


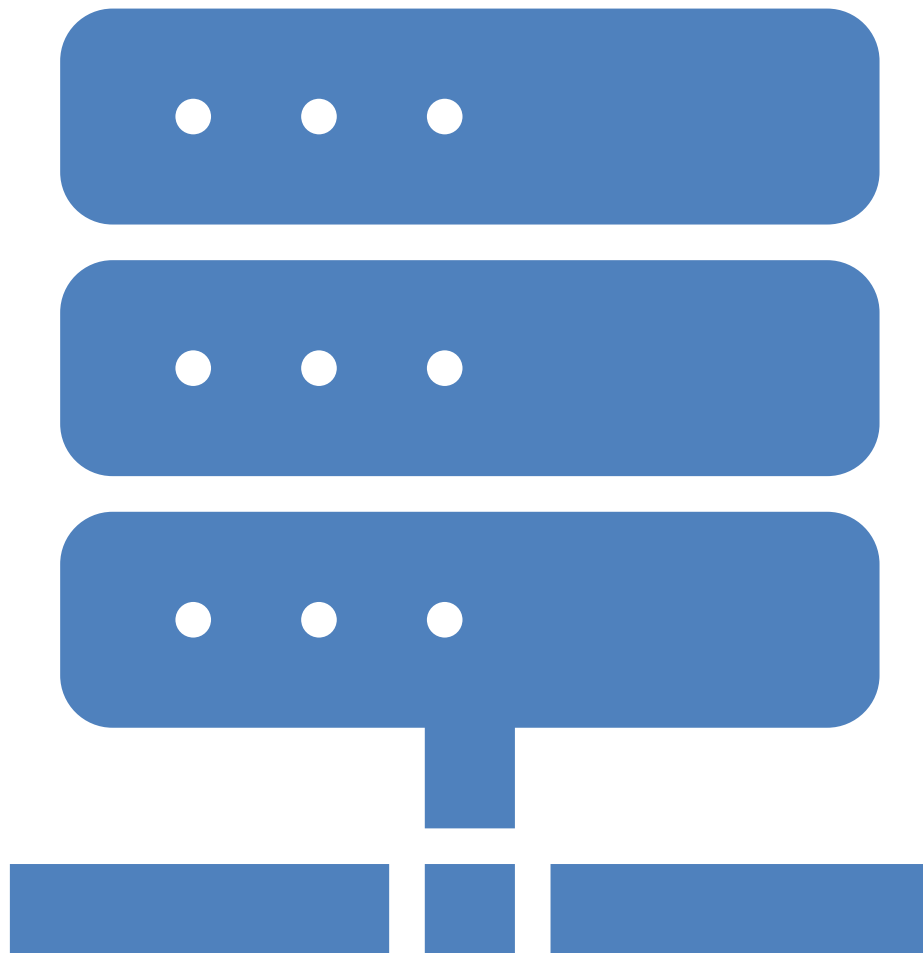
Personal

project

The background of the slide is a complex digital-themed abstract. It features a dark blue and purple gradient with glowing blue and white binary code (0s and 1s) scattered throughout. Overlaid on this are several data visualization elements: a line graph with a yellow and orange line, a bar chart with green and blue bars, and a network diagram with blue nodes and connecting lines. The overall effect is a sense of high-tech, data-driven technology.

ABOUT THE PROJECT

- Designed and developed an Electronic Accounting Management System aimed at facilitating accounting operations and improving work efficiency. Implemented using modern technologies such as PHP, MySQL, and HTML



FUNCTION REQUIREMENTS

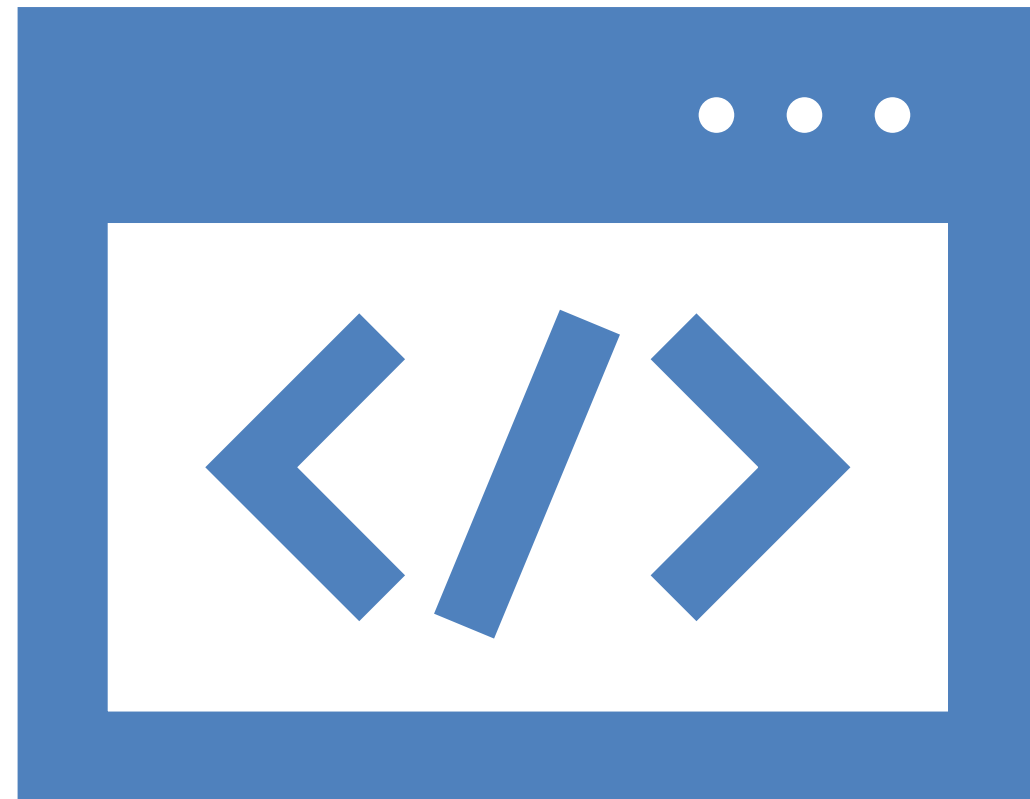
- 1- Allow normal employee to submit maintenance requests via a web form.
- 2- Store request data in a MySQL database.
- 3- Enable admin to view login logs
- 4- Enable normal employees to view their previous requests
- 5- Allow the Technical to update the status of each request.
- 6- Provide create new user for admin access only.
- 7- Display all requests in a table with key details (e.g., name, issue, status).

NON-FUNCTIONAL REQUIREMENTS

- 1- Safety
- 2- speed
- 3- Encryption
- 4- Ease of use



SYSTEM-DESIGN:



- Implemented a login system to restrict access to the admin dashboard.
- Focused on core features such as submitting and managing maintenance requests.
- Structured the database to store user requests and status updates efficiently.

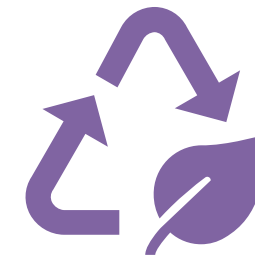
TECHNOLOGIES USED



Programming Languages: PHP,
HTML, CSS , JS



Database:MYSQL

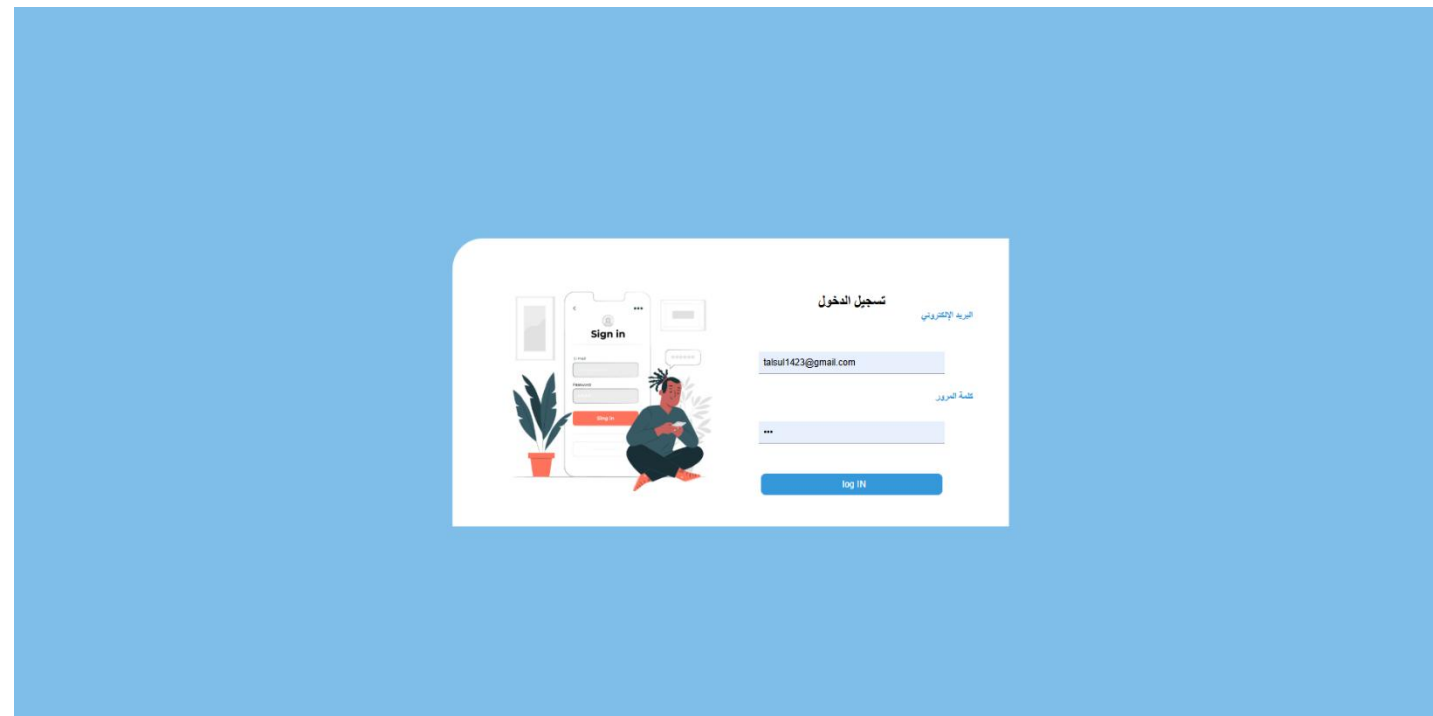


Development Environment:
XAMPP, NetBeans, Visual Studio

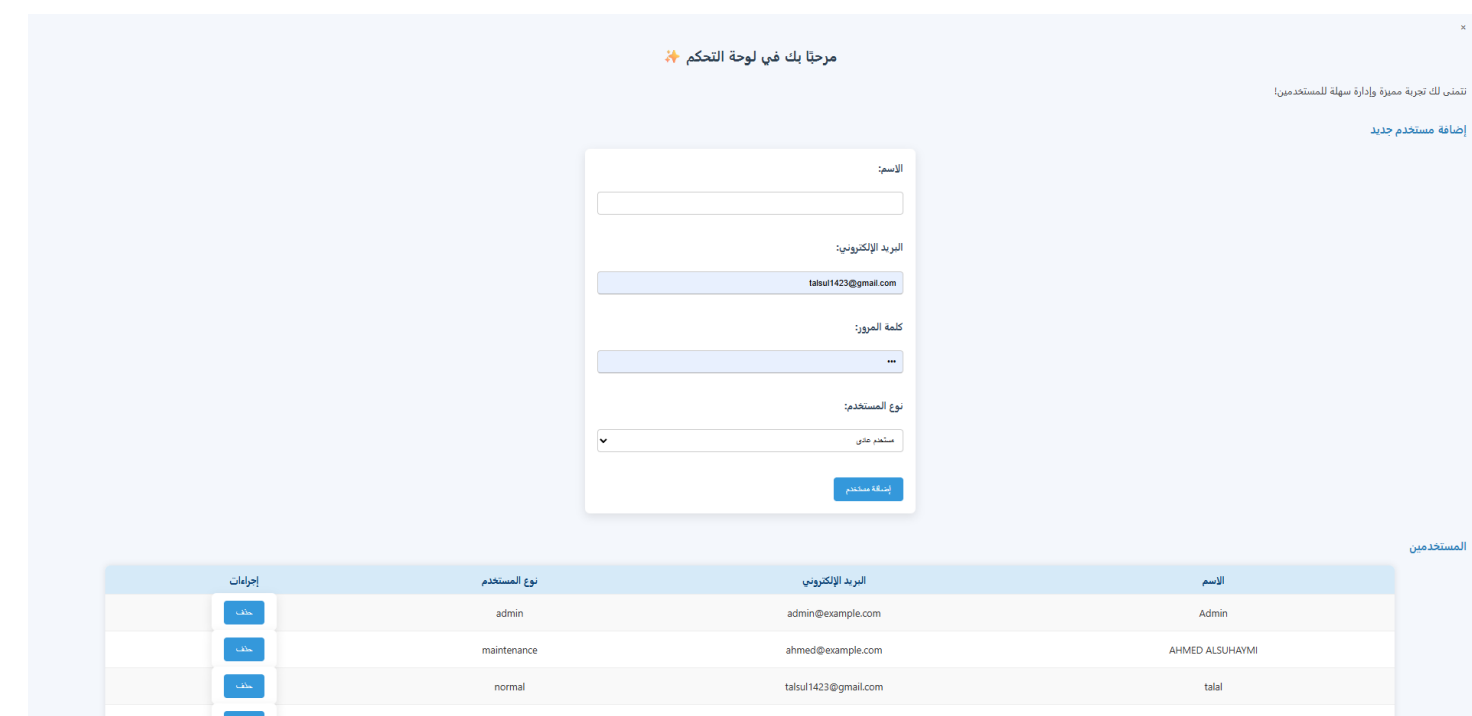


Design Tools: Draw.io,
Microsoft Access

PICTURES FROM THE PROJECT

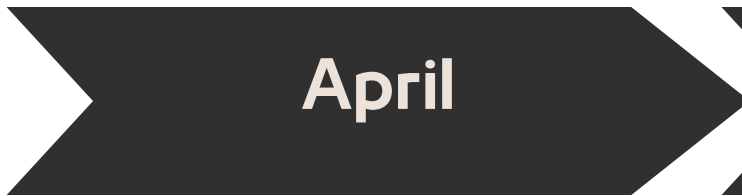


Login Page

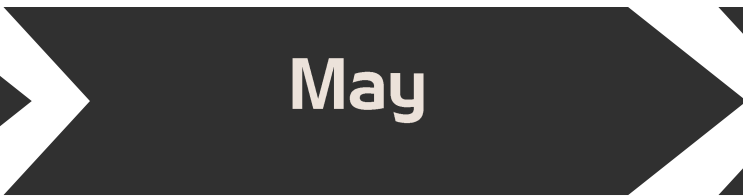


admin dashboard

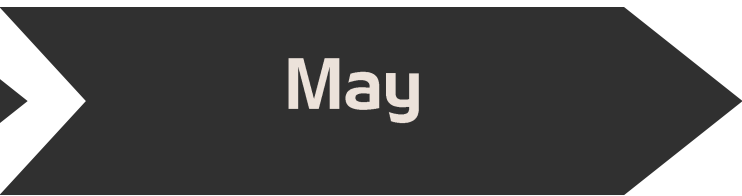
TIME LINE



Database



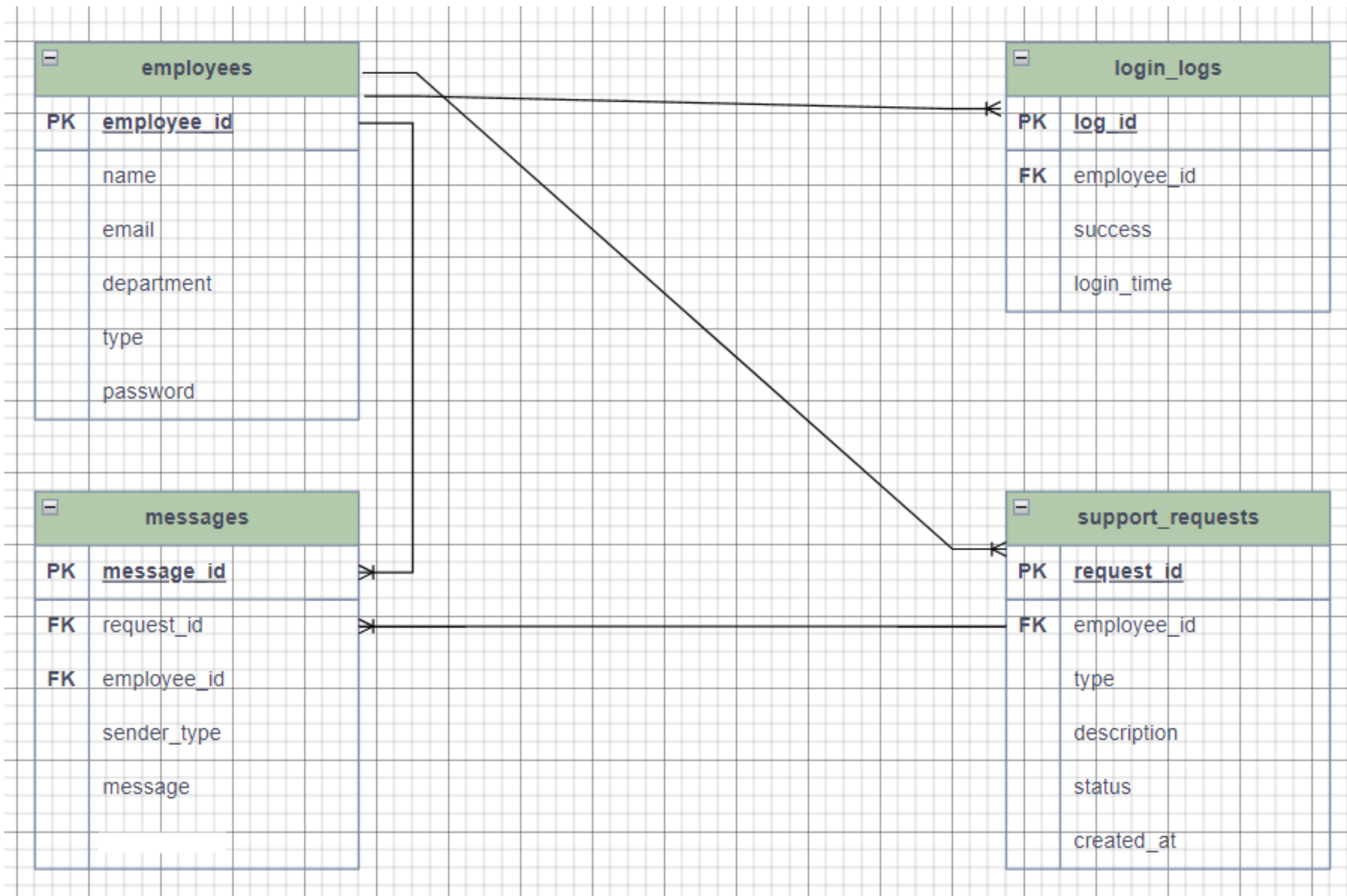
Back-End



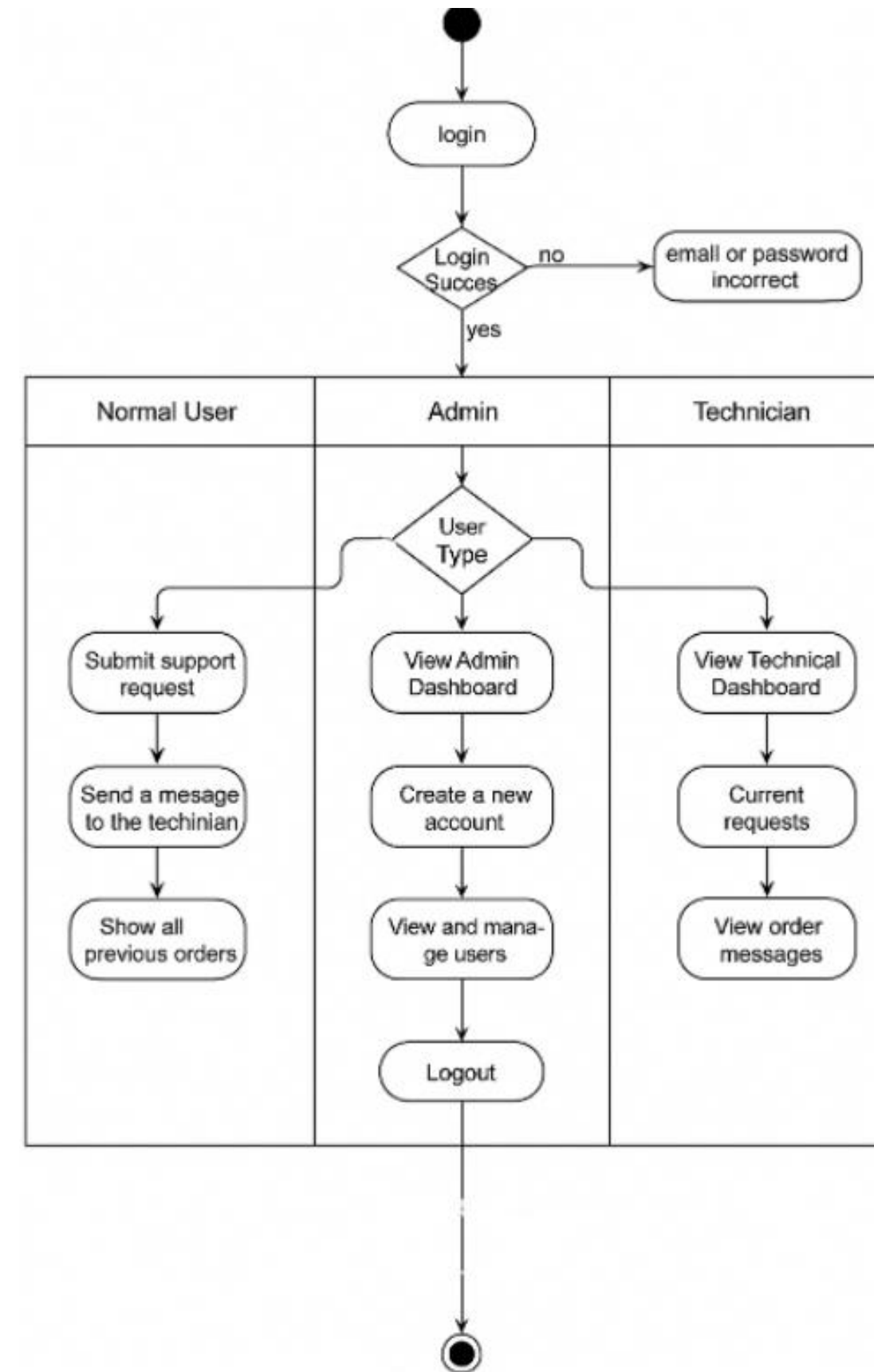
Front-End

ERD DIAGRAM

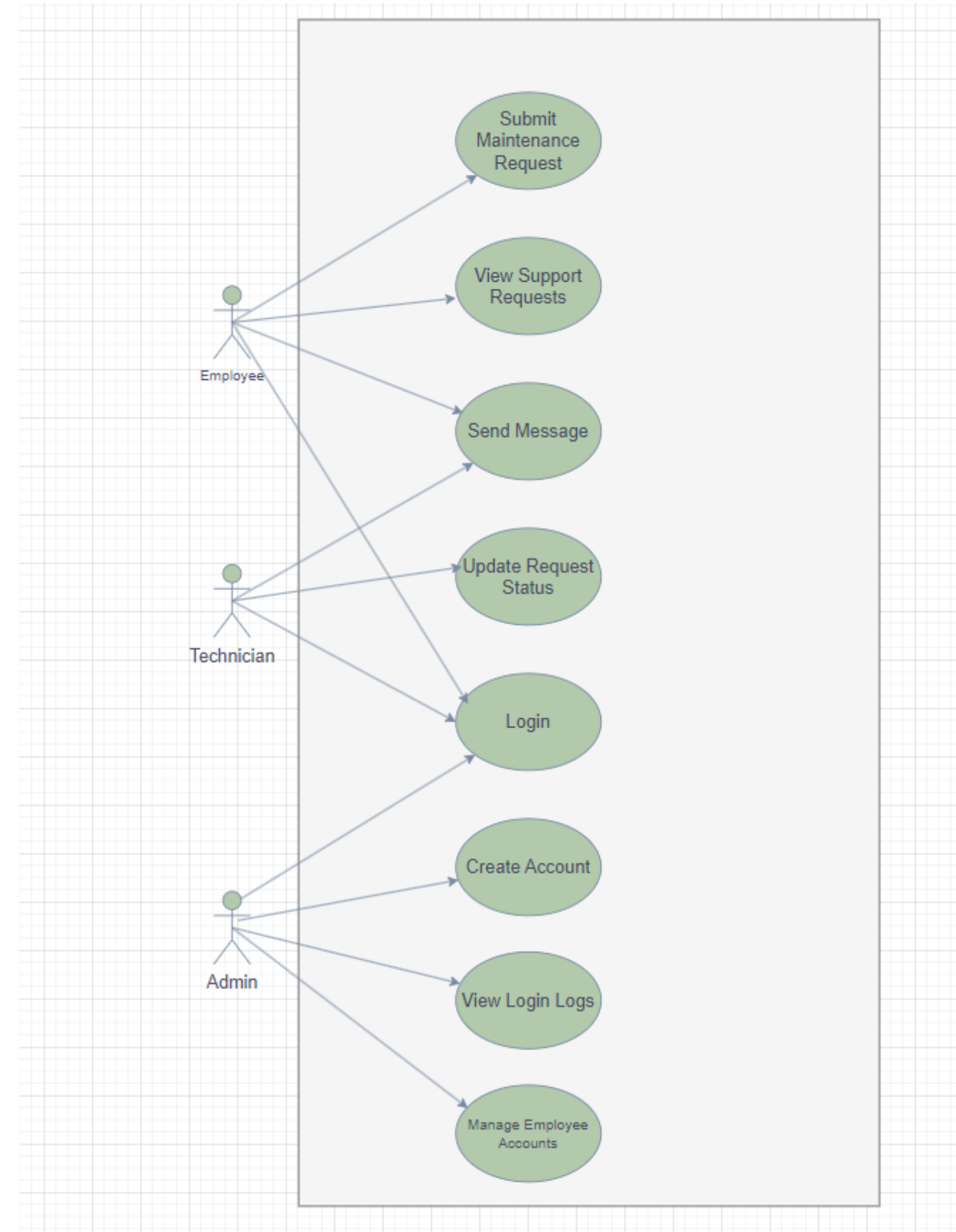
- The ERD for the Maintenance Request Management System was designed to represent the relationships between key entities such as users, maintenance requests, and request statuses. The main entities include:
 - Users:** Stores admin login credentials.
 - Requests:** Contains information about each maintenance request, including requester name, contact details, description, date, and status.
 - Status:** Defines the current state of each request (e.g., Pending, In Progress, Completed).The relationships between these entities ensure efficient data handling, easy tracking of requests, and streamlined workflow for maintenance management.



ACTIVITY DIAGRAM:



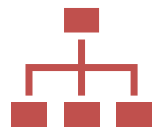
USE CASE DIAGRAM




TESTING

Test Case	Description	Input	Expected Result	Pass/Fail
TC01 – Submit Request	Test submitting a maintenance request	Name + Issue description + Contact info	Request saved in database, success message displayed	Pass
TC02 – Admin Login	Test admin login functionality	Username + Password	Redirect to admin dashboard if credentials are valid	Pass
TC03 – Invalid Login	Test login with incorrect credentials	Username + Wrong password	Error message shown: "Invalid credentials"	Pass
TC04 – View Requests	Test viewing all submitted requests	None	Table showing all requests with details is displayed	Pass
TC05 – Update Status	Test updating request status	Select request + Change status	Status is updated and saved in the database	Pass
TC06 – Empty Form	Test submitting form without filling fields	Submit with empty fields	Show validation messages requesting required fields	Pass

Tasks I Handled




 **Planning & Analysis –**
Understanding project
requirements and defining
scope




 **Database Design –** Creating
ERD and building MySQL tables




 **Frontend Development –**
Designing user interface using
HTML, CSS, and JavaScript




 **Backend Development –**
Building system logic using PHP
and connecting to the database




 **Authentication –**
Implementing admin login
system



 **Testing –** Performing
manual tests to ensure
functionality and fix bugs



 **Documentation –** Writing
reports and describing system
structure clearly

**THANKS FOR YOUR
TIME.**

