**Project on Employee Management System**

**By Group no. 2 – AIML (C)**

|  |  |
| --- | --- |
| 1. Shreya Shahi | 6. Pawan Shetty |
| 2. Aayan Shaikh | 7. Rishith Shetty |
| 3. Ayan Shaikh | 8. Aryan Shinde |
| 4. Saaem Shaikh | 9. Rohit Shinde |
| 5. Amankumar Sharma | 10. Anmol Shukla |

**An Innovative Examination (IE) Report**

**Submitted for the**

**Subject of**

**Programming for Problem Solving**

**Under the Guidance of**

**Ms. Priya Salunke**

**Asst. Professor**

**A.Y. 2024-25**

**Abstract:**

**The Employee Management System (EMS) is a web-based application designed to streamline and enhance the management of human resources within an organization. Built using Python, Flask, Appwrite, and Bootstrap, EMS offers a modern, user-friendly interface for managing employees, departments, and HR-related tasks efficiently. Currently in its early development stages, EMS aims to provide an intuitive and robust solution, with future updates driven by user feedback.**

**Introduction:**

**Human Resource Management is a crucial aspect of any organization, involving tasks like employee data management, leave tracking, and departmental coordination. The Employee Management System (EMS) addresses these needs by providing a digital platform that simplifies HR tasks, ensuring efficiency and accuracy. This report outlines the system's architecture, implementation, and potential for future enhancements.**

**Background:**

Managing human resources manually or through outdated systems can lead to inefficiencies, data errors, and time-consuming processes. EMS offers a solution by leveraging modern web technologies to create a scalable, secure, and easy-to-use HR management tool. By integrating a responsive front-end and a reliable backend, EMS ensures seamless interaction and data management.

**Objectives:**

* Automate HR tasks like employee registration, leave requests, and department management.
* Provide a secure platform for storing employee data.
* Enhance productivity through a user-friendly interface.

**System Description:**

**Block Diagram:**

Below is a conceptual block diagram representing the EMS architecture:

A black background with pink and purple text

Description automatically generated

**Module Overview:**

* **Admin Module:** Handles employee and department management, leave approvals, and advance tracking.
* **Authentication Module:** Ensures secure user login, registration, and password management.
* **User Module:** Allows employees to request leaves, advances, and view their follow-ups.

**System Workflow:**

1. **Login/Register:** Users log in or register via the authentication module.
2. **Admin Dashboard:** Administrators can manage employees, view leave requests, and track advances.
3. **User Dashboard:** Employees can manage their profile, request leaves, and check approvals.
4. **Database Interaction:** The Flask SQL Alchemy ORM connects the application to the Appwrite database.

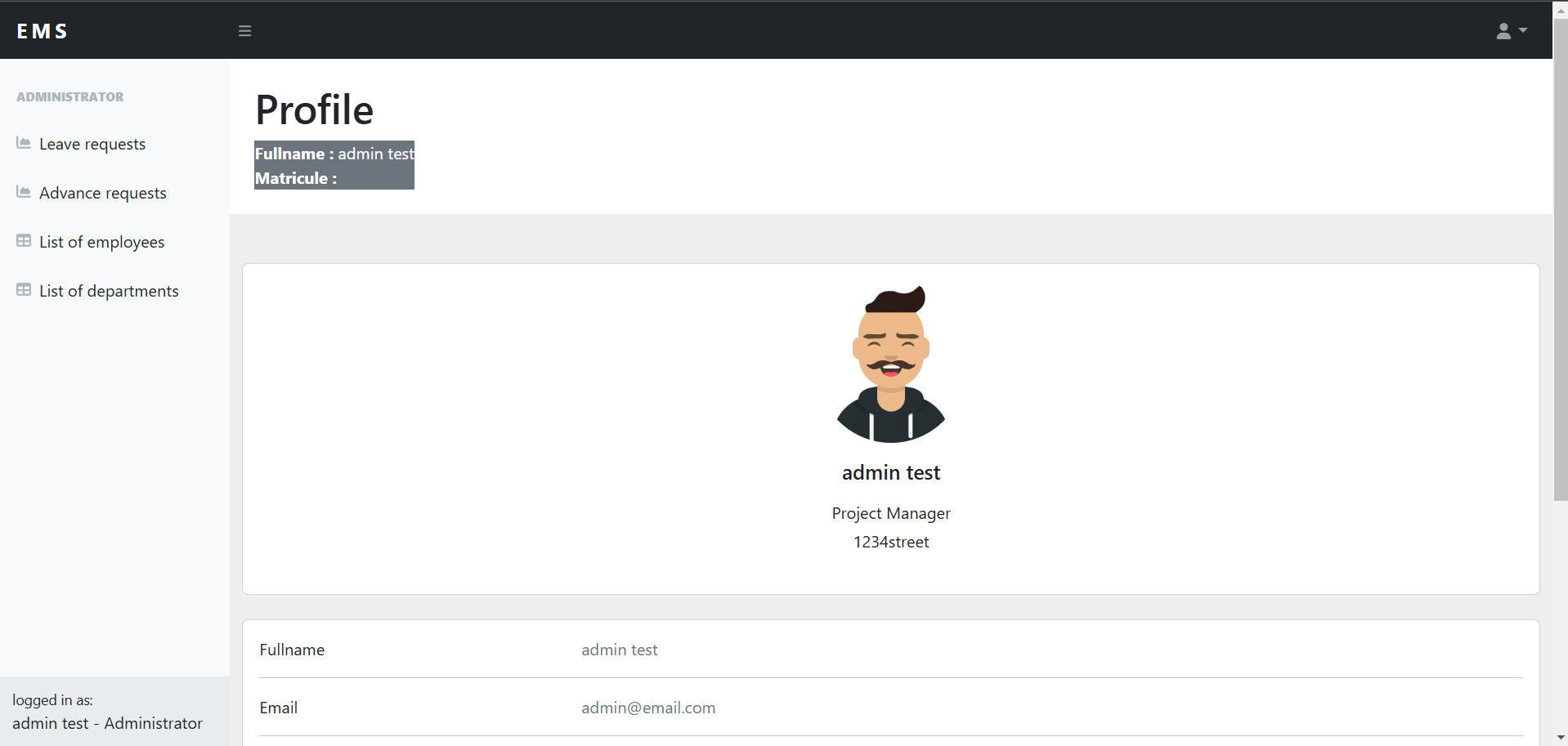
**Results and Discussions:**

**Screenshots of the Project:**

A screenshot of a login box

Description automatically generated

**Login Page**

****

**Admin Dashboard**

A screenshot of a computer

Description automatically generated

**Employee List**

A screenshot of a computer

Description automatically generated

**Advance Request Form**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Leave Management**

**Conclusion and Future Scope:**

**Conclusion:**

The Employee Management System demonstrates the potential of web technologies in simplifying HR tasks. With its modular structure and robust backend, the system is well-suited for small to medium-sized organizations.

**Future Scope:**

* **Integration of Analytics:** Add data visualization to track employee performance and HR trends.
* **Cloud Deployment:** Host the system on a cloud platform for wider accessibility.
* **Advanced Features:** Incorporate AI for predictive HR management, such as employee retention analysis.

**Acknowledgment:**

**We extend our gratitude to Professor Ms. Priya Salunke for her invaluable guidance in the subject of Programming for Problem Solving. Her insights and challenging questions significantly enhanced this work. Special thanks to Thakur College of Engineering & Technology for providing access to research resources and facilities essential for this project.**

**References**

1. **Flask Documentation**
2. **Appwrite Documentation**
3. **Bootsrap Documentation**

**Annexure (Code):**

**Authentication:**

* **Login Page**

1. <!DOCTYPE html>

2. <html lang="en">

3. <head>

4. <!-- Charset declaration to specify the character encoding for the document -->

5. <meta charset="utf-8" />

6. <!-- Ensures that the page is displayed correctly on different versions of Internet Explorer -->

7. <meta http-equiv="X-UA-Compatible" content="IE=edge" />

8. <!-- Makes the page responsive by setting the viewport properties for mobile devices -->

9. <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />

10. <!-- Description meta tag for SEO purposes, currently empty -->

11. <meta name="description" content="" />

12. <!-- Author meta tag, currently empty -->

13. <meta name="author" content="" />

14. <!-- Title of the webpage, displayed in the browser tab -->

15. <title>Login</title>

16. <!-- Link to the external CSS file for styling the page -->

17. <link href="/static/css/styles.css" rel="stylesheet" />

18. <!-- FontAwesome script to provide icons for the page (used for FontAwesome icons) -->

19. <script src="https://use.fontawesome.com/releases/v6.3.0/js/all.js" crossorigin="anonymous"></script>

20. </head>

21.

22. <body class="bg-primary"> <!-- Set the background color of the page using the 'bg-primary' class -->

23. <div id="layoutAuthentication">

24. <!-- Main content area for authentication -->

25. <div id="layoutAuthentication\_content">

26. <main>

27. <div class="container">

28. <!-- Row to center the login card -->

29. <div class="row justify-content-center">

30. <div class="col-lg-5"> <!-- Define the column size for the login card -->

31. <div class="card shadow-lg border-0 rounded-lg mt-5">

32. <!-- Header section of the card -->

33. <div class="card-header">

34. <h3 class="text-center font-weight-light my-4">Login</h3>

35. </div>

36.

37. <!-- Flask error message handling -->

38. <div class="container">

39. {% for message in get\_flashed\_messages() %}

40. <div class="alert alert-warning" role="alert">

41. <div style="text-align:center">{{ message }}</div>

42. </div>

43. {% endfor %}

44. </div>

45. <!-- Body of the card -->

46. <div class="card-body">

47. <!-- Login form that sends POST request -->

48. <form id="form" action="#" method="POST">

49. <!-- Employee ID field -->

50. <div class="form-floating mb-3">

51. <input class="form-control" name="employee\_id" id="employee\_id" type="number" placeholder="Employee Id" required/>

52. <label for="matricule">Employee Id</label>

53. </div>

54.

55. <!-- Password field -->

56. <div class="form-floating mb-3">

57. <input class="form-control" name="password" id="inputPassword" type="password" placeholder="Password" required/>

58. <label for="inputPassword">Password</label>

59. </div>

60.

61. <!-- Forgot password link and submit button -->

62. <div class="d-flex align-items-center justify-content-between mt-4 mb-0">

63. <a class="small" href="{{url\_for('auth.motdepass')}}">Forgot Password?</a>

64. <input type="submit" class="btn btn-primary" value="Login">

65. </div>

66. </form>

67. </div>

68.

69. <!-- Footer of the card with a link to the registration page -->

70. <div class="card-footer text-center py-3">

71. <div class="small"><a href="{{url\_for('auth.register')}}">Need an account? Sign up!</a></div>

72. </div>

73. </div>

74. </div>

75. </div>

76. </div>

77. </main>

78. </div>

79. </div>

80.

81. <!-- Bootstrap JavaScript for handling UI components -->

82. <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" crossorigin="anonymous"></script>

83. <!-- Custom JavaScript for additional scripts if required -->

84. <script src="/static/js/scripts.js"></script>

85. </body>

86. </html>

87.

* **Registration Page**

1. <!DOCTYPE html>

2. <html lang="en">

3. <head>

4. <!-- Charset declaration to specify the character encoding for the document -->

5. <meta charset="utf-8" />

6.

7. <!-- Ensures that the page is displayed correctly on different versions of Internet Explorer -->

8. <meta http-equiv="X-UA-Compatible" content="IE=edge" />

9.

10. <!-- Makes the page responsive by setting the viewport properties for mobile devices -->

11. <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />

12.

13. <!-- Description meta tag for SEO purposes, currently empty -->

14. <meta name="description" content="" />

15.

16. <!-- Author meta tag, currently empty -->

17. <meta name="author" content="" />

18.

19. <!-- Title of the webpage, displayed in the browser tab -->

20. <title>Login</title>

21.

22. <!-- Link to the external CSS file for styling the page -->

23. <link href="/static/css/styles.css" rel="stylesheet" />

24.

25. <!-- FontAwesome script to provide icons for the page (used for FontAwesome icons) -->

26. <script src="https://use.fontawesome.com/releases/v6.3.0/js/all.js" crossorigin="anonymous"></script>

27. </head>

28.

29. <body class="bg-primary"> <!-- Set the background color of the page using the 'bg-primary' class -->

30. <div id="layoutAuthentication">

31. <!-- Main content area for authentication -->

32. <div id="layoutAuthentication\_content">

33. <main>

34. <div class="container">

35. <!-- Row to center the login card -->

36. <div class="row justify-content-center">

37. <div class="col-lg-5"> <!-- Define the column size for the login card -->

38. <div class="card shadow-lg border-0 rounded-lg mt-5">

39. <!-- Header section of the card -->

40. <div class="card-header">

41. <h3 class="text-center font-weight-light my-4">Login</h3>

42. </div>

43.

44. <!-- Flask error message handling -->

45. <div class="container">

46. {% for message in get\_flashed\_messages() %}

47. <div class="alert alert-warning" role="alert">

48. <!-- Error message is displayed here from Flask's flash system -->

49. <div style="text-align:center">{{ message }}</div>

50. </div>

51. {% endfor %}

52. </div>

53.

54. <!-- Body of the card with the form for user login -->

55. <div class="card-body">

56. <!-- Login form that sends POST request to the server for authentication -->

57. <form id="form" action="#" method="POST">

58.

59. <!-- Employee ID input field -->

60. <div class="form-floating mb-3">

61. <input class="form-control" name="employee\_id" id="employee\_id" type="number" placeholder="Employee Id" required/>

62. <label for="employee\_id">Employee Id</label>

63. </div>

64.

65. <!-- Password input field -->

66. <div class="form-floating mb-3">

67. <input class="form-control" name="password" id="inputPassword" type="password" placeholder="Password" required/>

68. <label for="inputPassword">Password</label>

69. </div>

70.

71. <!-- Forgot password link and login button -->

72. <div class="d-flex align-items-center justify-content-between mt-4 mb-0">

73. <a class="small" href="{{url\_for('auth.motdepass')}}">Forgot Password?</a>

74. <input type="submit" class="btn btn-primary" value="Login">

75. </div>

76. </form>

77. </div>

78.

79. <!-- Footer of the card with a link to the registration page -->

80. <div class="card-footer text-center py-3">

81. <div class="small"><a href="{{url\_for('auth.register')}}">Need an account? Sign up!</a></div>

82. </div>

83. </div>

84. </div>

85. </div>

86. </div>

87. </main>

88. </div>

89. </div>

90.

91. <!-- Bootstrap JavaScript for handling UI components -->

92. <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" crossorigin="anonymous"></script>

93.

94. <!-- Custom JavaScript for additional functionality or scripts -->

95. <script src="/static/js/scripts.js"></script>

96. </body>

97. </html>

98.

* **Forget Password**

1. <!DOCTYPE html>

2. <html lang="en">

3. <head>

4. <!-- Charset declaration to specify the character encoding for the document -->

5. <meta charset="utf-8" />

6.

7. <!-- Ensures that the page is displayed correctly on different versions of Internet Explorer -->

8. <meta http-equiv="X-UA-Compatible" content="IE=edge" />

9.

10. <!-- Makes the page responsive by setting the viewport properties for mobile devices -->

11. <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />

12.

13. <!-- Description meta tag for SEO purposes, currently empty -->

14. <meta name="description" content="" />

15.

16. <!-- Author meta tag, currently empty -->

17. <meta name="author" content="" />

18.

19. <!-- Title of the webpage, displayed in the browser tab -->

20. <title>Forgot your password</title>

21.

22. <!-- Link to the external CSS file for styling the page -->

23. <link href="/static/css/styles.css" rel="stylesheet" />

24.

25. <!-- FontAwesome script to provide icons for the page (used for FontAwesome icons) -->

26. <script src="https://use.fontawesome.com/releases/v6.3.0/js/all.js" crossorigin="anonymous"></script>

27. </head>

28.

29. <body class="bg-primary"> <!-- Set the background color of the page using the 'bg-primary' class -->

30. <div id="layoutAuthentication">

31. <!-- Main content area for authentication -->

32. <div id="layoutAuthentication\_content">

33. <main>

34. <div class="container">

35. <!-- Row to center the password recovery card -->

36. <div class="row justify-content-center">

37. <div class="col-lg-5"> <!-- Define the column size for the recovery card -->

38. <div class="card shadow-lg border-0 rounded-lg mt-5">

39. <!-- Header section of the card -->

40. <div class="card-header">

41. <h3 class="text-center font-weight-light my-4">Password Recovery</h3>

42. </div>

43.

44. <!-- Body of the card with the password recovery form -->

45. <div class="card-body">

46. <div class="small mb-3 text-muted"></div>

47. <!-- Password recovery form -->

48. <form id="form" action="#" method="POST">

49. <!-- Matricule input field -->

50. <div class="form-floating mb-3">

51. <input class="form-control" id="matricule" type="number" placeholder="000000" />

52. <label for="matricule">Matricule</label>

53. </div>

54.

55. <!-- New password input field -->

56. <div class="form-floating mb-3">

57. <input class="form-control" id="nvpass" type="password" />

58. <label for="nvpass">New Password</label>

59. </div>

60.

61. <!-- Reset password button -->

62. <div class="d-flex align-items-center justify-content-between mt-4 mb-0">

63. <button type="submit" class="btn btn-primary pmd-ripple-effect pmd-btn-raised" name="resetpass" value="C Leave">Reset password</button>

64. </div>

65. </form>

66. </div>

67. </div>

68. </div>

69. </div>

70. </div>

71. </main>

72. </div>

73. </div>

74.

75. <!-- Bootstrap JavaScript for handling UI components like modals, tooltips, etc. -->

76. <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" crossorigin="anonymous"></script>

77.

78. <!-- Custom JavaScript for additional functionality or scripts -->

79. <script src="/static/js/scripts.js"></script>

80. </body>

81. </html>

82.

**Home Page:**

1. <!DOCTYPE html>

2. <html lang="en">

3. <head>

4. <!-- Charset declaration to specify the character encoding for the document -->

5. <meta charset="utf-8" />

6.

7. <!-- Ensures that the page is displayed correctly on different versions of Internet Explorer -->

8. <meta http-equiv="X-UA-Compatible" content="IE=edge" />

9.

10. <!-- Makes the page responsive by setting the viewport properties for mobile devices -->

11. <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />

12.

13. <!-- Description meta tag for SEO purposes, currently empty -->

14. <meta name="description" content="" />

15.

16. <!-- Author meta tag, currently empty -->

17. <meta name="author" content="" />

18.

19. <!-- Title of the webpage, displayed in the browser tab -->

20. <title>Forgot your password</title>

21.

22. <!-- Link to the external CSS file for styling the page -->

23. <link href="/static/css/styles.css" rel="stylesheet" />

24.

25. <!-- FontAwesome script to provide icons for the page (used for FontAwesome icons) -->

26. <script src="https://use.fontawesome.com/releases/v6.3.0/js/all.js" crossorigin="anonymous"></script>

27. </head>

28.

29. <body class="bg-primary"> <!-- Set the background color of the page using the 'bg-primary' class -->

30. <div id="layoutAuthentication">

31. <!-- Main content area for authentication -->

32. <div id="layoutAuthentication\_content">

33. <main>

34. <div class="container">

35. <!-- Row to center the password recovery card -->

36. <div class="row justify-content-center">

37. <div class="col-lg-5"> <!-- Define the column size for the recovery card -->

38. <div class="card shadow-lg border-0 rounded-lg mt-5">

39. <!-- Header section of the card -->

40. <div class="card-header">

41. <h3 class="text-center font-weight-light my-4">Password Recovery</h3>

42. </div>

43.

44. <!-- Body of the card with the password recovery form -->

45. <div class="card-body">

46. <div class="small mb-3 text-muted"></div>

47. <!-- Password recovery form -->

48. <form id="form" action="#" method="POST">

49. <!-- Matricule input field -->

50. <div class="form-floating mb-3">

51. <input class="form-control" id="matricule" type="number" placeholder="000000" />

52. <label for="matricule">Matricule</label>

53. </div>

54.

55. <!-- New password input field -->

56. <div class="form-floating mb-3">

57. <input class="form-control" id="nvpass" type="password" />

58. <label for="nvpass">New Password</label>

59. </div>

60.

61. <!-- Reset password button -->

62. <div class="d-flex align-items-center justify-content-between mt-4 mb-0">

63. <button type="submit" class="btn btn-primary pmd-ripple-effect pmd-btn-raised" name="resetpass" value="C Leave">Reset password</button>

64. </div>

65. </form>

66. </div>

67. </div>

68. </div>

69. </div>

70. </div>

71. </main>

72. </div>

73. </div>

74.

75. <!-- Bootstrap JavaScript for handling UI components like modals, tooltips, etc. -->

76. <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" crossorigin="anonymous"></script>

77.

78. <!-- Custom JavaScript for additional functionality or scripts -->

79. <script src="/static/js/scripts.js"></script>

80. </body>

81. </html>

82.

**User Functions**

* **Follow-Up Page**

1. {% extends "base.html" %} <!-- Inherit from the base template to maintain consistent layout and structure across the application -->

2.

3. {% block title %} <!-- Set the page title -->

4. Request follow-up <!-- Title displayed on the browser tab -->

5. {% endblock title %}

6.

7. {% block content %} <!-- Main content block to render the page content -->

8. <div id="layoutSidenav\_content">

9. <main>

10. <div class="container-fluid px-4">

11. <!-- Section header for leave requests -->

12. <h1 class="mt-4">Your leave requests</h1><hr>

13. <div class="card mb-4">

14. <div class="card-header">

15. <i class="fas fa-table me-1"></i>

16. Leave requests list

17. </div>

18. <div class="card-body">

19. <table class="table table-striped">

20. <thead>

21. <tr>

22. <th scope="col">Employee Id</th>

23. <th scope="col">Name</th>

24. <th scope="col">Start date</th>

25. <th scope="col">End date</th>

26. <th scope="col">Leave type</th>

27. <th scope="col">Reason</th>

28. <th scope="col">Status</th>

29. <th scope="col"></th>

30. </tr>

31. </thead>

32. <tbody>

33. <!-- Loop through leave requests and display each row -->

34. <tr>

35. <th scope="row">{{ leave['employee\_id'] }}</th> <!-- Display employee ID -->

36. <td>{{ leave['nom'] }}</td> <!-- Display employee name -->

37. <td>{{ leave['date\_deb'] }}</td> <!-- Display start date -->

38. <td>{{ leave['date\_fin'] }}</td> <!-- Display end date -->

39. <td>{{ leave['type'] }}</td> <!-- Display leave type -->

40. <td>{{ leave['motif'] }}</td> <!-- Display reason for leave -->

41.

42. <!-- Display status with conditional styling based on its value -->

43. {% if leave['status'] == 'Accepted' %}

44. <td class="text-success">{{ leave['status'] }}</td> <!-- Display 'Accepted' status in green -->

45. {% elif leave['status'] == 'Declined' %}

46. <td class="text-danger">{{ leave['status'] }}</td> <!-- Display 'Declined' status in red -->

47. {% else %}

48. <td class=".text-primary">{{ leave['status'] }}</td> <!-- Display other statuses in blue -->

49. {% endif %}

50. </tr>

51. </tbody>

52. </table>

53. </div>

54. </div>

55.

56. <!-- Section header for salary advance requests -->

57. <h1 class="mt-4">Your salary advance requests</h1><hr>

58.

59. <div class="card mb-4">

60. <div class="card-header">

61. <i class="fas fa-table me-1"></i>

62. Salary Advance requests list

63. </div>

64. <div class="card-body">

65. <table class="table table-striped">

66. <thead>

67. <tr>

68. <th scope="col">Employee Id</th>

69. <th scope="col">Name</th>

70. <th scope="col">Requested amount</th>

71. <th scope="col">Reason</th>

72. <th scope="col">Status</th>

73. <th scope="col"></th>

74. </tr>

75. </thead>

76. <tbody>

77. <!-- Loop through salary advance requests and display each row -->

78. <tr>

79. <th scope="row">{{ advance['employee\_id'] }}</th> <!-- Display employee ID -->

80. <td>{{ advance['nom'] }}</td> <!-- Display employee name -->

81. <td>{{ advance['montant'] }}</td> <!-- Display requested amount -->

82. <td>{{ advance['motif'] }}</td> <!-- Display reason for salary advance -->

83.

84. <!-- Display status with conditional styling based on its value -->

85. {% if advance['status'] == 'Accepted' %}

86. <td class="text-success">{{ advance['status'] }}</td> <!-- Display 'Accepted' status in green -->

87. {% elif advance['status'] == 'Declined' %}

88. <td class="text-danger">{{ advance['status'] }}</td> <!-- Display 'Declined' status in red -->

89. {% else %}

90. <td class=".text-primary">{{ advance['status'] }}</td> <!-- Display other statuses in blue -->

91. {% endif %}

92. </tr>

93. </tbody>

94. </table>

95. </div>

96. </div>

97. </div>

98. </main>

99. {% endblock content %}

100.

* **Request Advance Salary Page**

1. {% extends "base.html" %} <!-- Inherit from the base template to maintain consistent layout and structure across the application -->

2.

3. {% block title %} <!-- Set the page title -->

4. Request for a salary advance <!-- Title displayed on the browser tab -->

5. {% endblock title %}

6.

7. {% block content %} <!-- Main content block where the page content is rendered -->

8. <div id="layoutSidenav\_content">

9. <div class="container">

10.

11. <!-- Flash messages for displaying errors or success messages passed from Flask -->

12. <div class="container">

13. {% for message in get\_flashed\_messages() %}

14. <div class="alert alert-warning" role="alert">

15. <!--<button type="button" class="close" data-dismiss="alert">×</button>-->

16. <div style="text-align:center">{{ message }}</div> <!-- Display the flash message in a centered box -->

17. </div>

18. {% endfor %}

19. </div>

20.

21. <!-- Section header for the salary advance request form -->

22. <h1 class="mt-4">Request for a salary advance</h1><hr>

23.

24. <div class="card pmd-card single-col-form">

25. <form id="apply-leave" method="post" action="">

26. <div class="card-body">

27. <div class="row">

28.

29. <!-- Input field for the requested amount -->

30. <div class="col-md-6">

31. <div class="form-group pmd-textfield pmd-textfield-floating-label">

32. <label class="control-label" for="montant">Amount requested</label>

33. <input type="number" class="form-control" id="requested\_amount" name="requested\_amount" required>

34. <span class="pmd-textfield-focused"></span> <!-- Focused text field effect -->

35. </div>

36. </div>

37.

38. <!-- Text area for the reason for the salary advance request -->

39. <div class="col-12">

40. <div class="form-group pmd-textfield pmd-textfield-floating-label">

41. <label>Reason</label>

42. <textarea class="form-control" id="reason" name="reason" required></textarea>

43. <span class="pmd-textfield-focused"></span> <!-- Focused text area effect -->

44. </div>

45. </div>

46. </div>

47. </div>

48.

49. <!-- Footer section with the submit button to apply for the salary advance -->

50. <div class="card-footer">

51. <button type="submit" class="btn btn-primary pmd-ripple-effect pmd-btn-raised" name="applyleave" value="Apply Leave">

52. Apply

53. </button>

54. </div>

55. </form>

56. </div>

57. </div>

58. </div>

59. {% endblock content %}

60.

* **Request Leave Page**

1. {% extends "base.html" %} <!-- Extending the base template for consistent layout across the site -->

2.

3. {% block title %} <!-- Setting the title of the page, visible in the browser tab -->

4. Request leave <!-- Page title -->

5. {% endblock title %}

6.

7. {% block content %} <!-- Main content block where the page content will be rendered -->

8.

9. <!-- Flash messages (error or success) passed from the Flask application -->

10. <div id="layoutSidenav\_content">

11.

12. <div class="container">

13.

14. <!-- Display any flashed messages (like error or success notifications) from Flask -->

15. <div class="container">

16. {% for message in get\_flashed\_messages() %}

17. <div class="alert alert-warning" role="alert">

18. <!--<button type="button" class="close" data-dismiss="alert">×</button>-->

19. <div style="text-align:center">{{ message }}</div> <!-- Display the message in the center -->

20. </div>

21. {% endfor %}

22. </div>

23.

24. <!-- Page header -->

25. <h1 class="mt-4">Request leave</h1><hr>

26.

27. <!-- Apply Leave Form -->

28. <div class="card pmd-card single-col-form">

29. <form id="apply-leave" method="post" action="">

30. <div class="card-body">

31. <div class="row">

32.

33. <!-- Leave type selection dropdown -->

34. <div class="col-12">

35. <div class="form-group pmd-textfield pmd-textfield-floating-label">

36. <label for="leave-type">Leave type</label>

37. <select name="leave-type" id="leave-type" class="form-control" title="Please select a Leave Type" required>

38. <option></option> <!-- Default empty option -->

39. <option>Sick leave</option> <!-- Option for sick leave -->

40. <option>Casual leave</option> <!-- Option for casual leave -->

41. <option>Unpaid leave</option> <!-- Option for unpaid leave -->

42. <option>Hourly leave</option> <!-- Option for hourly leave -->

43. </select>

44. <span class="pmd-textfield-focused"></span> <!-- Floating label effect -->

45. </div>

46. </div>

47.

48. <!-- Start date input -->

49. <div class="col-md-6">

50. <div class="form-group pmd-textfield pmd-textfield-floating-label">

51. <label class="control-label" for="datepickerstart">Start Date</label>

52. <input type="text" class="form-control" name="date\_start" placeholder="dd/mm/yy" required>

53. <span class="pmd-textfield-focused"></span> <!-- Floating label effect -->

54. </div>

55. </div>

56.

57. <!-- End date input -->

58. <div class="col-md-6">

59. <div class="form-group pmd-textfield pmd-textfield-floating-label">

60. <label class="control-label" for="datepickerend">End Date</label>

61. <input type="text" class="form-control" name="date\_end" placeholder="dd/mm/yy" required>

62. <span class="pmd-textfield-focused"></span> <!-- Floating label effect -->

63. </div>

64. </div>

65.

66. <!-- Reason text area -->

67. <div class="col-12">

68. <div class="form-group pmd-textfield pmd-textfield-floating-label">

69. <label>Reason</label>

70. <textarea class="form-control" id="reason" name="reason" required></textarea>

71. <span class="pmd-textfield-focused"></span> <!-- Floating label effect -->

72. </div>

73. </div>

74.

75. </div>

76. </div>

77.

78. <!-- Footer section with submit button -->

79. <div class="card-footer">

80. <button type="submit" class="btn btn-primary pmd-ripple-effect pmd-btn-raised" name="applyleave" value="Apply Leave">Apply Request</button>

81. </div>

82. </form>

83. </div>

84. </div>

85. </div>

86. </div>

87. </div>

88.

89. {% endblock content %}

90.

**Administration Control:**

* **Edit Employee Details Page**

1. {% extends "base.html" %} <!-- Extending the base template for consistent layout across the site -->

2.

3. {% block title %} <!-- Setting the title of the page, visible in the browser tab -->

4. Dashboard <!-- Page title -->

5. {% endblock title %}

6.

7. {% block content %} <!-- Main content block where the page content will be rendered -->

8.

9. <style>

10. .d-flex {

11. gap:20px; /\* Adding gap between elements in flex container \*/

12. }

13. </style>

14.

15. <div id="layoutSidenav\_content">

16.

17. <!-- Container for the page content -->

18. <div class="container-fluid px-4">

19. <h1 class="mt-4">Edit employee's information</h1> <!-- Page header -->

20. </div>

21. <hr>

22.

23. <!-- Flash messages (error or success) passed from the Flask application -->

24. <div class="container">

25. {% for message in get\_flashed\_messages() %}

26. <div class="alert alert-warning" role="alert">

27. <!--<button type="button" class="close" data-dismiss="alert">×</button>-->

28. <div style="text-align:center">{{ message }}</div> <!-- Display the message in the center -->

29. </div>

30. {% endfor %}

31. </div>

32.

33. <!-- Section to display employee's information and edit form -->

34. <section style="background-color: #eee;">

35. <div class="container py-5">

36. <div class="row">

37. <div class="col-lg">

38. <div class="card mb-4">

39. <!-- Employee profile card -->

40. <div class="card-body text-center">

41. <img src="https://mdbcdn.b-cdn.net/img/Photos/new-templates/bootstrap-chat/ava3.webp" alt="avatar"

42. class="rounded-circle img-fluid" style="width: 150px;"> <!-- Employee avatar image -->

43. <h5 class="my-3">{{ fullname }}</h5> <!-- Full name of the employee -->

44. <p class="text-muted mb-1">{{ dep }}</p> <!-- Department of the employee -->

45. <p class="text-muted mb-4">{{ address }}</p> <!-- Address of the employee -->

46. <div class="d-flex justify-content-center mb-2">

47. <!-- Contact button linking to email -->

48. <a href="mailto: {{ email }}" target="\_blank" class="btn btn-primary">Contact</a>

49. </div>

50. </div>

51. </div>

52.

53. <!-- Edit employee information form -->

54. <form action="#" method="POST">

55. <div class="col-lg">

56. <div class="card mb-4">

57. <div class="card-body">

58. <!-- First Name input -->

59. <div class="row">

60. <div class="col-sm-3">

61. <p class="mb-0">First Name</p>

62. </div>

63. <div class="col-sm-9">

64. <input type="text" value="{{ firstname }}" name="firstname"> <!-- Pre-filled value for first name -->

65. </div>

66. </div>

67. <hr>

68. <!-- Last Name input -->

69. <div class="row">

70. <div class="col-sm-3">

71. <p class="mb-0">Last Name</p>

72. </div>

73. <div class="col-sm-9">

74. <input type="text" value="{{ lastname }}" name="lastname"> <!-- Pre-filled value for last name -->

75. </div>

76. </div>

77. <hr>

78. <!-- Gender input -->

79. <div class="row">

80. <div class="col-sm-3">

81. <p class="mb-0">Gender</p>

82. </div>

83. <div class="col-sm-9">

84. <input type="text" value="{{ gender }}" name="gender"/> <!-- Pre-filled value for gender -->

85. </div>

86. </div>

87. <hr>

88. <!-- Date of Birth input -->

89. <div class="row">

90. <div class="col-sm-3">

91. <p class="mb-0">Date of Birth</p>

92. </div>

93. <div class="col-sm-9">

94. <input type="date" value="{{ date\_of\_birth }}" name="date\_of\_birth"/> <!-- Pre-filled value for date of birth -->

95. </div>

96. </div>

97. <hr>

98. <!-- Email input -->

99. <div class="row">

100. <div class="col-sm-3">

101. <p class="mb-0">Email</p>

102. </div>

103. <div class="col-sm-9">

104. <input type="text" value="{{ email }}" name="email"/> <!-- Pre-filled value for email -->

105. </div>

106. </div>

107. <hr>

108. <!-- Phone Number input -->

109. <div class="row">

110. <div class="col-sm-3">

111. <p class="mb-0">Phone Number</p>

112. </div>

113. <div class="col-sm-9">

114. <input type="text" value="{{ phone }}" name="phone"/> <!-- Pre-filled value for phone -->

115. </div>

116. </div>

117. <hr>

118. <!-- Department input -->

119. <div class="row">

120. <div class="col-sm-3">

121. <p class="mb-0">Department</p>

122. </div>

123. <div class="col-sm-9">

124. <input type="text" value="{{ dep }}" name="dep"/> <!-- Pre-filled value for department -->

125. </div>

126. </div>

127. <hr>

128. <!-- Address input -->

129. <div class="row">

130. <div class="col-sm-3">

131. <p class="mb-0">Address</p>

132. </div>

133. <div class="col-sm-9">

134. <input type="text" value="{{ address }}" name="address"/> <!-- Pre-filled value for address -->

135. </div>

136. </div>

137. <hr>

138. <!-- Salary input -->

139. <div class="row">

140. <div class="col-sm-3">

141. <p class="mb-0">Salary</p>

142. </div>

143. <div class="col-sm-9">

144. <input type="text" value="{{ salary }}" name="salary"/> <!-- Pre-filled value for salary -->

145. </div>

146. </div>

147. </div>

148. </div>

149. </div>

150.

151. <!-- Submit and Delete buttons -->

152. <div class="d-flex justify-content-center">

153. <input type="submit" name="submit\_b" value="Apply" class="btn btn-primary" /> <!-- Button to apply changes -->

154. <input type="submit" name="submit\_b" value="Delete" class="btn btn-danger" /> <!-- Button to delete employee -->

155. </div>

156. </form>

157. </div>

158. </div>

159. </div>

160. </section>

161. </div>

162.

163. {% endblock content %}

164.

* **Employee List Page**

1. {% extends "base.html" %} <!-- Extending the base template for consistent layout across the site -->

2.

3. {% block title %} <!-- Setting the title of the page -->

4. List of employees <!-- Title that will appear on the browser tab -->

5. {% endblock title %}

6.

7. {% block content %} <!-- Main content block where the page content will be rendered -->

8.

9. <div id="layoutSidenav\_content">

10. <main>

11. <div class="container-fluid px-4">

12. <!-- Page header for the employee list -->

13. <h1 class="mt-4">List of employees</h1>

14. <hr>

15.

16. <!-- Flash messages (error or success) passed from Flask -->

17. <div class="container">

18. {% for message in get\_flashed\_messages() %}

19. <div class="alert alert-warning" role="alert">

20. <!--<button type="button" class="close" data-dismiss="alert">×</button>-->

21. <div style="text-align:center">{{ message }}</div> <!-- Display the message in the center -->

22. </div>

23. {% endfor %}

24. </div>

25.

26. <!-- Card component for displaying the employee list -->

27. <div class="card mb-4">

28. <div class="card-header">

29. <i class="fas fa-table me-1"></i> <!-- Icon for the table -->

30. List of employees <!-- Title inside the card -->

31. </div>

32. <div class="card-body">

33. <!-- Employee data table -->

34. <table class="table table-striped">

35. <thead>

36. <tr>

37. <th scope="col">Employee Id</th> <!-- Employee ID column -->

38. <th scope="col">First name</th> <!-- First name column -->

39. <th scope="col">Name</th> <!-- Last name column -->

40. <th scope="col">Department</th> <!-- Department column -->

41. <th scope="col">Salary</th> <!-- Salary column -->

42. <th scope="col">Action</th> <!-- Action column for edit button -->

43. </tr>

44. </thead>

45. <tbody>

46. {% for user in list\_users %}

47. <form action="#" method="POST"> <!-- Form for submitting edit action -->

48. <tr>

49. <!-- Employee ID is hidden in the form and displayed -->

50. <td><input type="hidden" name="employee\_id" value="{{ user['employee\_id'] }}"/>{{ user['employee\_id'] }}</td>

51. <!-- Displaying user details -->

52. <td>{{ user['firstname'] }}</td>

53. <td>{{ user['lastname'] }}</td>

54. <td>{{ user['dep'] }}</td>

55. <td>{{ user['salary'] }}</td>

56. <!-- Edit button to trigger the form submission -->

57. <td><input name="Edit" type="submit" value="Edit" class="btn btn-primary" style="width:100%"/></td>

58. </tr>

59. </form>

60. {% endfor %}

61. </tbody>

62. </table>

63. </div>

64. </div>

65. </div>

66. </main>

67.

68. {% endblock content %}

69.

* **Employee Advance Salary Management Page**

1. {% extends "base.html" %} <!-- Extending the base template to maintain a consistent layout -->

2.

3. {% block title %} <!-- Title of the page, displayed in the browser's tab -->

4. Salary advance requests

5. {% endblock title %}

6.

7. {% block content %} <!-- Main content block for the page -->

8.

9. <div id="layoutSidenav\_content">

10. <main>

11. <div class="container-fluid px-4">

12. <!-- Display any flashed messages from Flask (e.g., success or error notifications) -->

13. <div class="container">

14. {% for message in get\_flashed\_messages() %}

15. <div class="alert alert-warning" role="alert">

16. <!--<button type="button" class="close" data-dismiss="alert">×</button>-->

17. <div style="text-align:center">{{ message }}</div> <!-- Centered message display -->

18. </div>

19. {% endfor %}

20. </div>

21.

22. <!-- Page header for Salary Advance Requests -->

23. <h1 class="mt-4">Salary advance requests</h1><hr>

24.

25. <!-- Card to display the Salary Advance Requests table -->

26. <div class="card mb-4">

27. <div class="card-header">

28. <i class="fas fa-table me-1"></i> <!-- Table icon -->

29. Salary Advance requests list <!-- Header for the card -->

30. </div>

31. <div class="card-body">

32. <!-- Table to display the salary advance requests -->

33. <table class="table table-striped">

34. <thead>

35. <tr>

36. <th scope="col">Employee Id</th> <!-- Column for Employee ID -->

37. <th scope="col">Name</th> <!-- Column for Employee Name -->

38. <th scope="col">Requested amount</th> <!-- Column for the requested amount -->

39. <th scope="col">Reason</th> <!-- Column for the reason for the salary advance -->

40. <th scope="col"></th> <!-- Column for the Accept/Decline actions -->

41. </tr>

42. </thead>

43. <tbody>

44. <!-- Loop through the list of advances and create a row for each -->

45. {% for advance in advances\_list %}

46. <form action="#" method="POST"> <!-- Form to handle the submit actions -->

47. <tr>

48. <!-- Employee ID is hidden but displayed in the table -->

49. <th scope="row"><input type="hidden" name="employee\_id" value="{{ advance['employee\_id'] }}"/>{{ advance['employee\_id'] }}</th>

50. <!-- Display the employee's name, requested amount, and reason -->

51. <td>{{ advance['firstname'] }}</td>

52. <td>{{ advance['requested\_amount'] }}</td>

53. <td>{{ advance['reason'] }}</td>

54. <!-- Buttons to accept or decline the request -->

55. <td><input type="submit" name="submit\_b" value="Accept" class="btn btn-success" style="width:100%"/></td>

56. <td><input type="submit" name="submit\_b" value="Decline" class="btn btn-danger" style="width:100%"/></td>

57. </tr>

58. </form>

59. {% endfor %}

60. </tbody>

61. </table>

62. </div>

63. </div>

64. </div>

65. </main>

66.

67. {% endblock content %}

68.

* **Employee Leave Salary Request Management Page**

1. {% extends "base.html" %} <!-- Extending the base template for consistent layout -->

2.

3. {% block title %} <!-- Title of the page -->

4. Leave requests

5. {% endblock title %}

6.

7. {% block content %} <!-- Main content block -->

8.

9. <div id="layoutSidenav\_content">

10. <main>

11. <div class="container-fluid px-4">

12. <!-- Display any flashed messages from Flask (success or error) -->

13. <div class="container">

14. {% for message in get\_flashed\_messages() %}

15. <div class="alert alert-warning" role="alert">

16. <!--<button type="button" class="close" data-dismiss="alert">×</button>-->

17. <div style="text-align:center">{{ message }}</div> <!-- Centered message display -->

18. </div>

19. {% endfor %}

20. </div>

21.

22. <!-- Page header for Leave Requests -->

23. <h1 class="mt-4">Leave requests</h1><hr>

24.

25. <!-- Card to display the Leave Requests table -->

26. <div class="card mb-4">

27. <div class="card-header">

28. <i class="fas fa-table me-1"></i> <!-- Table icon -->

29. Leave requests list <!-- Header for the card -->

30. </div>

31. <div class="card-body">

32. <!-- Table to display the leave requests -->

33. <table class="table table-striped">

34. <thead>

35. <tr>

36. <th scope="col">Employee Id</th> <!-- Column for Employee ID -->

37. <th scope="col">Name</th> <!-- Column for Employee Name -->

38. <th scope="col">Start date</th> <!-- Column for the start date of leave -->

39. <th scope="col">End date</th> <!-- Column for the end date of leave -->

40. <th scope="col">Leave type</th> <!-- Column for type of leave -->

41. <th scope="col">Reason</th> <!-- Column for the reason for leave -->

42. <th scope="col"></th> <!-- Column for Accept/Decline actions -->

43. </tr>

44. </thead>

45. <tbody>

46. <!-- Loop through the list of leave requests and create a row for each -->

47. {% for leave in leave\_list %}

48. <form action="#" method="POST"> <!-- Form to handle the submit actions -->

49. <tr>

50. <!-- Employee ID is hidden but displayed in the table -->

51. <th scope="row"><input type="hidden" name="employee\_id" value="{{ leave['employee\_id'] }}"/>{{ leave['employee\_id'] }}</th>

52. <!-- Display the employee's name, leave dates, type, and reason -->

53. <td>{{ leave['firstname'] }}</td>

54. <td>{{ leave['start\_date'] }}</td>

55. <td>{{ leave['end\_date'] }}</td>

56. <td>{{ leave['type'] }}</td>

57. <td>{{ leave['reason'] }}</td>

58. <!-- Buttons to accept or decline the leave request -->

59. <td><input type="submit" name="submit\_b" value="Accept" class="btn btn-success" style="width:100%"/></td>

60. <td><input type="submit" name="submit\_b" value="Decline" class="btn btn-danger" style="width:100%"/></td>

61. </tr>

62. </form>

63. {% endfor %}

64. </tbody>

65. </table>

66. </div>

67. </div>

68. </div>

69. </main>

70.

71. {% endblock content %}

72.

**Page Styling :**

* **Global CSS (Bootstrap & Tailwind CSS): It is Made up of bootstrap and Tailwind CSS import classes.**
* **They are mainly Prebuilt libraries of 100s of lines of code.**
* **Displaying it in a limited sized report is not possible.**

**Page Logic , Function & Database :**

* **For Logic and Function:**

1. We implemented in line logic operators which is a property of TypeScript Similar to JavaScript
2. It can be used after making a “config.ts” file and configuring the inline typescript for the project.

* **Database File (Appwrite Database):**

1. It is a no SQL database system based in browser.
2. It can be setup using a GUI and it is based up on the concept of routing and APIs
3. We used Python’s Flask framework to implement these API’s cause it is just a copy and paste code from the official documentation of the Flask and Appwrite database.