	Payroll record ID
employee_id (FK)	INT	Employee reference
month	VARCHAR	Salary month
basic_salary	DECIMAL	Basic salary
deductions	DECIMAL	Deductions
net_salary	DECIMAL	Final salary

5. SCREENSHOTS

The following screenshots illustrate the working of the Employee Management System.

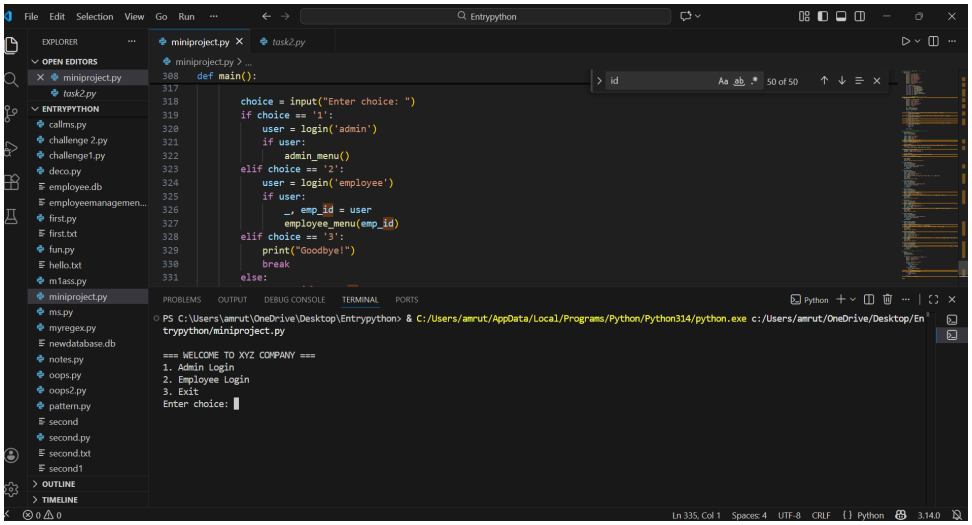

Figure 1: Welcome Screen with Admin and Employee Login Options

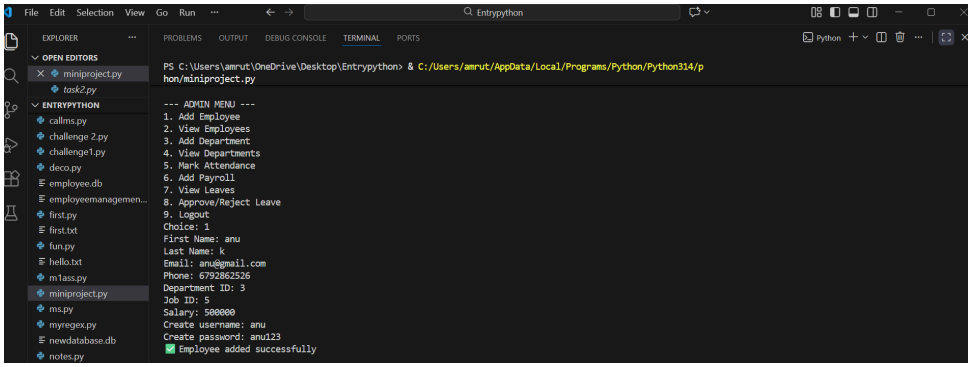

Figure 2: Admin Adding New Employee Details

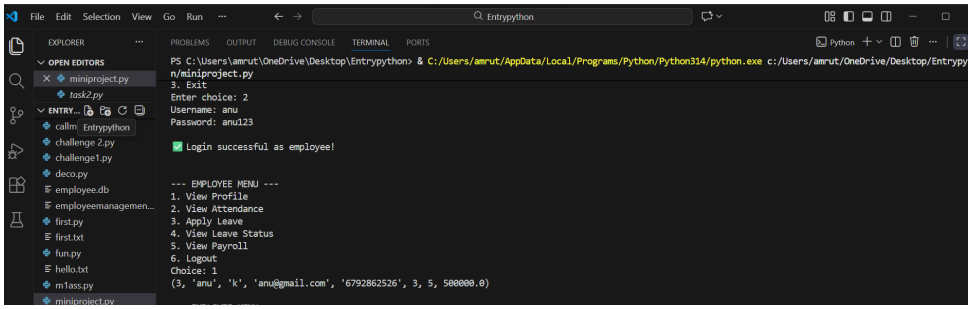

Figure 3: Admin Viewing Employee Records

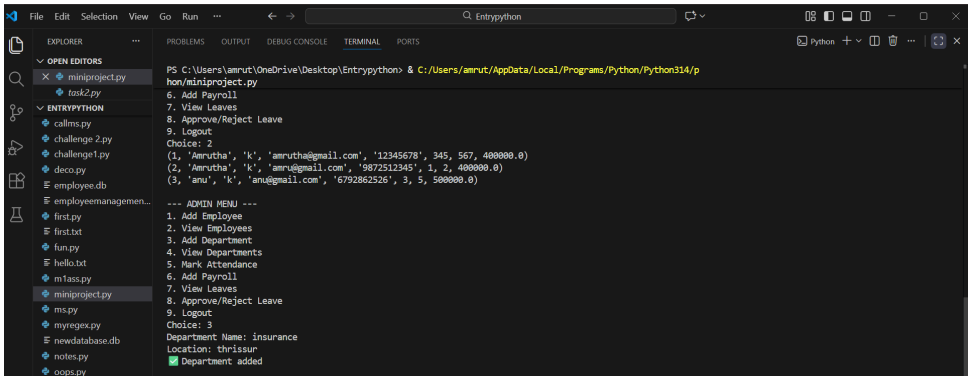

Figure 4: Admin Adding Department Information

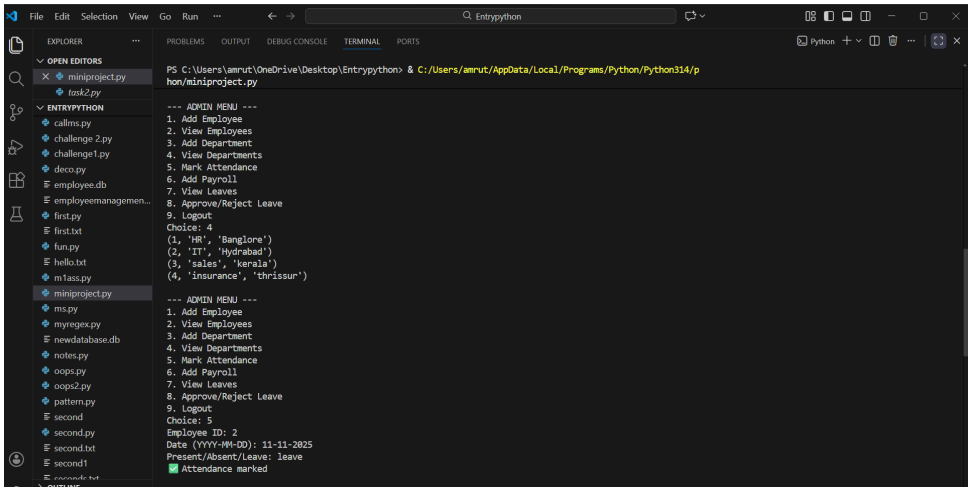

Figure 5: Admin Marking Employee Attendance

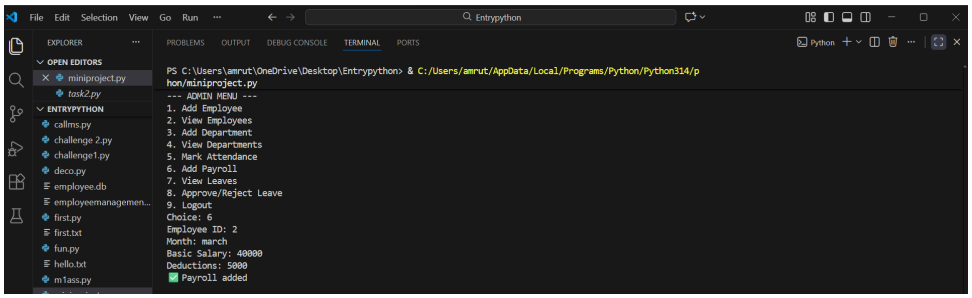

Figure 6: Payroll Generation and Salary Details

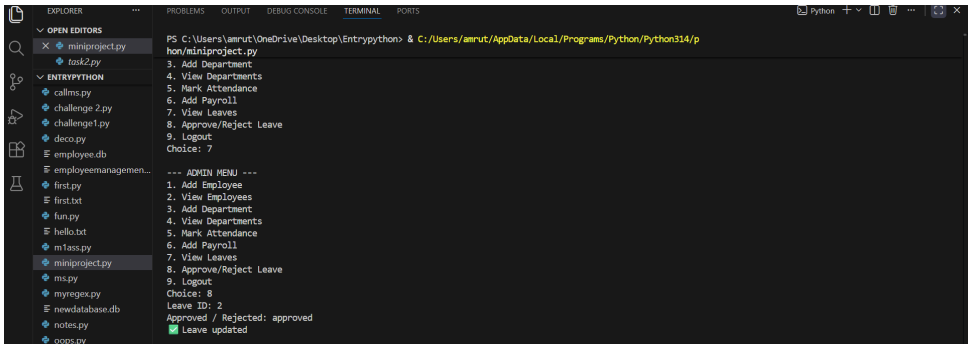

Figure 7: Admin Approving or Rejecting Leave Requests

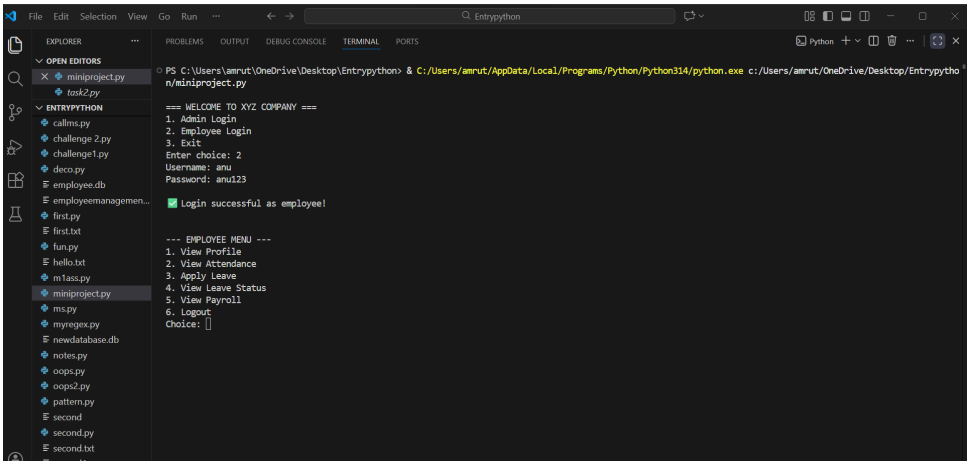

Figure 8: Employee Login Screen

Figure 9: Employee Applying for Leave

6. CONCLUSION

The Employee Management System developed using Python and SQLite provides an efficient and secure solution for managing employee data within an organization. The system enables administrators to control employee records, attendance, leave approvals, and payroll processing, while employees can independently view their information and submit requests.

This project demonstrates practical understanding of database design, authentication, and software development principles. The modular structure and command-line interface make the system lightweight and easy to maintain. Future enhancements may include a graphical user interface, web-based deployment, and enhanced security features for enterprise-level use.