Absolute Global Outsourcing Private Limited

Project report: Human Resource Management System (HRMS)

Project Overview:

The Human Resource Management System (HRMS) is a Flask-based web application designed to streamline various HR processes for an organization. It provides functionalities for both employees and administrators, facilitating efficient management of employee data, leave requests, attendance, and document generation.

Project Goals:

1. Automate HR processes for efficiency
2. Provide employee self-service features
3. Implement leave and attendance management
4. Automate document generation (offer letters, salary slips)
5. Ensure secure, role-based access for employees and admins
6. Integrate email notifications for key processes
7. Enable reporting and data analytics
8. Centralize employee data management
9. Improve HR-employee communication
10. Create a user-friendly, scalable system

These goals aim to streamline HR operations, enhance employee experience, and improve overall organizational efficiency.

Technology Stack:

1. Backend Framework: Flask
   - Flask is a lightweight Python web framework used to build the backend of this application.
   - It provides routing, request handling, and integrates well with other Python libraries.
2. Programming Language: Python
   - Python is the core language used for server-side logic.
   - It's known for its readability and extensive library support.

- Version: Python 3.

3. Database: MongoDB
    - MongoDB is a NoSQL database used for storing application data.
    - It's accessed using the PyMongo driver, which allows Python to interact with MongoDB.
    - Collections are used to store different types of data (employees, leave requests, etc.).
    - MongoDB's flexible schema is well-suited for evolving data structures in HR systems.
4. Frontend:
    - HTML for structuring web pages and CSS for styling
5. Template Engine: Jinja2
    - Jinja2 is used for rendering dynamic content in HTML templates.
    - It's the default template engine for Flask applications.
6. Email Service: Flask-Mail
    - Flask-Mail is used for sending emails from the application.
    - It's configured to use Gmail's SMTP server for email delivery.
7. PDF Generation:
    - WeasyPrint: Used for converting HTML templates to PDF (for documents like offer letters).
    - ReportLab: Used for creating PDF reports (like attendance reports).
8. Security:
    - Bcrypt: Used for hashing passwords before storing them in the database.
    - Flask's built-in session management for user authentication.
9. Date and Time Handling:
    - Python's datetime module for date and time operations.
10. Additional Libraries:
    - itsdangerous: For generating secure tokens (used in password reset functionality).
    - tempfile: For creating temporary files (used in PDF generation)..

This technology stack combines popular, reliable, and well-supported tools in the Python ecosystem. It's designed to create a scalable, maintainable web application with a focus on backend functionality and data management. The use of MongoDB allows for flexible data storage, while Flask provides a lightweight but powerful framework for building web applications. The integration of various libraries for tasks like PDF generation and email sending enhances the system's capabilities in handling HR-specific requirements.

Key Features and Functionality:

- **Employee Features:**

a) Registration and Login:

1. Registration:
   - Employees can register by providing their personal details through a registration form.
   - The registration form captures:
     - Name
     - Email (used as unique identifier)
     - Position
     - Date of Joining
   - Upon submission, the data is stored in the 'employees' collection in MongoDB.
   - The initial status of the registration is set to 'unapproved' (approved: False).
2. Admin Approval:
   - New registrations are queued for admin approval.
   - Admins can view pending registrations in their dashboard.
   - Admins can approve or reject each registration.
3. Login Credential Delivery:
   - Once approved, the system generates a unique token for the employee.
   - An email is sent to the employee's registered email address containing:
     - A welcome message
     - A link to set their password (using the generated token)
   - The email is sent using Flask-Mail, configured with Gmail's SMTP server.
4. Password Setting:
   - Employees click the link in the email to set their password.
   - The password is hashed using bcrypt before being stored in the database.

5. Login Process:
    - Employees can now log in using their email and newly set password.
    - The system verifies the credentials against the stored data.
    - Upon successful login, a session is created for the employee.

b) Leave Management:

1. Leave Request Submission:
    - Logged-in employees can access a leave request form.
    - The form captures:
        - Start date of leave
        - End date of leave
        - Reason for leave
    - The system calculates the number of days requested.
2. Leave Balance Calculation:
    - The system maintains a record of total leaves allocated (e.g., 24 per year).
    - It calculates remaining leaves by subtracting used leaves from total allocated leaves.
3. Leave History:
    - Employees can view a list of their previous leave requests.
    - This includes details such as dates, reasons, and approval status.
4. Leave Balance Display:
    - The current leave balance is displayed to the employee.
    - It shows total leaves, used leaves, and remaining leaves.

c) Attendance Viewing:

1. Weekly Attendance Record:
    - Employees can select a specific week to view their attendance.
    - The system retrieves attendance data from the 'attendance_request' collection.
2. Attendance Details:
    - For each day in the selected week, the following is displayed:
        - Date
        - Presence status (Present/Absent)
        - Login time (if present)

    - Logout time (if present)
    - Hours worked (calculated from login and logout times)
3. Total Hours Calculation:
    - The system calculates and displays the total hours worked for the selected week.

d) Document Requests:

1. Document Selection:
    - Employees can choose from different document types:
        - Offer letter
        - Salary slip
        - Other report
2. Offer Letter Request:
    - Employees fill a form with details like:
        - Recipient name
        - Company name
        - Reporting person
        - Joining date
    - The request is stored in the 'letter_requests' collection.
3. Salary Slip Request:
    - Employees provide details such as:
        - Month for which salary slip is needed
        - Gross salary
        - Days attended
    - The request is stored in the 'salary_slip_collection'.
4. Request Status:
    - Employees can view the status of their document requests (pending/approved).
5. Document Delivery:
    - Once approved and generated by admin, documents are sent to the employee's email.

These features provide a comprehensive self-service portal for employees, allowing them to manage their leaves, view attendance, and request important documents, all while maintaining a streamlined workflow that involves necessary admin approvals.

- **Admin Features:**

a) Employee Management:

1. Approving New Employee Registrations:
    - Admins can view a list of pending employee registrations in their dashboard.
    - For each pending registration, the admin can see details like name, email, position, and date of joining.
    - The admin has options to approve or reject each registration.
    - Upon approval:
        - The employee's status is updated to 'approved' in the database.
        - An email is automatically sent to the employee with login credentials.
    - The approval process triggers the 'send_login_credentials' function, which:
        - Generates a secure token for password reset.
        - Creates a password reset URL.
        - Sends an email to the employee with instructions to set their password.
2. Viewing All Employee Details:
    - Admins have access to a comprehensive list of all approved employees.
    - They can view details such as:
        - Employee name
        - Email
        - Position
        - Date of Joining
        - Current status
    - This feature allows admins to quickly access and review employee information.

b) Leave Management:

1. Approving or Disapproving Leave Requests:
    - Admins can view a list of pending leave requests in their dashboard.

- Each leave request shows:
  - Employee name
  - Leave start and end dates
  - Reason for leave
  - Number of days requested
- Admins have options to approve or disapprove each request.

2. Leave Approval Process:
   - When approving a leave:
     - The leave status is updated in the database.
     - An approval email is sent to the employee.
     - The employee's leave balance is updated.
3. Leave Disapproval Process:
   - When disapproving a leave:
     - Admins are required to provide a reason for disapproval.
     - This reason is recorded in the database.
     - An email is sent to the employee notifying them of the disapproval and the reason.
4. Leave Balance Management:
   - The system automatically updates leave balances based on approvals.
   - Admins can view current leave balances for all employees.

c) Attendance Management:

1. Filling Daily Attendance:
   - Admins have access to a daily attendance form.
   - This form lists all employees with options to mark:
     - Present/Absent status
     - Login time (for present employees)
     - Logout time (for present employees)
   - The system calculates hours worked based on login and logout times.
2. Bulk Attendance Entry:
   - Admins can fill in attendance for multiple employees at once.
   - The data is stored in the 'attendance_request' collection.
3. Viewing Attendance Reports:
   - Admins can select a specific date to view attendance records.
   - The report shows:

        - Employee name
        - Present/Absent status
        - Login and logout times
        - Hours worked
4. Downloading Attendance Reports:
    - Admins can generate PDF reports of attendance data.
    - The PDF generation uses the ReportLab library.
    - Reports can be customized for specific date ranges.
    - The generated PDF includes a table with attendance details for all employees.

d) Document Management:

1. Approving Document Requests:
    - Admins can view pending requests for documents like offer letters and salary slips.
    - Each request shows the employee name and the type of document requested.
2. Generating Offer Letters:
    - Upon approval, the system generates an offer letter using a predefined template.
    - The template is filled with specific employee details.
    - The offer letter is generated as a PDF using WeasyPrint.
3. Generating Salary Slips:
    - Admins can approve salary slip requests.
    - The system calculates salary components based on the gross salary.
    - A salary slip PDF is generated using a template.
4. Document Approval and Sending Process:
    - Once a document is approved and generated:
        - Its status is updated in the database.
        - An email is automatically sent to the employee.
        - The generated PDF is attached to the email.
5. Email Notifications:
    - The system uses Flask-Mail to send emails.
    - Emails include:
        - A personalized message to the employee.
        - The requested document as an attachment.

These admin features provide comprehensive control over employee management, leave processing, attendance tracking, and document generation. They are designed to streamline HR processes, ensure accurate record-keeping, and facilitate efficient communication with employees.

- **Detailed System Architecture:**

1. User Authentication:

1. Separate Login Routes:
   - Employee Login ('/'):
     - Validates email and password against the 'employees' collection.
     - Checks for 'approved' status and 'employee' role.
   - Admin Login ('/admin/login'):
     - Validates against the same 'employees' collection.
     - Checks for 'admin' role.
2. Password Hashing:
   - Uses bcrypt library for secure password hashing.
   - Process:
     - When a password is set or changed, it's hashed before storage.
     - During login, the entered password is hashed and compared with the stored hash.
   - Bcrypt automatically handles salt generation and incorporation.
3. Session Management:
   - Utilizes Flask's session object to maintain user state.
   - On successful login:
     - For employees: sets session['email'] = user's email
     - For admins: sets session['admin_email'] = admin's email
   - Sessions are used to protect routes and maintain login state across requests.

2. Database Structure:

MongoDB is used with the following collections:

1. 'employees' Collection:
   - Stores employee data including:

        - Name, email, position, date of joining
        - Hashed password
        - Role (employee/admin)
        - Approval status

2. 'leave_requests' Collection:
    - Stores leave request data:
        - Employee email
        - Start and end dates
        - Reason for leave
        - Approval status
3. 'letter_requests' Collection:
    - Stores document request data for offer letters:
        - Employee email
        - Recipient name, company name
        - Reporting person, joining date
        - Approval status
4. 'attendance_request' Collection:
    - Stores daily attendance records:
        - Employee email
        - Date
        - Present/Absent status
        - Login and logout times
        - Hours worked
5. 'salary_slip_collection' Collection:
    - Stores salary slip request and data:
        - Employee email and name
        - Month and year
        - Gross salary, days attended
        - Approval status

3. Email Integration:

1. Flask-Mail Configuration:
    - Set up with Gmail SMTP server:
        - MAIL_SERVER = 'smtp.gmail.com'
        - MAIL_PORT = 587
        - MAIL_USE_TLS = True
    - Credentials are stored in app.config

2. Email Sending Functions:
   - send_login_credentials(): Sends account activation emails.
   - send_leave_approval_email(): Notifies about leave request status.
   - send_letter_email(): Sends generated documents.
   - send_salary_slip_email(): Sends salary slips.
3. Email Content:
   - Uses HTML templates for structured email content.
   - Includes personalized information and attachments where necessary.

4. PDF Generation:

1. WeasyPrint:
   - Used for generating offer letters.
   - Process:
     - HTML template is rendered with Jinja2.
     - WeasyPrint converts the HTML to PDF.
2. ReportLab:
   - Used for generating attendance reports.
   - Creates tables and formats data into PDF.
3. Salary Slip Generation:
   - Combines template rendering with PDF generation.
   - Calculates salary components dynamically.

5. Security Measures:

1. Password Reset Functionality:
   - Uses itsdangerous. URLSafeTimedSerializer for generating secure tokens.
   - Process:
     - Admin approves registration.
     - A time-limited token is generated and sent via email.
     - Token is validated when the reset link is accessed.
     - Tokens expire after a set time (e.g., 3600 seconds).
2. Session-based Authentication:
   - Routes are protected using session checks.

This architecture provides a robust foundation for the HRMS, with secure user authentication, efficient data management, integrated communication systems, and dynamic document generation capabilities. The use of MongoDB allows for flexible data storage and retrieval, while the various security measures ensure the protection of sensitive employee information.

- **User Interface:**

  - The application uses HTML templates for rendering pages.
  - Flash messages are used for user notifications.
  - Forms are used for data input in various functionalities.

- **Error Handling and Logging:**

  - The application includes basic error handling for database operations and email sending.
  - Flash messages are used to communicate errors to users.

- **Future Enhancements:**

  - Implement more robust error handling and logging.
  - Add data visualization for attendance and leave statistics.
  - Implement a more sophisticated salary calculation system.
  - Enhance the security features, possibly implementing two-factor authentication.
  - Develop a mobile application for easier access.

- **Conclusion:**

The HRMS application provides a comprehensive solution for managing various HR processes. It offers a user-friendly interface for employees to manage their leaves and access important documents, while providing administrators with tools to efficiently manage employee data, attendance, and document requests. The use of modern web technologies ensures scalability and maintainability of the system.