Bytexl's guided project

Final Project report

Name of the educator	N Narasimha
Project title	Intelligent Faculty Leave Management with Proxy Assignment: A Comprehensive System Approach
Tools / platforms used	Nimbus and Vscode

About the project:

The project is an Intelligent Faculty Leave Management System with Proxy Assignment. This system automates the process of managing faculty leave requests and assigning proxy faculty members for lectures. It allows administrators to review leave requests, approve or reject them, and assign proxies to cover the lectures in the absence of the faculty member. The system also sends notifications to both the original faculty and the proxy faculty to keep them informed of the changes.

System requirements:

Software Requirements

Flask: Web framework used to build the application.
MySQL: Database management system used for storing leave request details, faculty information, proxy assignments, etc.
Nimbus Platform: A cloud platform to host the application and manage the infrastructure.
Flask-Mail: To send email notifications for proxy assignments and leave request updates.
Python 3.x: Programming language for backend development.
HTML/CSS/JS: Frontend technologies for user interfaces.
Bootstrap: Frontend framework for styling and responsiveness.
Jinja2: Templating engine used by Flask for rendering dynamic HTML pages.

Hardware Requirements

A computer with at least 4 GB RAM and 2.0 GHz processor to run the web application in a development or testing environment.
Server resources in Nimbus for hosting the application.
Internet connection to access and manage cloud services and email notifications.

Functional requirements:

- **Admin Interface:**
  - View Leave Requests: Admin can view all leave requests submitted by faculty.
  - Approve or Reject Leave Requests: Admin can approve or reject the leave requests.
  - Assign Proxy Faculty: Admin can assign proxy faculty for approved leave requests.
  - View Proxy Assignments: Admin can view all proxy assignments and manage faculty details.
- **Leave Request Handling:**
  - Faculty Leave Requests: Faculty members can submit leave requests for specific dates and times, specifying the reason for the leave.

- Admin Approval or Rejection: Admin reviews and approves or rejects leave requests based on the provided details.

 Proxy Assignment:
- Assign Proxies: Admin assigns proxy faculty to cover for approved leave requests based on their schedule and subject.
- Notifications: Proxies and the original faculty member are notified via email about the proxy assignments.

 User Interfaces:
- **Admin Interface:** A dashboard for admins to manage leave requests, approve/reject requests, and assign proxy faculty.
- **Faculty Interface:**
  - Faculty members can submit leave requests, specifying the date, time, and reason for the leave.
  - Faculty can view the status of their leave requests, whether approved, rejected, or pending.
  - Faculty members are notified via email when their leave request is approved or when a proxy is assigned.

 Notification System:
- Email notifications are sent to both faculty members and proxies about the approval of leave requests and the assignment of proxy faculty.

Inputs and Outputs:

Inputs:
- **Leave Requests**: Faculty members input the date, time, and reason for their leave.
- **Proxy Assignments**: Admin inputs the proxy faculty details along with the slot and subject.
- **Emails**: Faculty and proxies receive emails about leave approval and proxy assignments.

Outputs:
- **Leave Status**: Whether the leave request is approved or rejected.
- **Proxy Assignment**: Details of the proxy faculty and the scheduled time for proxy assignments.
- **Email Notifications**: Informing faculty and proxies about leave approvals and proxy assignments.

List of subsystems:

 Leave Request Handling:
- Subsystem to manage leave requests, track their status, and manage approvals or rejections.

 Proxy Assignment Management:
- Subsystem that handles the assignment of proxy faculty for approved leave requests.

 User Authentication:
- Login system to secure access to admin and faculty panels.

 Email Notification:
- Subsystem for sending email alerts to faculty and proxies.

 Database Management:
- Database system to store faculty details, leave requests, proxy assignments, and related data.

Other Applications relevant to your project:

 HR Systems: Similar systems could be used in corporate HR departments to handle employee leave and proxy management.

 University Leave Management: This project can be extended to manage leave requests for other staff members (e.g., administrative staff, lab assistants) in an educational institution.

 Substitute Teacher Management: It can be adapted to manage substitute teacher assignments in schools.

Designing of Test cases:

Test Cases:

1. **Test Case: Submit Leave Request**
    - **Function**: Ensure that faculty can successfully submit a leave request with valid inputs (date, time, reason).
    - **Expected Output**: Leave request should be saved in the database, and a confirmation should be sent.
2. **Test Case: Reject Leave Request**
    - **Function**: Ensure that the admin can reject a leave request.
    - **Expected Output**: Leave status should change to "Rejected" and a notification should be sent to the faculty.
3. **Test Case: Approve Leave Request**
    - **Function**: Ensure that the admin can approve a leave request and assign proxies.
    - **Expected Output**: Leave status should change to "Approved", and proxies should be notified via email.
4. **Test Case: Assign Proxy**
    - **Function**: Ensure that a proxy faculty is correctly assigned to cover the lecture during the leave.
    - **Expected Output**: A proxy assignment should be added to the database, and email notifications should be sent to both the original and proxy faculty.
5. **Test Case: Check Email Notifications**
    - **Function**: Ensure that emails are sent correctly to the faculty and proxy.
    - **Expected Output**: The faculty and proxy should receive emails about the leave approval and proxy assignment.
6. **Test Case: View Leave Requests**
    - **Function**: Ensure that the admin can view a list of all leave requests and their statuses.
    - **Expected Output**: Admin should be able to view and manage all leave requests.

Future Work:

 Integration with Calendar: Sync leave requests with calendar applications (e.g., Google Calendar, Outlook).
 Mobile App: Develop a mobile app to allow faculty and admin to manage leave requests and proxy assignments on the go.
 Advanced Notification System: Implement SMS or push notifications along with email alerts.

References:

- Flask Documentation
- MySQL Documentation
- Python Programming Guide

Reflection of the project creation:

• Describe the *technical challenges* you encountered in the development of your project

One major challenge was ensuring that proxy assignments were handled efficiently and the correct notifications were sent to both the proxy and the original faculty. Debugging the email notifications took time due to SMTP configuration issues.

• Describe how your existing software engineering knowledge / techniques helped you to address those challenges

My experience with Flask, Python, and database management helped me design and implement the application. Techniques like MVC architecture and database normalization were key to building a clean and efficient application.

- What benefits did you individually experience while working on this project?

The project allowed me to apply my knowledge of Flask, MySQL, and web development in a real-world scenario. It also helped me improve my skills in backend development and user interface design.

- Describe what other knowledge you feel might have helped you with the project development

Knowledge of advanced database queries, email service integrations, and cloud platform management (Nimbus) could have improved the deployment and scalability aspects of the project.