Employee Management System Documentation

Generated on: 2025-04-05 17:28:57

Table of Contents

- templates/base.html
- templates/dashboard.html
- templates/employee_detail.html
- templates/employee_form.html
- templates/employee_list.html
- templates/errors/404.html
- templates/errors/500.html
- templates/errors/base_error.html
- templates/feedback_form.html
- templates/feedback_list.html
- templates/import_export.html
- templates/login.html
- templates/profile.html
- ./static/js/datatables-config.js
- ./static/js/main.js
- static/js/datatables-config.js
- static/js/main.js

- ./app.py
- ./auth.py
- ./create_functionality_docs.py
- ./docs.py
- ./employee.py
- ./feedback.py
- ./generate_complete_documentation.py
- ./main.py
- ./models.py
- ./simple_documentation_generator.py
- ./utils.py

File Listing

./static/css/custom.css

./static/css/custom.css

```
/* Custom styles for Employee Management System */ /* Global Styles */ :root { --sidebar-width:
250px; } /* Sidebar styles */ .sidebar { width: var(--sidebar-width); height: 100vh; position:
fixed; left: 0; top: 0; z-index: 100; transition: all 0.3s; overflow-y: auto; } .sidebar-toggled
.sidebar { width: 90px; } .sidebar-toggled .sidebar .nav-item span { display: none; } .sidebar
.nav-item .nav-link { padding: 0.75rem 1rem; width: 100%; } .sidebar .nav-item .nav-link i {
margin-right: 0.5rem; } .sidebar-divider { margin: 1rem 0; border-top: 1px solid rgba(255, 255,
255, 0.15); } /* Main content wrapper */ .content-wrapper { margin-left: var(--sidebar-width);
padding: 2rem; min-height: 100vh; transition: all 0.3s; } .sidebar-toggled .content-wrapper {
margin-left: 90px; } /* Dashboard cards */ .dashboard-card { border-radius: 0.5rem; border-left:
0.25rem solid var(--bs-primary); box-shadow: 0 0.15rem 1.75rem 0 rgba(58, 59, 69, 0.15);
transition: transform 0.2s; } .dashboard-card:hover { transform: translateY(-5px); }
.dashboard-card-header { font-weight: 700; text-transform: uppercase; } .dashboard-card-body {
font-size: 2.5rem; font-weight: 700; } /* Star rating system */ .rating-container { display: flex;
align-items: center; margin-bottom: 1rem; } .rating-stars { display: inline-flex; font-size:
1.5rem; cursor: pointer; } .rating-star { color: var(--bs-secondary); padding: 0 0.25rem;
transition: color 0.2s; } .rating-star:hover, .rating-star.rated { color: var(--bs-warning); } /*
Feedback card styles */ .feedback-card { border-radius: 0.5rem; box-shadow: 0 0.15rem 1.75rem 0
rgba(58, 59, 69, 0.15); margin-bottom: 1.5rem; } .feedback-card .card-header { display: flex;
justify-content: space-between; align-items: center; } .feedback-rating { display: inline-flex;
font-size: 1.2rem; } .feedback-star { color: var(--bs-warning); padding: 0 0.1rem; }
.feedback-quarter { font-weight: 700; color: var(--bs-primary); } .feedback-date { font-size:
0.875rem; color: var(--bs-secondary); } /* Employee detail page */ .employee-profile { padding:
2rem; background-color: var(--bs-light); border-radius: 0.5rem; box-shadow: 0 0.15rem 1.75rem 0
rgba(58, 59, 69, 0.15); } .employee-avatar { width: 150px; height: 150px; border-radius: 50%;
background-color: var(--bs-primary); color: white; display: flex; align-items: center;
justify-content: center; font-size: 3rem; margin: 0 auto 1rem; font-weight: 700; } .employee-info {
display: flex; flex-wrap: wrap; } .employee-info-item { width: 50%; padding: 0.5rem 1rem; }
.employee-info-label { font-weight: 700; text-transform: uppercase; font-size: 0.875rem; color:
var(--bs-secondary); } .employee-info-value { font-size: 1.1rem; } /* Responsive adjustments */
@media (max-width: 768px) { .sidebar { width: 90px; } .sidebar .nav-item span { display: none; }
.content-wrapper { margin-left: 90px; } .employee-info-item { width: 100%; } } /* DataTables
customization */ .dataTables_wrapper .dataTables_filter { margin-bottom: 1rem; }
.dataTables_wrapper .dataTables_length select { padding: 0.375rem 2.25rem 0.375rem 0.75rem;
font-size: 1rem; border-radius: 0.25rem; } .dataTables_wrapper .dataTables_paginate
.paginate_button.current { background-color: var(--bs-primary); color: white !important; border:
none; } .dataTables_wrapper .dataTables_paginate .paginate_button:hover { background-color:
var(--bs-light); color: var(--bs-primary) !important; } /* Chart containers */ .chart-container {
position: relative; height: 300px; margin-bottom: 2rem; }
```

static/css/custom.css

static/css/custom.css

```
/* Custom styles for Employee Management System */ /* Global Styles */ :root { --sidebar-width: 250px; } /* Sidebar styles */ .sidebar { width: var(--sidebar-width); height: 100vh; position: fixed; left: 0; top: 0; z-index: 100; transition: all 0.3s; overflow-y: auto; } .sidebar-toggled .sidebar { width: 90px; } .sidebar-toggled .sidebar .nav-item span { display: none; } .sidebar .nav-item .nav-link { padding: 0.75rem 1rem; width: 100%; } .sidebar .nav-item .nav-link i { margin-right: 0.5rem; } .sidebar-divider { margin: 1rem 0; border-top: 1px solid rgba(255, 255, 255, 0.15); } /* Main content wrapper */ .content-wrapper { margin-left: var(--sidebar-width); padding: 2rem; min-height: 100vh; transition: all 0.3s; } .sidebar-toggled .content-wrapper { margin-left: 90px; } /* Dashboard cards */ .dashboard-card { border-radius: 0.5rem; border-left: 0.25rem solid var(--bs-primary); box-shadow: 0 0.15rem 1.75rem 0 rgba(58, 59, 69, 0.15); transition: transform 0.2s; } .dashboard-card:hover { transform: translateY(-5px); } .dashboard-card-header { font-weight: 700; text-transform: uppercase; } .dashboard-card-body { font-size: 2.5rem; font-weight: 700; } /* Star rating system */ .rating-container { display: flex; align-items: center; margin-bottom: 1rem; } .rating-stars { display: inline-flex; font-size: 1.5rem; cursor: pointer; } .rating-star { color: var(--bs-secondary); padding: 0 0.25rem; transition: color 0.2s; } .rating-star:hover, .rating-star.rated { color: var(--bs-warning); } /* Feedback card styles */ .feedback-card { border-radius: 0.5rem; box-shadow: 0 0.15rem 1.75rem 0 rgba(58, 59, 69, 0.15); margin-bottom: 1.5rem; } .feedback-card .card-header { display: flex; justify-content: space-between; align-items: center; } .feedback-rating { display: inline-flex; font-size: 1.2rem; } .feedback-star { color: var(--bs-warning); padding: 0 0.1rem; } .feedback-quarter { font-weight: 700; color: var(--bs-primary); } .feedback-date { font-size: 0.875rem; color: var(--bs-secondary); } /* Employee detail page */ .employee-profile { padding: 2rem; background-color: var(--bs-light); border-radius: 0.5rem; box-shadow: 0 0.15rem 1.75rem 0 rgba(58, 59, 69, 0.15); } .employee-avatar { width: 150px; height: 150px; border-radius: 50%; background-color: var(--bs-primary); color: white; display: flex; align-items: center; justify-content: center; font-size: 3rem; margin: 0 auto 1rem; font-weight: 700; } .employee-info { display: flex; flex-wrap: wrap; } .employee-info-item { width: 50%; padding: 0.5rem 1rem; } .employee-info-label { font-weight: 700; text-transform: uppercase; font-size: 0.875rem; color: var(--bs-secondary); } .employee-info-value { font-size: 1.1rem; } /* Responsive adjustments */ @media (max-width: 768px) { .sidebar { width: 90px; } .sidebar .nav-item span { display: none; } .content-wrapper { margin-left: 90px; } .employee-info-item { width: 100%; } } /* DataTables customization */ .dataTables_wrapper .dataTables_filter { margin-bottom: 1rem; } .dataTables_wrapper .dataTables_length select { padding: 0.375rem 2.25rem 0.375rem 0.75rem; font-size: 1rem; border-radius: 0.25rem; } .dataTables_wrapper .dataTables_paginate .paginate_button.current { background-color: var(--bs-primary); color: white !important; border: none; } .dataTables_wrapper .dataTables_paginate .paginate_button:hover { background-color: var(--bs-light); color: var(--bs-primary) !important; } /* Chart containers */ .chart-container { position: relative; height: 300px; margin-bottom: 2rem; }
```

./templates/base.html

./templates/base.html

```
<!DOCTYPE html> <html lang="en" data-bs-theme="dark"> <head> <meta charset="UTF-8"> <meta
name="viewport" content="width=device-width, initial-scale=1.0"> <title>{% block title %}Employee
Management System{% endblock %}</title> <!-- Bootstrap CSS from Replit CDN --> <link
rel="stylesheet" href="https://cdn.replit.com/agent/bootstrap-agent-dark-theme.min.css"> <!-- Font
Awesome --> <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css"> <!-- DataTables
CSS --> <link rel="stylesheet"
href="https://cdn.datatables.net/1.13.4/css/dataTables.bootstrap5.min.css"> <link rel="stylesheet"
href="https://cdn.datatables.net/buttons/2.3.6/css/buttons.bootstrap5.min.css"> <!-- Custom CSS -->
<link rel="stylesheet" href="{{ url_for('static', filename='css/custom.css') }}"> {% block
extra_css %}{% endblock %} </head> <body> <!-- Sidebar --> {% if current_user.is_authenticated %}
<div class="sidebar bg-dark"> <div class="sidebar-brand d-flex align-items-center
justify-content-center p-3"> <div class="sidebar-brand-icon"> <i class="fas fa-users"></i> </div>
<div class="sidebar-brand-text mx-3">EMS</div> </div> <hr class="sidebar-divider"> <ul class="nav
flex-column"> <li class="nav-item"> <a class="nav-link {% if request.endpoint and request.endpoint
== 'employee.dashboard' %}active{% endif %}" href="{{ url_for('employee.dashboard') }}"> <i
class="fas fa-fw fa-tachometer-alt"></i> <span>Dashboard</span> </a> </li> <li class="nav-item"> <a
class="nav-link {% if request.endpoint and 'employee_list' in request.endpoint %}active{% endif %}"
href="{{ url_for('employee.employee_list') }}"> <i class="fas fa-fw fa-users"></i>
<span>Employees</span> </a> </li> <li class="nav-item"> <a class="nav-link {% if request.endpoint
and 'feedback' in request.endpoint %}active{% endif %}" href="{{ url_for('feedback.feedback_list')
}}"> <i class="fas fa-fw fa-comments"></i> <span>Feedback</span> </a> </li> {% if
current_user.is_manager %} <li class="nav-item"> <a class="nav-link {% if request.endpoint and
'import_export' in request.endpoint %}active{% endif %}" href="{{ url_for('employee.import_export')
}}"> <i class="fas fa-fw fa-file-excel"></i> <span>Import/Export</span> </a> </li> <li
class="nav-item"> <a class="nav-link" href="{{ url_for('docs.generate_and_download_docs') }}"> <i
class="fas fa-fw fa-file-pdf"></i> <span>Download Documentation</span> </a> </li> {% endif %} <hr
class="sidebar-divider"> <li class="nav-item"> <a class="nav-link" href="{{ url_for('auth.profile')
}}"> <i class="fas fa-fw fa-user"></i> <span>Profile</span> </a> </li> <li class="nav-item"> <a
class="nav-link" href="{{ url_for('auth.logout') }}"> <i class="fas fa-fw fa-sign-out-alt"></i>
<span>Logout</span> </a> </li> </ul> </div> {% endif %} <!-- Main Content --> <div
class="content-wrapper {% if not current_user.is_authenticated %}ml-0{% endif %}"> {% if
current_user.is_authenticated %} <!-- Topbar --> <nav class="navbar navbar-expand navbar-dark
bg-dark mb-4"> <button id="sidebarToggle" class="btn btn-link d-md-none rounded-circle mr-3"> <i
class="fa fa-bars"></i> </button> <div class="d-none d-md-inline-block ms-auto form-inline mr-0
navbar-search"> <div class="input-group"> <input type="text" id="searchInput" class="form-control
bg-light border-0 small" placeholder="Search..." aria-label="Search"> <div
class="input-group-append"> <button class="btn btn-primary" type="button"> <i class="fas fa-search
fa-sm"></i> </button> </div> </div> </div> <ul class="navbar-nav ml-auto ml-md-0"> <li
class="nav-item dropdown no-arrow"> <a class="nav-link dropdown-toggle" href="#" id="userDropdown"
role="button" data-bs-toggle="dropdown" aria-haspopup="true" aria-expanded="false"> <span
class="mr-2 d-none d-lg-inline text-gray-600 small">{{ current_user.username }}</span> <i
class="fas fa-user-circle fa-fw"></i> </a> <div class="dropdown-menu dropdown-menu-right shadow
animated--grow-in" aria-labelledby="userDropdown"> <a class="dropdown-item" href="{{
url_for('auth.profile') }}"> <i class="fas fa-user fa-sm fa-fw mr-2 text-gray-400"></i> Profile
</a> <div class="dropdown-divider"></div> <a class="dropdown-item" href="{{ url_for('auth.logout')
}}"> <i class="fas fa-sign-out-alt fa-sm fa-fw mr-2 text-gray-400"></i> Logout </a> </div> </li>
</ul> </nav> {% endif %} <!-- Flash Messages --> <div class="container-fluid"> {% with messages =
get_flashed_messages(with_categories=true) %} {% if messages %} {% for category, message in
messages %} <div class="alert alert-{{ category }} alert-dismissible fade show" role="alert"> {{
message }} <button type="button" class="btn-close" data-bs-dismiss="alert"
aria-label="Close"></button> </div> {% endfor %} {% endif %} {% endwith %} <!-- Page Content --> {%
block content %}{% endblock %} </div> </div> <!-- Bootstrap JS Bundle --> <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script> <!--
jQuery --> <script src="https://code.jquery.com/jquery-3.6.4.min.js"></script> <!-- DataTables JS
--> <script src="https://cdn.datatables.net/1.13.4/js/jquery.dataTables.min.js"></script> <script
src="https://cdn.datatables.net/1.13.4/js/dataTables.bootstrap5.min.js"></script> <script
src="https://cdn.datatables.net/buttons/2.3.6/js/dataTables.buttons.min.js"></script> <script
src="https://cdn.datatables.net/buttons/2.3.6/js/buttons.bootstrap5.min.js"></script> <script
src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.10.1/jszip.min.js"></script> <script
src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/pdfmake.min.js"></script> <script
src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/vfs_fonts.js"></script> <script
src="https://cdn.datatables.net/buttons/2.3.6/js/buttons.html5.min.js"></script> <script
src="https://cdn.datatables.net/buttons/2.3.6/js/buttons.print.min.js"></script> <!-- Chart.js -->
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script> <!-- Custom JS --> <script src="{{
url_for('static', filename='js/main.js') }}"></script> <script src="{{ url_for('static',
filename='js/datatables-config.js') }}"></script> {% block extra_js %}{% endblock %} </body>
</html>
```

./templates/dashboard.html

./templates/dashboard.html

```
{% extends "base.html" %} {% block title %}Dashboard - Employee Management System{% endblock %} {%
block content %} <div class="container-fluid"> <h1 class="h3 mb-4 text-gray-800">Dashboard</h1>
<!-- Content Row - Stats --> <div class="row"> <!-- Total Employees Card --> <div class="col-xl-3
col-md-6 mb-4"> <div class="card dashboard-card border-left-primary shadow h-100 py-2"> <div
class="card-body"> <div class="row no-gutters align-items-center"> <div class="col mr-2"> <div
class="text-xs font-weight-bold text-primary text-uppercase mb-1 dashboard-card-header"> Total
Employees </div> <div class="h5 mb-0 font-weight-bold text-gray-800 dashboard-card-body"> {{
total_employees }} </div> </div> <div class="col-auto"> <i class="fas fa-users fa-2x
text-gray-300"></i> </div> </div> </div> </div> </div> {% if is_manager %} <!-- Managed Employees
Card --> <div class="col-xl-3 col-md-6 mb-4"> <div class="card dashboard-card border-left-success
shadow h-100 py-2"> <div class="card-body"> <div class="row no-gutters align-items-center"> <div
class="col mr-2"> <div class="text-xs font-weight-bold text-success text-uppercase mb-1
dashboard-card-header"> Team Members </div> <div class="h5 mb-0 font-weight-bold text-gray-800
dashboard-card-body"> {{ managed_employees }} </div> </div> <div class="col-auto"> <i class="fas
fa-user-friends fa-2x text-gray-300"></i> </div> </div> </div> </div> </div> <!-- Recent Feedback
Card --> <div class="col-xl-3 col-md-6 mb-4"> <div class="card dashboard-card border-left-info
shadow h-100 py-2"> <div class="card-body"> <div class="row no-gutters align-items-center"> <div
class="col mr-2"> <div class="text-xs font-weight-bold text-info text-uppercase mb-1
dashboard-card-header"> Recent Feedback </div> <div class="h5 mb-0 font-weight-bold text-gray-800
dashboard-card-body"> {{ recent_feedback|length }} </div> </div> <div class="col-auto"> <i
class="fas fa-comments fa-2x text-gray-300"></i> </div> </div> </div> </div> </div> <!-- Pending
Feedback Card --> <div class="col-xl-3 col-md-6 mb-4"> <div class="card dashboard-card
border-left-warning shadow h-100 py-2"> <div class="card-body"> <div class="row no-gutters
align-items-center"> <div class="col mr-2"> <div class="text-xs font-weight-bold text-warning
text-uppercase mb-1 dashboard-card-header"> Current Quarter </div> <div class="h5 mb-0
font-weight-bold text-gray-800 dashboard-card-body"> Q{{ ((now.month - 1) // 3) + 1 }}-{{ now.year
}} </div> </div> <div class="col-auto"> <i class="fas fa-calendar fa-2x text-gray-300"></i> </div>
</div> </div> </div> </div> {% else %} <!-- Employee Info Cards --> {% if employee %} <div
class="col-xl-3 col-md-6 mb-4"> <div class="card dashboard-card border-left-success shadow h-100
py-2"> <div class="card-body"> <div class="row no-gutters align-items-center"> <div class="col
mr-2"> <div class="text-xs font-weight-bold text-success text-uppercase mb-1
dashboard-card-header"> Your Role </div> <div class="h5 mb-0 font-weight-bold text-gray-800
dashboard-card-body"> {{ employee.role }} </div> </div> <div class="col-auto"> <i class="fas
fa-user-tag fa-2x text-gray-300"></i> </div> </div> </div> </div> </div> <div class="col-xl-3
col-md-6 mb-4"> <div class="card dashboard-card border-left-info shadow h-100 py-2"> <div
class="card-body"> <div class="row no-gutters align-items-center"> <div class="col mr-2"> <div
class="text-xs font-weight-bold text-info text-uppercase mb-1 dashboard-card-header"> Team </div>
<div class="h5 mb-0 font-weight-bold text-gray-800 dashboard-card-body"> {{ employee.team }} </div>
</div> <div class="col-auto"> <i class="fas fa-users-cog fa-2x text-gray-300"></i> </div> </div>
</div> </div> </div> <div class="col-xl-3 col-md-6 mb-4"> <div class="card dashboard-card
border-left-warning shadow h-100 py-2"> <div class="card-body"> <div class="row no-gutters
align-items-center"> <div class="col mr-2"> <div class="text-xs font-weight-bold text-warning
text-uppercase mb-1 dashboard-card-header"> Recent Feedback </div> <div class="h5 mb-0
font-weight-bold text-gray-800 dashboard-card-body"> {{ recent_feedback|length }} </div> </div>
<div class="col-auto"> <i class="fas fa-comments fa-2x text-gray-300"></i> </div> </div> </div>
</div> </div> {% endif %} {% endif %} </div> <!-- Content Row - Details --> <div class="row"> {% if
is_manager %} <!-- Team Skills Distribution --> <div class="col-lg-6 mb-4"> <div class="card shadow
mb-4"> <div class="card-header py-3"> <h6 class="m-0 font-weight-bold text-primary">Team Skills
Distribution</h6> </div> <div class="card-body"> <div class="chart-container"> <canvas
id="teamSkillsChart"></canvas> </div> <!-- Hidden data for the chart --> {% for skill, count in
team_skills %} <div class="skill-stat d-none" data-skill="{{ skill }}" data-count="{{ count
}}"></div> {% endfor %} </div> </div> </div> <!-- Recent Feedback List --> <div class="col-lg-6
mb-4"> <div class="card shadow mb-4"> <div class="card-header py-3"> <h6 class="m-0
font-weight-bold text-primary">Recent Feedback Provided</h6> </div> <div class="card-body"> {% if
recent_feedback %} <div class="table-responsive"> <table class="table table-bordered" width="100%"
cellspacing="0"> <thead> <tr> <th>Employee</th> <th>Rating</th> <th>Quarter</th> <th>Date</th>
</tr> </thead> <tbody> {% for feedback in recent_feedback %} <tr> <td>{{
feedback.employee.full_name }}</td> <td> <div class="feedback-rating"> {% for i in
range(feedback.rating) %} <i class="fas fa-star feedback-star"></i> {% endfor %} {% for i in
range(5 - feedback.rating) %} <i class="far fa-star feedback-star"></i> {% endfor %} </div> </td>
<td>{{ feedback.quarter }}</td> <td>{{ feedback.feedback_date.strftime('%Y-%m-%d') }}</td> </tr> {%
endfor %} </tbody> </table> </div> <a href="{{ url_for('feedback.feedback_list') }}" class="btn
btn-primary btn-sm"> View All Feedback </a> {% else %} <p>No recent feedback provided.</p> {% endif
%} </div> </div> </div> <!-- Quick Actions --> <div class="col-lg-12 mb-4"> <div class="card shadow
mb-4"> <div class="card-header py-3"> <h6 class="m-0 font-weight-bold text-primary">Quick
Actions</h6> </div> <div class="card-body"> <div class="row"> <div class="col-md-4 mb-3"> <a
href="{{ url_for('employee.create_employee') }}" class="btn btn-primary w-100"> <i class="fas
fa-user-plus mr-2"></i> Add New Employee </a> </div> <div class="col-md-4 mb-3"> <a href="{{
url_for('employee.import_export') }}" class="btn btn-success w-100"> <i class="fas fa-file-import
mr-2"></i> Import Employee Data </a> </div> <div class="col-md-4 mb-3"> <a href="{{
```

```
url_for('employee.import_export') }}?action=export" class="btn btn-info w-100"> <i class="fas
fa-file-export mr-2"></i> Export Employee Data </a> </div> </div> </div> </div> </div> {% else %}
<!-- Employee Recent Feedback --> {% if employee and recent_feedback %} <div class="col-lg-12
mb-4"> <div class="card shadow mb-4"> <div class="card-header py-3"> <h6 class="m-0
font-weight-bold text-primary">Your Recent Feedback</h6> </div> <div class="card-body"> {% for
feedback in recent_feedback %} <div class="feedback-card"> <div class="card-header"> <div> <span
class="feedback-quarter">{{ feedback.quarter }}</span> <span class="feedback-date">{{
feedback.feedback_date.strftime('%Y-%m-%d') }}</span> </div> <div class="feedback-rating"> {% for i
in range(feedback.rating) %} <i class="fas fa-star feedback-star"></i> {% endfor %} {% for i in
range(5 - feedback.rating) %} <i class="far fa-star feedback-star"></i> {% endfor %} </div> </div>
<div class="card-body"> <p>{{ feedback.feedback_text }}</p> <small class="text-muted">Provided by:
{{ feedback.provided_by.username }}</small> </div> </div> {% endfor %} <a href="{{
url_for('feedback.feedback_list') }}" class="btn btn-primary btn-sm mt-3"> View All Feedback </a>
</div> </div> </div> {% elif employee %} <div class="col-lg-12 mb-4"> <div class="card shadow
mb-4"> <div class="card-header py-3"> <h6 class="m-0 font-weight-bold text-primary">Your
Feedback</h6> </div> <div class="card-body"> <p>No feedback has been provided yet.</p> </div>
</div> </div> {% endif %} <!-- Employee Profile --> {% if employee %} <div class="col-lg-12 mb-4">
<div class="card shadow mb-4"> <div class="card-header py-3"> <h6 class="m-0 font-weight-bold
text-primary">Your Profile</h6> </div> <div class="card-body"> <div class="row"> <div
class="col-md-3 text-center"> <div class="employee-avatar"> {{ employee.full_name[0] }} </div> <h4
class="mt-3">{{ employee.full_name }}</h4> <p class="text-muted">{{ employee.designation }}</p>
</div> <div class="col-md-9"> <div class="employee-info"> <div class="employee-info-item"> <div
class="employee-info-label">Employee ID</div> <div class="employee-info-value">{{ employee.benzyl
}}</div> </div> <div class="employee-info-item"> <div class="employee-info-label">Joining
Date</div> <div class="employee-info-value">{{ employee.joining_date }}</div> </div> <div
class="employee-info-item"> <div class="employee-info-label">Role</div> <div
class="employee-info-value">{{ employee.role }}</div> </div> <div class="employee-info-item"> <div
class="employee-info-label">Skill</div> <div class="employee-info-value">{{ employee.skill }}</div>
</div> <div class="employee-info-item"> <div class="employee-info-label">Team</div> <div
class="employee-info-value">{{ employee.team }}</div> </div> <div class="employee-info-item"> <div
class="employee-info-label">Grade</div> <div class="employee-info-value">{{ employee.grade }}</div>
</div> <div class="employee-info-item"> <div class="employee-info-label">Location</div> <div
class="employee-info-value">{{ employee.location }}</div> </div> </div> </div> </div> </div> </div>
</div> {% endif %} {% endif %} </div> </div> {% endblock %} {% block extra_js %} <script> // Use
the current date for the dashboard document.addEventListener('DOMContentLoaded', function() { //
Set the current date object for the template window.now = new Date(); }); </script> {% endblock %}
```

./templates/employee_detail.html

> ./templates/employee_detail.html

```
{% extends "base.html" %} {% block title %}{{ employee.full_name }} - Employee Management System{%
endblock %} {% block content %} <div class="container-fluid"> <div class="d-flex
justify-content-between align-items-center mb-4"> <h1 class="h3 text-gray-800">Employee
Details</h1> <div> <a href="{{ url_for('employee.employee_list') }}" class="btn btn-secondary"> <i
class="fas fa-arrow-left mr-1"></i> Back to List </a> {% if is_manager and employee.manager_id ==
current_user.id %} <a href="{{ url_for('employee.edit_employee', employee_id=employee.id) }}"
class="btn btn-warning"> <i class="fas fa-edit mr-1"></i> Edit </a> {% endif %} </div> </div> <!--
Employee Profile Card --> <div class="row mb-4"> <div class="col-lg-12"> <div class="card shadow">
<div class="card-header py-3 d-flex justify-content-between align-items-center"> <h6 class="m-0
font-weight-bold text-primary">Employee Profile</h6> {% if is_manager and employee.manager_id ==
current_user.id %} <a href="{{ url_for('feedback.create_feedback', employee_id=employee.id) }}"
class="btn btn-success btn-sm"> <i class="fas fa-comment-dots mr-1"></i> Provide Feedback </a> {%
endif %} </div> <div class="card-body"> <div class="row"> <div class="col-md-3 text-center"> <div
class="employee-avatar"> {{ employee.full_name[0] }} </div> <h4 class="mt-3">{{ employee.full_name
}}</h4> <p class="text-muted">{{ employee.designation }}</p> </div> <div class="col-md-9"> <div
class="employee-info"> <div class="employee-info-item"> <div class="employee-info-label">Employee
ID</div> <div class="employee-info-value">{{ employee.benzyl }}</div> </div> <div
class="employee-info-item"> <div class="employee-info-label">Joining Date</div> <div
class="employee-info-value">{{ employee.joining_date }}</div> </div> <div
class="employee-info-item"> <div class="employee-info-label">Role</div> <div
class="employee-info-value">{{ employee.role }}</div> </div> <div class="employee-info-item"> <div
class="employee-info-label">Skill</div> <div class="employee-info-value">{{ employee.skill }}</div>
</div> <div class="employee-info-item"> <div class="employee-info-label">Team</div> <div
class="employee-info-value">{{ employee.team }}</div> </div> <div class="employee-info-item"> <div
class="employee-info-label">Grade</div> <div class="employee-info-value">{{ employee.grade }}</div>
</div> <div class="employee-info-item"> <div class="employee-info-label">Location</div> <div
class="employee-info-value">{{ employee.location }}</div> </div> <div class="employee-info-item">
<div class="employee-info-label">Manager</div> <div class="employee-info-value"> {% if manager %}
{{ manager.username }} {% else %} Not assigned {% endif %} </div> </div> </div> </div> </div>
</div> </div> </div> </div> <!-- Feedback History Card --> <div class="row"> <div
class="col-lg-12"> <div class="card shadow mb-4"> <div class="card-header py-3"> <h6 class="m-0
font-weight-bold text-primary">Feedback History</h6> </div> <div class="card-body"> {% if feedback
%} <div class="table-responsive"> <table class="table table-bordered" id="feedbackTable"
width="100%" cellspacing="0"> <thead> <tr> <th>Quarter</th> <th>Rating</th> <th>Feedback</th>
<th>Provided By</th> <th>Date</th> {% if is_manager %} <th>Actions</th> {% endif %} </tr> </thead>
<tbody> {% for fb in feedback %} <tr> <td>{{ fb.quarter }}</td> <td> <div class="feedback-rating">
{% for i in range(fb.rating) %} <i class="fas fa-star feedback-star"></i> {% endfor %} {% for i in
range(5 - fb.rating) %} <i class="far fa-star feedback-star"></i> {% endfor %} </div> </td> <td>{{
fb.feedback_text }}</td> <td>{{ fb.provided_by.username }}</td> <td>{{
fb.feedback_date.strftime('%Y-%m-%d') }}</td> {% if is_manager and fb.provided_by_id ==
current_user.id %} <td> <div class="btn-group" role="group"> <a href="{{
url_for('feedback.edit_feedback', feedback_id=fb.id) }}" class="btn btn-warning btn-sm"
data-bs-toggle="tooltip" title="Edit Feedback"> <i class="fas fa-edit"></i> </a> <button
type="button" class="btn btn-danger btn-sm confirm-delete" data-bs-toggle="modal"
data-bs-target="#deleteFeedbackModal{{ fb.id }}" title="Delete Feedback"> <i class="fas
fa-trash"></i> </button> <!-- Delete Modal --> <div class="modal fade" id="deleteFeedbackModal{{
fb.id }}" tabindex="-1" aria-labelledby="deleteFeedbackModalLabel" aria-hidden="true"> <div
class="modal-dialog"> <div class="modal-content"> <div class="modal-header"> <h5
class="modal-title" id="deleteFeedbackModalLabel"> Confirm Delete </h5> <button type="button"
class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button> </div> <div
class="modal-body"> Are you sure you want to delete this feedback? This action cannot be undone.
</div> <div class="modal-footer"> <button type="button" class="btn btn-secondary"
data-bs-dismiss="modal">Cancel</button> <form action="{{ url_for('feedback.delete_feedback',
feedback_id=fb.id) }}" method="POST"> <button type="submit" class="btn btn-danger">Delete</button>
</form> </div> </div> </div> </div> </div> </td> {% elif is_manager %} <td>-</td> {% endif %} </tr>
{% endfor %} </tbody> </table> </div> {% else %} <div class="text-center py-5"> <i class="fas
fa-comments fa-4x mb-3 text-gray-300"></i> <p class="lead text-gray-500">No feedback records
found</p> {% if is_manager and employee.manager_id == current_user.id %} <a href="{{
url_for('feedback.create_feedback', employee_id=employee.id) }}" class="btn btn-success mt-3"> <i
class="fas fa-comment-dots mr-1"></i> Provide First Feedback </a> {% endif %} </div> {% endif %}
</div> </div> </div> </div> </div> {% endblock %}
```

./templates/employee_form.html

./templates/employee_form.html

```
{% extends "base.html" %} {% block title %} {% if action == 'create' %} Add New Employee - Employee
Management System {% else %} Edit Employee - Employee Management System {% endif %} {% endblock %}
{% block content %} <div class="container-fluid"> <div class="d-flex justify-content-between
align-items-center mb-4"> <h1 class="h3 text-gray-800"> {% if action == 'create' %} Add New
Employee {% else %} Edit Employee {% endif %} </h1> <a href="{{ url_for('employee.employee_list')
}}" class="btn btn-secondary"> <i class="fas fa-arrow-left mr-1"></i> Back to List </a> </div> <!--
Employee Form Card --> <div class="row"> <div class="col-lg-12"> <div class="card shadow mb-4">
<div class="card-header py-3"> <h6 class="m-0 font-weight-bold text-primary"> {% if action ==
'create' %} Employee Information {% else %} Update Employee Information {% endif %} </h6> </div>
<div class="card-body"> <form method="POST" action="{% if action == 'create' %}{{
url_for('employee.create_employee') }}{% else %}{{ url_for('employee.edit_employee',
employee_id=employee.id) }}{% endif %}"> <div class="row"> <div class="col-md-6 mb-3"> <label
for="full_name" class="form-label">Full Name</label> <input type="text" class="form-control"
id="full_name" name="full_name" value="{{ employee.full_name if employee else '' }}" required>
</div> <div class="col-md-6 mb-3"> <label for="benzyl" class="form-label">Employee ID
(Benzyl)</label> <input type="text" class="form-control" id="benzyl" name="benzyl" value="{{
employee.benzyl if employee else '' }}" required> </div> </div> <div class="row"> <div
class="col-md-6 mb-3"> <label for="joining_date" class="form-label">Joining Date</label> <input
type="date" class="form-control" id="joining_date" name="joining_date" value="{{
employee.joining_date if employee else '' }}" required> </div> <div class="col-md-6 mb-3"> <label
for="role" class="form-label">Role</label> <input type="text" class="form-control" id="role"
name="role" value="{{ employee.role if employee else '' }}" required> </div> </div> <div
class="row"> <div class="col-md-6 mb-3"> <label for="skill" class="form-label">Skill</label> <input
type="text" class="form-control" id="skill" name="skill" value="{{ employee.skill if employee else
'' }}" required> </div> <div class="col-md-6 mb-3"> <label for="team"
class="form-label">Team</label> <input type="text" class="form-control" id="team" name="team"
value="{{ employee.team if employee else '' }}" required> </div> </div> <div class="row"> <div
class="col-md-6 mb-3"> <label for="grade" class="form-label">Grade</label> <input type="text"
class="form-control" id="grade" name="grade" value="{{ employee.grade if employee else '' }}"
required> </div> <div class="col-md-6 mb-3"> <label for="designation"
class="form-label">Designation</label> <input type="text" class="form-control" id="designation"
name="designation" value="{{ employee.designation if employee else '' }}" required> </div> </div>
<div class="row"> <div class="col-md-6 mb-3"> <label for="location"
class="form-label">Location</label> <input type="text" class="form-control" id="location"
name="location" value="{{ employee.location if employee else '' }}" required> </div> <div
class="col-md-6 mb-3"> <label for="manager_id" class="form-label">Manager</label> <select
class="form-select" id="manager_id" name="manager_id"> <option value="">Select Manager</option> {%
for manager in managers %} <option value="{{ manager.id }}" {% if employee and employee.manager_id
== manager.id %}selected{% endif %} {% if not employee and manager.id == current_user.id
%}selected{% endif %}> {{ manager.username }} </option> {% endfor %} </select> </div> </div> <div
class="mt-4"> <button type="submit" class="btn btn-primary"> {% if action == 'create' %} <i
class="fas fa-plus-circle mr-1"></i> Create Employee {% else %} <i class="fas fa-save mr-1"></i>
Update Employee {% endif %} </button> <a href="{{ url_for('employee.employee_list') }}" class="btn
btn-secondary"> <i class="fas fa-times-circle mr-1"></i> Cancel </a> </div> </form> </div> </div>
</div> </div> </div> {% endblock %}
```

./templates/employee_list.html

./templates/employee_list.html

```
{% extends "base.html" %} {% block title %}Employees - Employee Management System{% endblock %} {%
block content %} <div class="container-fluid"> <div class="d-flex justify-content-between
align-items-center mb-4"> <h1 class="h3 text-gray-800">Employees</h1> {% if is_manager %} <a
href="{{ url_for('employee.create_employee') }}" class="btn btn-primary"> <i class="fas
fa-user-plus mr-1"></i> Add Employee </a> {% endif %} </div> <!-- Employee List Table --> <div
class="card shadow mb-4"> <div class="card-header py-3"> <h6 class="m-0 font-weight-bold
text-primary">Employee Directory</h6> </div> <div class="card-body"> {% if employees %} <div
class="table-responsive"> <table class="table table-bordered" id="employeeTable" width="100%"
cellspacing="0"> <thead> <tr> <th>Full Name</th> <th>Benzyl</th> <th>Role</th> <th>Skill</th>
<th>Team</th> <th>Grade</th> <th>Location</th> <th>Actions</th> </tr> </thead> <tbody> {% for
employee in employees %} <tr {% if employee.manager_id == current_user.id %}class="table-primary"{%
endif %}> <td>{{ employee.full_name }}</td> <td>{{ employee.benzyl }}</td> <td>{{ employee.role
}}</td> <td>{{ employee.skill }}</td> <td>{{ employee.team }}</td> <td>{{ employee.grade }}</td>
<td>{{ employee.location }}</td> <td> <div class="btn-group" role="group"> <a href="{{
url_for('employee.employee_detail', employee_id=employee.id) }}" class="btn btn-info btn-sm"
data-bs-toggle="tooltip" title="View Details"> <i class="fas fa-eye"></i> </a> {% if is_manager and
employee.manager_id == current_user.id %} <a href="{{ url_for('feedback.create_feedback',
employee_id=employee.id) }}" class="btn btn-success btn-sm" data-bs-toggle="tooltip" title="Provide
Feedback"> <i class="fas fa-comment-dots"></i> </a> <a href="{{ url_for('employee.edit_employee',
employee_id=employee.id) }}" class="btn btn-warning btn-sm" data-bs-toggle="tooltip" title="Edit
Employee"> <i class="fas fa-edit"></i> </a> <button type="button" class="btn btn-danger btn-sm
confirm-delete" data-bs-toggle="modal" data-bs-target="#deleteModal{{ employee.id }}" title="Delete
Employee"> <i class="fas fa-trash"></i> </button> <!-- Delete Modal --> <div class="modal fade"
id="deleteModal{{ employee.id }}" tabindex="-1" aria-labelledby="deleteModalLabel"
aria-hidden="true"> <div class="modal-dialog"> <div class="modal-content"> <div
class="modal-header"> <h5 class="modal-title" id="deleteModalLabel"> Confirm Delete </h5> <button
type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button> </div> <div
class="modal-body"> Are you sure you want to delete {{ employee.full_name }}? This action cannot be
undone. </div> <div class="modal-footer"> <button type="button" class="btn btn-secondary"
data-bs-dismiss="modal">Cancel</button> <form action="{{ url_for('employee.delete_employee',
employee_id=employee.id) }}" method="POST"> <button type="submit" class="btn
btn-danger">Delete</button> </form> </div> </div> </div> </div> {% endif %} </div> </td> </tr> {%
endfor %} </tbody> </table> </div> {% else %} <div class="text-center py-5"> <i class="fas fa-users
fa-4x mb-3 text-gray-300"></i> <p class="lead text-gray-500">No employees found</p> {% if
is_manager %} <a href="{{ url_for('employee.create_employee') }}" class="btn btn-primary mt-3"> <i
class="fas fa-user-plus mr-1"></i> Add First Employee </a> {% endif %} </div> {% endif %} </div>
</div> </div> {% endblock %}
```

./templates/errors/404.html

> ./templates/errors/404.html

```
<!DOCTYPE html> <html lang="en" data-bs-theme="dark"> <head> <meta charset="UTF-8"> <meta
name="viewport" content="width=device-width, initial-scale=1.0"> <title>404 Not Found - Employee
Management System</title> <!-- Bootstrap CSS from Replit CDN --> <link rel="stylesheet"
href="https://cdn.replit.com/agent/bootstrap-agent-dark-theme.min.css"> <!-- Font Awesome --> <link
rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
</head> <body> <div class="container mt-5"> <div class="text-center"> <div
class="display-1">404</div> <p class="lead mb-5">Page Not Found</p> <p class="text-muted mb-4">It
looks like you found a glitch in the matrix...</p> <a href="/" class="btn btn-primary">&larr; Back
to Home</a> </div> </div> <!-- Bootstrap JS Bundle --> <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body> </html>
```

./templates/errors/500.html

./templates/errors/500.html

```
{% extends "base.html" %} {% block title %}500 Server Error - Employee Management System{% endblock
%} {% block content %} <div class="container-fluid"> <div class="text-center"> <div class="error
mx-auto" data-text="500">500</div> <p class="lead text-gray-800 mb-5">Internal Server Error</p> <p
class="text-gray-500 mb-0">Something went wrong. Please try again later.</p> <a href="{{
url_for('employee.dashboard') }}">&larr; Back to Dashboard</a> </div> </div> {% endblock %}
```

./templates/errors/base_error.html

> ./templates/errors/base_error.html

```
<!DOCTYPE html> <html lang="en" data-bs-theme="dark"> <head> <meta charset="UTF-8"> <meta
name="viewport" content="width=device-width, initial-scale=1.0"> <title>{% block title %}Error -
Employee Management System{% endblock %}</title> <!-- Bootstrap CSS from Replit CDN --> <link
rel="stylesheet" href="https://cdn.replit.com/agent/bootstrap-agent-dark-theme.min.css"> <!-- Font
Awesome --> <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css"> </head> <body>
<div class="container mt-5"> {% block content %}{% endblock %} </div> <!-- Bootstrap JS Bundle -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body> </html>
```

./templates/feedback_form.html

./templates/feedback_form.html

```
{% extends "base.html" %} {% block title %} {% if action == 'create' %} Provide Feedback - Employee
Management System {% else %} Edit Feedback - Employee Management System {% endif %} {% endblock %}
{% block content %} <div class="container-fluid"> <div class="d-flex justify-content-between
align-items-center mb-4"> <h1 class="h3 text-gray-800"> {% if action == 'create' %} Provide
Feedback for {{ employee.full_name }} {% else %} Edit Feedback for {{ employee.full_name }} {%
endif %} </h1> <a href="{{ url_for('feedback.employee_feedback', employee_id=employee.id) }}"
class="btn btn-secondary"> <i class="fas fa-arrow-left mr-1"></i> Back to Feedback </a> </div> <!--
Feedback Form Card --> <div class="row"> <div class="col-lg-12"> <div class="card shadow mb-4">
<div class="card-header py-3"> <h6 class="m-0 font-weight-bold text-primary"> {% if action ==
'create' %} New Feedback {% else %} Update Feedback {% endif %} </h6> </div> <div
class="card-body"> <form method="POST" action="{% if action == 'create' %}{{
url_for('feedback.create_feedback', employee_id=employee.id) }}{% else %}{{
url_for('feedback.edit_feedback', feedback_id=feedback.id) }}{% endif %}"> <div class="row"> <div
class="col-md-6 mb-3"> <label for="quarter" class="form-label">Quarter</label> <select
class="form-select" id="quarter" name="quarter" required> <option value="">Select Quarter</option>
{% set years = [2023, 2024, 2025] %} {% for year in years %} {% for q in range(1, 5) %} <option
value="Q{{ q }}-{{ year }}" {% if (feedback and feedback.quarter == 'Q' ~ q ~ '-' ~ year) or (not
feedback and current_quarter == 'Q' ~ q ~ '-' ~ year) %} selected {% endif %}> Q{{ q }}-{{ year }}
</option> {% endfor %} {% endfor %} </select> </div> <div class="col-md-6 mb-3"> <label
for="rating" class="form-label">Rating</label> <input type="hidden" id="rating" name="rating"
value="{{ feedback.rating if feedback else 0 }}" required> <div class="rating-container"> <div
class="rating-stars"> {% for i in range(1, 6) %} <i class="fas fa-star rating-star {% if feedback
and i <= feedback.rating %}rated{% endif %}" data-value="{{ i }}"></i> {% endfor %} </div> <span
class="ms-2 rating-label"> {% if feedback %} {{ feedback.rating }} out of 5 {% else %} Select a
rating {% endif %} </span> </div> </div> </div> <div class="mb-3"> <label for="feedback_text"
class="form-label">Feedback</label> <textarea class="form-control" id="feedback_text"
name="feedback_text" rows="6" required>{{ feedback.feedback_text if feedback else '' }}</textarea>
</div> <div class="mt-4"> <button type="submit" class="btn btn-primary"> {% if action == 'create'
%} <i class="fas fa-paper-plane mr-1"></i> Submit Feedback {% else %} <i class="fas fa-save
mr-1"></i> Update Feedback {% endif %} </button> <a href="{{ url_for('feedback.employee_feedback',
employee_id=employee.id) }}" class="btn btn-secondary"> <i class="fas fa-times-circle mr-1"></i>
Cancel </a> </div> </form> </div> </div> </div> </div> </div> {% endblock %} {% block extra_js %}
<script> document.addEventListener('DOMContentLoaded', function() { const ratingInput =
document.getElementById('rating'); const ratingStars = document.querySelectorAll('.rating-star');
const ratingLabel = document.querySelector('.rating-label'); ratingStars.forEach(star => {
star.addEventListener('click', function() { const value = this.getAttribute('data-value');
ratingInput.value = value; // Update star display ratingStars.forEach(s => { const starValue =
s.getAttribute('data-value'); if (starValue <= value) { s.classList.add('rated'); } else {
s.classList.remove('rated'); } }); // Update label ratingLabel.textContent = `${value} out of 5`;
}); }); }); </script> {% endblock %}
```

./templates/feedback_list.html

> ./templates/feedback_list.html

```
{% extends "base.html" %} {% block title %}Feedback - Employee Management System{% endblock %} {%
block content %} <div class="container-fluid"> <div class="d-flex justify-content-between
align-items-center mb-4"> <h1 class="h3 text-gray-800"> {% if view_type == 'provided' %} Feedback
Provided {% elif view_type == 'received' %} Feedback Received {% elif view_type == 'for_employee'
%} Feedback for {{ employee.full_name }} {% endif %} </h1> {% if is_manager and view_type ==
'for_employee' and employee.manager_id == current_user.id %} <a href="{{
url_for('feedback.create_feedback', employee_id=employee.id) }}" class="btn btn-success"> <i
class="fas fa-comment-dots mr-1"></i> Provide Feedback </a> {% endif %} </div> <!-- Feedback List
Card --> <div class="card shadow mb-4"> <div class="card-header py-3"> <h6 class="m-0
font-weight-bold text-primary"> {% if view_type == 'provided' %} Feedback You Have Provided {% elif
view_type == 'received' %} Feedback You Have Received {% elif view_type == 'for_employee' %} All
Feedback for {{ employee.full_name }} {% endif %} </h6> </div> <div class="card-body"> {% if
feedback %} <div class="table-responsive"> <table class="table table-bordered" id="feedbackTable"
width="100%" cellspacing="0"> <thead> <tr> {% if view_type != 'for_employee' %} <th>Employee</th>
{% endif %} <th>Quarter</th> <th>Rating</th> <th>Feedback</th> {% if view_type == 'received' or
view_type == 'for_employee' %} <th>Provided By</th> {% endif %} <th>Date</th> {% if is_manager and
view_type != 'received' %} <th>Actions</th> {% endif %} </tr> </thead> <tbody> {% for fb in
feedback %} <tr> {% if view_type != 'for_employee' %} <td> <a href="{{
url_for('employee.employee_detail', employee_id=fb.employee.id) }}"> {{ fb.employee.full_name }}
</a> </td> {% endif %} <td>{{ fb.quarter }}</td> <td> <div class="feedback-rating"> {% for i in
range(fb.rating) %} <i class="fas fa-star feedback-star"></i> {% endfor %} {% for i in range(5 -
fb.rating) %} <i class="far fa-star feedback-star"></i> {% endfor %} </div> </td> <td>{{
fb.feedback_text }}</td> {% if view_type == 'received' or view_type == 'for_employee' %} <td>{{
fb.provided_by.username }}</td> {% endif %} <td>{{ fb.feedback_date.strftime('%Y-%m-%d') }}</td> {%
if is_manager and view_type != 'received' and fb.provided_by_id == current_user.id %} <td> <div
class="btn-group" role="group"> <a href="{{ url_for('feedback.edit_feedback', feedback_id=fb.id)
}}" class="btn btn-warning btn-sm" data-bs-toggle="tooltip" title="Edit Feedback"> <i class="fas
fa-edit"></i> </a> <button type="button" class="btn btn-danger btn-sm confirm-delete"
data-bs-toggle="modal" data-bs-target="#deleteFeedbackModal{{ fb.id }}" title="Delete Feedback"> <i
class="fas fa-trash"></i> </button> <!-- Delete Modal --> <div class="modal fade"
id="deleteFeedbackModal{{ fb.id }}" tabindex="-1" aria-labelledby="deleteFeedbackModalLabel"
aria-hidden="true"> <div class="modal-dialog"> <div class="modal-content"> <div
class="modal-header"> <h5 class="modal-title" id="deleteFeedbackModalLabel"> Confirm Delete </h5>
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button> </div>
<div class="modal-body"> Are you sure you want to delete this feedback? This action cannot be
undone. </div> <div class="modal-footer"> <button type="button" class="btn btn-secondary"
data-bs-dismiss="modal">Cancel</button> <form action="{{ url_for('feedback.delete_feedback',
feedback_id=fb.id) }}" method="POST"> <button type="submit" class="btn btn-danger">Delete</button>
</form> </div> </div> </div> </div> </div> </td> {% elif is_manager and view_type != 'received' %}
<td>-</td> {% endif %} </tr> {% endfor %} </tbody> </table> </div> {% else %} <div
class="text-center py-5"> <i class="fas fa-comments fa-4x mb-3 text-gray-300"></i> <p class="lead
text-gray-500">No feedback records found</p> {% if is_manager and view_type == 'for_employee' and
employee.manager_id == current_user.id %} <a href="{{ url_for('feedback.create_feedback',
employee_id=employee.id) }}" class="btn btn-success mt-3"> <i class="fas fa-comment-dots mr-1"></i>
Provide First Feedback </a> {% endif %} </div> {% endif %} </div> </div> </div> {% endblock %}
```

./templates/import_export.html

./templates/import_export.html

```
{% extends "base.html" %} {% block title %}Import/Export - Employee Management System{% endblock %}
{% block content %} <div class="container-fluid"> <div class="d-flex justify-content-between
align-items-center mb-4"> <h1 class="h3 text-gray-800">Import/Export Employee Data</h1> <a href="{{
url_for('employee.dashboard') }}" class="btn btn-secondary"> <i class="fas fa-arrow-left mr-1"></i>
Back to Dashboard </a> </div> <div class="row"> <!-- Import Card --> <div class="col-lg-6"> <div
class="card shadow mb-4"> <div class="card-header py-3"> <h6 class="m-0 font-weight-bold
text-primary">Import Employee Data</h6> </div> <div class="card-body"> <p>Upload an Excel file
(.xlsx, .xls) or CSV file containing employee data.</p> <form method="POST" action="{{
url_for('employee.import_export') }}" enctype="multipart/form-data"> <input type="hidden"
name="action" value="import"> <div class="mb-3"> <label for="file" class="form-label">Select
File</label> <input class="form-control" type="file" id="file" name="file" accept=".xlsx,.xls,.csv"
required> </div> <div class="alert alert-info"> <h6 class="alert-heading">Expected Column
Headers</h6> <p class="mb-0"> The following columns are required: "Full Name", "Joining Date",
"Benzyl", "Role", "Skill", "Team", "Manager Name", "Grade", "Designation", "Location" </p> </div>
<button type="submit" class="btn btn-success"> <i class="fas fa-file-import mr-1"></i> Import Data
</button> </form> </div> </div> </div> <!-- Export Card --> <div class="col-lg-6"> <div class="card
shadow mb-4"> <div class="card-header py-3"> <h6 class="m-0 font-weight-bold text-primary">Export
Employee Data</h6> </div> <div class="card-body"> <p>Export all employee data to an Excel file
(.xlsx) for backup or reporting.</p> <form method="POST" action="{{
url_for('employee.import_export') }}"> <input type="hidden" name="action" value="export"> <div
class="mb-3"> <div class="alert alert-info"> <p class="mb-0"> This will export all employee data
including names, roles, skills, teams, etc. The exported file will not include feedback records.
</p> </div> </div> <button type="submit" class="btn btn-info"> <i class="fas fa-file-export
mr-1"></i> Export Data </button> </form> </div> </div> </div> </div> </div> {% endblock %}
```

./templates/login.html

./templates/login.html

```
{% extends "base.html" %} {% block title %}Login - Employee Management System{% endblock %} {%
block content %} <div class="container"> <div class="row justify-content-center"> <div
class="col-xl-10 col-lg-12 col-md-9"> <div class="card o-hidden border-0 shadow-lg my-5"> <div
class="card-body p-0"> <div class="row"> <div class="col-lg-6 d-none d-lg-block bg-primary"> <div
class="p-5 text-center text-white d-flex align-items-center justify-content-center h-100"> <div>
<h1 class="display-4 mb-4"> <i class="fas fa-users fa-2x mb-3"></i><br> Employee Management System
</h1> <p class="lead">Manage your team, track feedback, and improve performance</p> </div> </div>
</div> <div class="col-lg-6"> <div class="p-5"> <div class="text-center"> <h1 class="h4
mb-4">Welcome Back!</h1> </div> <form class="user" method="POST" action="{{ url_for('auth.login')
}}"> <div class="form-group mb-3"> <input type="text" class="form-control form-control-user"
id="username" name="username" placeholder="Username" required> </div> <div class="form-group mb-3">
<input type="password" class="form-control form-control-user" id="password" name="password"
placeholder="Password" required> </div> <button type="submit" class="btn btn-primary btn-user
btn-block w-100"> Login </button> </form> <hr> <div class="text-center"> <p>Not registered? Contact
your system administrator.</p> </div> <!-- Link to initialize users if none exist --> <div
class="text-center"> <a href="{{ url_for('auth.create_initial_users') }}" class="small"> Initialize
system </a> </div> </div> </div> </div> </div> </div> </div> </div> </div> {% endblock %}
```

./templates/profile.html

./templates/profile.html

```
{% extends "base.html" %} {% block title %}Profile - Employee Management System{% endblock %} {%
block content %} <div class="container-fluid"> <div class="d-flex justify-content-between
align-items-center mb-4"> <h1 class="h3 text-gray-800">Your Profile</h1> <a href="{{
url_for('employee.dashboard') }}" class="btn btn-secondary"> <i class="fas fa-arrow-left mr-1"></i>
Back to Dashboard </a> </div> <!-- User Profile Card --> <div class="row"> <div class="col-lg-12">
<div class="card shadow mb-4"> <div class="card-header py-3"> <h6 class="m-0 font-weight-bold
text-primary">User Information</h6> </div> <div class="card-body"> <div class="row"> <div
class="col-md-3 text-center"> <div class="employee-avatar"> {{ current_user.username[0]|upper }}
</div> <h4 class="mt-3">{{ current_user.username }}</h4> <p class="text-muted">{{ 'Manager' if
current_user.is_manager else 'Employee' }}</p> </div> <div class="col-md-9"> <div
class="employee-info"> <div class="employee-info-item"> <div
class="employee-info-label">Username</div> <div class="employee-info-value">{{
current_user.username }}</div> </div> <div class="employee-info-item"> <div
class="employee-info-label">Email</div> <div class="employee-info-value">{{ current_user.email
}}</div> </div> <div class="employee-info-item"> <div class="employee-info-label">Role</div> <div
class="employee-info-value">{{ 'Manager' if current_user.is_manager else 'Employee' }}</div> </div>
{% if employee %} <div class="employee-info-item"> <div class="employee-info-label">Employee
ID</div> <div class="employee-info-value">{{ employee.benzyl }}</div> </div> <div
class="employee-info-item"> <div class="employee-info-label">Department</div> <div
class="employee-info-value">{{ employee.team }}</div> </div> <div class="employee-info-item"> <div
class="employee-info-label">Position</div> <div class="employee-info-value">{{ employee.designation
}}</div> </div> {% endif %} </div> </div> </div> <hr> <div class="text-center mt-3"> <a href="{{
url_for('auth.logout') }}" class="btn btn-danger"> <i class="fas fa-sign-out-alt mr-1"></i> Logout
</a> </div> </div> </div> </div> </div> </div> {% endblock %}
```

templates/base.html

templates/base.html

```
<!DOCTYPE html> <html lang="en" data-bs-theme="dark"> <head> <meta charset="UTF-8"> <meta
name="viewport" content="width=device-width, initial-scale=1.0"> <title>{% block title %}Employee
Management System{% endblock %}</title> <!-- Bootstrap CSS from Replit CDN --> <link
rel="stylesheet" href="https://cdn.replit.com/agent/bootstrap-agent-dark-theme.min.css"> <!-- Font
Awesome --> <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css"> <!-- DataTables
CSS --> <link rel="stylesheet"
href="https://cdn.datatables.net/1.13.4/css/dataTables.bootstrap5.min.css"> <link rel="stylesheet"
href="https://cdn.datatables.net/buttons/2.3.6/css/buttons.bootstrap5.min.css"> <!-- Custom CSS -->
<link rel="stylesheet" href="{{ url_for('static', filename='css/custom.css') }}"> {% block
extra_css %}{% endblock %} </head> <body> <!-- Sidebar --> {% if current_user.is_authenticated %}
<div class="sidebar bg-dark"> <div class="sidebar-brand d-flex align-items-center
justify-content-center p-3"> <div class="sidebar-brand-icon"> <i class="fas fa-users"></i> </div>
<div class="sidebar-brand-text mx-3">EMS</div> </div> <hr class="sidebar-divider"> <ul class="nav
flex-column"> <li class="nav-item"> <a class="nav-link {% if request.endpoint and request.endpoint
== 'employee.dashboard' %}active{% endif %}" href="{{ url_for('employee.dashboard') }}"> <i
class="fas fa-fw fa-tachometer-alt"></i> <span>Dashboard</span> </a> </li> <li class="nav-item"> <a
class="nav-link {% if request.endpoint and 'employee_list' in request.endpoint %}active{% endif %}"
href="{{ url_for('employee.employee_list') }}"> <i class="fas fa-fw fa-users"></i>
<span>Employees</span> </a> </li> <li class="nav-item"> <a class="nav-link {% if request.endpoint
and 'feedback' in request.endpoint %}active{% endif %}" href="{{ url_for('feedback.feedback_list')
}}"> <i class="fas fa-fw fa-comments"></i> <span>Feedback</span> </a> </li> {% if
current_user.is_manager %} <li class="nav-item"> <a class="nav-link {% if request.endpoint and
'import_export' in request.endpoint %}active{% endif %}" href="{{ url_for('employee.import_export')
}}"> <i class="fas fa-fw fa-file-excel"></i> <span>Import/Export</span> </a> </li> <li
class="nav-item"> <a class="nav-link" href="{{ url_for('docs.generate_and_download_docs') }}"> <i
class="fas fa-fw fa-file-pdf"></i> <span>Download Documentation</span> </a> </li> {% endif %} <hr
class="sidebar-divider"> <li class="nav-item"> <a class="nav-link" href="{{ url_for('auth.profile')
}}"> <i class="fas fa-fw fa-user"></i> <span>Profile</span> </a> </li> <li class="nav-item"> <a
class="nav-link" href="{{ url_for('auth.logout') }}"> <i class="fas fa-fw fa-sign-out-alt"></i>
<span>Logout</span> </a> </li> </ul> </div> {% endif %} <!-- Main Content --> <div
class="content-wrapper {% if not current_user.is_authenticated %}ml-0{% endif %}"> {% if
current_user.is_authenticated %} <!-- Topbar --> <nav class="navbar navbar-expand navbar-dark
bg-dark mb-4"> <button id="sidebarToggle" class="btn btn-link d-md-none rounded-circle mr-3"> <i
class="fa fa-bars"></i> </button> <div class="d-none d-md-inline-block ms-auto form-inline mr-0
navbar-search"> <div class="input-group"> <input type="text" id="searchInput" class="form-control
bg-light border-0 small" placeholder="Search..." aria-label="Search"> <div
class="input-group-append"> <button class="btn btn-primary" type="button"> <i class="fas fa-search
fa-sm"></i> </button> </div> </div> </div> <ul class="navbar-nav ml-auto ml-md-0"> <li
class="nav-item dropdown no-arrow"> <a class="nav-link dropdown-toggle" href="#" id="userDropdown"
role="button" data-bs-toggle="dropdown" aria-haspopup="true" aria-expanded="false"> <span
class="mr-2 d-none d-lg-inline text-gray-600 small">{{ current_user.username }}</span> <i
class="fas fa-user-circle fa-fw"></i> </a> <div class="dropdown-menu dropdown-menu-right shadow
animated--grow-in" aria-labelledby="userDropdown"> <a class="dropdown-item" href="{{
url_for('auth.profile') }}"> <i class="fas fa-user fa-sm fa-fw mr-2 text-gray-400"></i> Profile
</a> <div class="dropdown-divider"></div> <a class="dropdown-item" href="{{ url_for('auth.logout')
}}"> <i class="fas fa-sign-out-alt fa-sm fa-fw mr-2 text-gray-400"></i> Logout </a> </div> </li>
</ul> </nav> {% endif %} <!-- Flash Messages --> <div class="container-fluid"> {% with messages =
get_flashed_messages(with_categories=true) %} {% if messages %} {% for category, message in
messages %} <div class="alert alert-{{ category }} alert-dismissible fade show" role="alert"> {{
message }} <button type="button" class="btn-close" data-bs-dismiss="alert"
aria-label="Close"></button> </div> {% endfor %} {% endif %} {% endwith %} <!-- Page Content --> {%
block content %}{% endblock %} </div> </div> <!-- Bootstrap JS Bundle --> <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script> <!--
jQuery --> <script src="https://code.jquery.com/jquery-3.6.4.min.js"></script> <!-- DataTables JS
--> <script src="https://cdn.datatables.net/1.13.4/js/jquery.dataTables.min.js"></script> <script
src="https://cdn.datatables.net/1.13.4/js/dataTables.bootstrap5.min.js"></script> <script
src="https://cdn.datatables.net/buttons/2.3.6/js/dataTables.buttons.min.js"></script> <script
src="https://cdn.datatables.net/buttons/2.3.6/js/buttons.bootstrap5.min.js"></script> <script
src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.10.1/jszip.min.js"></script> <script
src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/pdfmake.min.js"></script> <script
src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/vfs_fonts.js"></script> <script
src="https://cdn.datatables.net/buttons/2.3.6/js/buttons.html5.min.js"></script> <script
src="https://cdn.datatables.net/buttons/2.3.6/js/buttons.print.min.js"></script> <!-- Chart.js -->
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script> <!-- Custom JS --> <script src="{{
url_for('static', filename='js/main.js') }}"></script> <script src="{{ url_for('static',
filename='js/datatables-config.js') }}"></script> {% block extra_js %}{% endblock %} </body>
</html>
```

templates/dashboard.html

templates/dashboard.html

```
{% extends "base.html" %} {% block title %}Dashboard - Employee Management System{% endblock %} {%
block content %} <div class="container-fluid"> <h1 class="h3 mb-4 text-gray-800">Dashboard</h1>
<!-- Content Row - Stats --> <div class="row"> <!-- Total Employees Card --> <div class="col-xl-3
col-md-6 mb-4"> <div class="card dashboard-card border-left-primary shadow h-100 py-2"> <div
class="card-body"> <div class="row no-gutters align-items-center"> <div class="col mr-2"> <div
class="text-xs font-weight-bold text-primary text-uppercase mb-1 dashboard-card-header"> Total
Employees </div> <div class="h5 mb-0 font-weight-bold text-gray-800 dashboard-card-body"> {{
total_employees }} </div> </div> <div class="col-auto"> <i class="fas fa-users fa-2x
text-gray-300"></i> </div> </div> </div> </div> </div> {% if is_manager %} <!-- Managed Employees
Card --> <div class="col-xl-3 col-md-6 mb-4"> <div class="card dashboard-card border-left-success
shadow h-100 py-2"> <div class="card-body"> <div class="row no-gutters align-items-center"> <div
class="col mr-2"> <div class="text-xs font-weight-bold text-success text-uppercase mb-1
dashboard-card-header"> Team Members </div> <div class="h5 mb-0 font-weight-bold text-gray-800
dashboard-card-body"> {{ managed_employees }} </div> </div> <div class="col-auto"> <i class="fas
fa-user-friends fa-2x text-gray-300"></i> </div> </div> </div> </div> </div> <!-- Recent Feedback
Card --> <div class="col-xl-3 col-md-6 mb-4"> <div class="card dashboard-card border-left-info
shadow h-100 py-2"> <div class="card-body"> <div class="row no-gutters align-items-center"> <div
class="col mr-2"> <div class="text-xs font-weight-bold text-info text-uppercase mb-1
dashboard-card-header"> Recent Feedback </div> <div class="h5 mb-0 font-weight-bold text-gray-800
dashboard-card-body"> {{ recent_feedback|length }} </div> </div> <div class="col-auto"> <i
class="fas fa-comments fa-2x text-gray-300"></i> </div> </div> </div> </div> </div> <!-- Pending
Feedback Card --> <div class="col-xl-3 col-md-6 mb-4"> <div class="card dashboard-card
border-left-warning shadow h-100 py-2"> <div class="card-body"> <div class="row no-gutters
align-items-center"> <div class="col mr-2"> <div class="text-xs font-weight-bold text-warning
text-uppercase mb-1 dashboard-card-header"> Current Quarter </div> <div class="h5 mb-0
font-weight-bold text-gray-800 dashboard-card-body"> Q{{ ((now.month - 1) // 3) + 1 }}-{{ now.year
}} </div> </div> <div class="col-auto"> <i class="fas fa-calendar fa-2x text-gray-300"></i> </div>
</div> </div> </div> </div> {% else %} <!-- Employee Info Cards --> {% if employee %} <div
class="col-xl-3 col-md-6 mb-4"> <div class="card dashboard-card border-left-success shadow h-100
py-2"> <div class="card-body"> <div class="row no-gutters align-items-center"> <div class="col
mr-2"> <div class="text-xs font-weight-bold text-success text-uppercase mb-1
dashboard-card-header"> Your Role </div> <div class="h5 mb-0 font-weight-bold text-gray-800
dashboard-card-body"> {{ employee.role }} </div> </div> <div class="col-auto"> <i class="fas
fa-user-tag fa-2x text-gray-300"></i> </div> </div> </div> </div> </div> <div class="col-xl-3
col-md-6 mb-4"> <div class="card dashboard-card border-left-info shadow h-100 py-2"> <div
class="card-body"> <div class="row no-gutters align-items-center"> <div class="col mr-2"> <div
class="text-xs font-weight-bold text-info text-uppercase mb-1 dashboard-card-header"> Team </div>
<div class="h5 mb-0 font-weight-bold text-gray-800 dashboard-card-body"> {{ employee.team }} </div>
</div> <div class="col-auto"> <i class="fas fa-users-cog fa-2x text-gray-300"></i> </div> </div>
</div> </div> </div> <div class="col-xl-3 col-md-6 mb-4"> <div class="card dashboard-card
border-left-warning shadow h-100 py-2"> <div class="card-body"> <div class="row no-gutters
align-items-center"> <div class="col mr-2"> <div class="text-xs font-weight-bold text-warning
text-uppercase mb-1 dashboard-card-header"> Recent Feedback </div> <div class="h5 mb-0
font-weight-bold text-gray-800 dashboard-card-body"> {{ recent_feedback|length }} </div> </div>
<div class="col-auto"> <i class="fas fa-comments fa-2x text-gray-300"></i> </div> </div> </div>
</div> </div> {% endif %} {% endif %} </div> <!-- Content Row - Details --> <div class="row"> {% if
is_manager %} <!-- Team Skills Distribution --> <div class="col-lg-6 mb-4"> <div class="card shadow
mb-4"> <div class="card-header py-3"> <h6 class="m-0 font-weight-bold text-primary">Team Skills
Distribution</h6> </div> <div class="card-body"> <div class="chart-container"> <canvas
id="teamSkillsChart"></canvas> </div> <!-- Hidden data for the chart --> {% for skill, count in
team_skills %} <div class="skill-stat d-none" data-skill="{{ skill }}" data-count="{{ count
}}"></div> {% endfor %} </div> </div> </div> <!-- Recent Feedback List --> <div class="col-lg-6
mb-4"> <div class="card shadow mb-4"> <div class="card-header py-3"> <h6 class="m-0
font-weight-bold text-primary">Recent Feedback Provided</h6> </div> <div class="card-body"> {% if
recent_feedback %} <div class="table-responsive"> <table class="table table-bordered" width="100%"
cellspacing="0"> <thead> <tr> <th>Employee</th> <th>Rating</th> <th>Quarter</th> <th>Date</th>
</tr> </thead> <tbody> {% for feedback in recent_feedback %} <tr> <td>{{
feedback.employee.full_name }}</td> <td> <div class="feedback-rating"> {% for i in
range(feedback.rating) %} <i class="fas fa-star feedback-star"></i> {% endfor %} {% for i in
range(5 - feedback.rating) %} <i class="far fa-star feedback-star"></i> {% endfor %} </div> </td>
<td>{{ feedback.quarter }}</td> <td>{{ feedback.feedback_date.strftime('%Y-%m-%d') }}</td> </tr> {%
endfor %} </tbody> </table> </div> <a href="{{ url_for('feedback.feedback_list') }}" class="btn
btn-primary btn-sm"> View All Feedback </a> {% else %} <p>No recent feedback provided.</p> {% endif
%} </div> </div> </div> <!-- Quick Actions --> <div class="col-lg-12 mb-4"> <div class="card shadow
mb-4"> <div class="card-header py-3"> <h6 class="m-0 font-weight-bold text-primary">Quick
Actions</h6> </div> <div class="card-body"> <div class="row"> <div class="col-md-4 mb-3"> <a
href="{{ url_for('employee.create_employee') }}" class="btn btn-primary w-100"> <i class="fas
fa-user-plus mr-2"></i> Add New Employee </a> </div> <div class="col-md-4 mb-3"> <a href="{{
url_for('employee.import_export') }}" class="btn btn-success w-100"> <i class="fas fa-file-import
mr-2"></i> Import Employee Data </a> </div> <div class="col-md-4 mb-3"> <a href="{{
```

```
url_for('employee.import_export') }}?action=export" class="btn btn-info w-100"> <i class="fas
fa-file-export mr-2"></i> Export Employee Data </a> </div> </div> </div> </div> </div> {% else %}
<!-- Employee Recent Feedback --> {% if employee and recent_feedback %} <div class="col-lg-12
mb-4"> <div class="card shadow mb-4"> <div class="card-header py-3"> <h6 class="m-0
font-weight-bold text-primary">Your Recent Feedback</h6> </div> <div class="card-body"> {% for
feedback in recent_feedback %} <div class="feedback-card"> <div class="card-header"> <div> <span
class="feedback-quarter">{{ feedback.quarter }}</span> <span class="feedback-date">{{
feedback.feedback_date.strftime('%Y-%m-%d') }}</span> </div> <div class="feedback-rating"> {% for i
in range(feedback.rating) %} <i class="fas fa-star feedback-star"></i> {% endfor %} {% for i in
range(5 - feedback.rating) %} <i class="far fa-star feedback-star"></i> {% endfor %} </div> </div>
<div class="card-body"> <p>{{ feedback.feedback_text }}</p> <small class="text-muted">Provided by:
{{ feedback.provided_by.username }}</small> </div> </div> {% endfor %} <a href="{{
url_for('feedback.feedback_list') }}" class="btn btn-primary btn-sm mt-3"> View All Feedback </a>
</div> </div> </div> {% elif employee %} <div class="col-lg-12 mb-4"> <div class="card shadow
mb-4"> <div class="card-header py-3"> <h6 class="m-0 font-weight-bold text-primary">Your
Feedback</h6> </div> <div class="card-body"> <p>No feedback has been provided yet.</p> </div>
</div> </div> {% endif %} <!-- Employee Profile --> {% if employee %} <div class="col-lg-12 mb-4">
<div class="card shadow mb-4"> <div class="card-header py-3"> <h6 class="m-0 font-weight-bold
text-primary">Your Profile</h6> </div> <div class="card-body"> <div class="row"> <div
class="col-md-3 text-center"> <div class="employee-avatar"> {{ employee.full_name[0] }} </div> <h4
class="mt-3">{{ employee.full_name }}</h4> <p class="text-muted">{{ employee.designation }}</p>
</div> <div class="col-md-9"> <div class="employee-info"> <div class="employee-info-item"> <div
class="employee-info-label">Employee ID</div> <div class="employee-info-value">{{ employee.benzyl
}}</div> </div> <div class="employee-info-item"> <div class="employee-info-label">Joining
Date</div> <div class="employee-info-value">{{ employee.joining_date }}</div> </div> <div
class="employee-info-item"> <div class="employee-info-label">Role</div> <div
class="employee-info-value">{{ employee.role }}</div> </div> <div class="employee-info-item"> <div
class="employee-info-label">Skill</div> <div class="employee-info-value">{{ employee.skill }}</div>
</div> <div class="employee-info-item"> <div class="employee-info-label">Team</div> <div
class="employee-info-value">{{ employee.team }}</div> </div> <div class="employee-info-item"> <div
class="employee-info-label">Grade</div> <div class="employee-info-value">{{ employee.grade }}</div>
</div> <div class="employee-info-item"> <div class="employee-info-label">Location</div> <div
class="employee-info-value">{{ employee.location }}</div> </div> </div> </div> </div> </div> </div>
</div> {% endif %} {% endif %} </div> </div> {% endblock %} {% block extra_js %} <script> // Use
the current date for the dashboard document.addEventListener('DOMContentLoaded', function() { //
Set the current date object for the template window.now = new Date(); }); </script> {% endblock %}
```

templates/employee_detail.html

templates/employee_detail.html

```
{% extends "base.html" %} {% block title %}{{ employee.full_name }} - Employee Management System{%
endblock %} {% block content %} <div class="container-fluid"> <div class="d-flex
justify-content-between align-items-center mb-4"> <h1 class="h3 text-gray-800">Employee
Details</h1> <div> <a href="{{ url_for('employee.employee_list') }}" class="btn btn-secondary"> <i
class="fas fa-arrow-left mr-1"></i> Back to List </a> {% if is_manager and employee.manager_id ==
current_user.id %} <a href="{{ url_for('employee.edit_employee', employee_id=employee.id) }}"
class="btn btn-warning"> <i class="fas fa-edit mr-1"></i> Edit </a> {% endif %} </div> </div> <!--
Employee Profile Card --> <div class="row mb-4"> <div class="col-lg-12"> <div class="card shadow">
<div class="card-header py-3 d-flex justify-content-between align-items-center"> <h6 class="m-0
font-weight-bold text-primary">Employee Profile</h6> {% if is_manager and employee.manager_id ==
current_user.id %} <a href="{{ url_for('feedback.create_feedback', employee_id=employee.id) }}"
class="btn btn-success btn-sm"> <i class="fas fa-comment-dots mr-1"></i> Provide Feedback </a> {%
endif %} </div> <div class="card-body"> <div class="row"> <div class="col-md-3 text-center"> <div
class="employee-avatar"> {{ employee.full_name[0] }} </div> <h4 class="mt-3">{{ employee.full_name
}}</h4> <p class="text-muted">{{ employee.designation }}</p> </div> <div class="col-md-9"> <div
class="employee-info"> <div class="employee-info-item"> <div class="employee-info-label">Employee
ID</div> <div class="employee-info-value">{{ employee.benzyl }}</div> </div> <div
class="employee-info-item"> <div class="employee-info-label">Joining Date</div> <div
class="employee-info-value">{{ employee.joining_date }}</div> </div> <div
class="employee-info-item"> <div class="employee-info-label">Role</div> <div
class="employee-info-value">{{ employee.role }}</div> </div> <div class="employee-info-item"> <div
class="employee-info-label">Skill</div> <div class="employee-info-value">{{ employee.skill }}</div>
</div> <div class="employee-info-item"> <div class="employee-info-label">Team</div> <div
class="employee-info-value">{{ employee.team }}</div> </div> <div class="employee-info-item"> <div
class="employee-info-label">Grade</div> <div class="employee-info-value">{{ employee.grade }}</div>
</div> <div class="employee-info-item"> <div class="employee-info-label">Location</div> <div
class="employee-info-value">{{ employee.location }}</div> </div> <div class="employee-info-item">
<div class="employee-info-label">Manager</div> <div class="employee-info-value"> {% if manager %}
{{ manager.username }} {% else %} Not assigned {% endif %} </div> </div> </div> </div> </div>
</div> </div> </div> </div> <!-- Feedback History Card --> <div class="row"> <div
class="col-lg-12"> <div class="card shadow mb-4"> <div class="card-header py-3"> <h6 class="m-0
font-weight-bold text-primary">Feedback History</h6> </div> <div class="card-body"> {% if feedback
%} <div class="table-responsive"> <table class="table table-bordered" id="feedbackTable"
width="100%" cellspacing="0"> <thead> <tr> <th>Quarter</th> <th>Rating</th> <th>Feedback</th>
<th>Provided By</th> <th>Date</th> {% if is_manager %} <th>Actions</th> {% endif %} </tr> </thead>
<tbody> {% for fb in feedback %} <tr> <td>{{ fb.quarter }}</td> <td> <div class="feedback-rating">
{% for i in range(fb.rating) %} <i class="fas fa-star feedback-star"></i> {% endfor %} {% for i in
range(5 - fb.rating) %} <i class="far fa-star feedback-star"></i> {% endfor %} </div> </td> <td>{{
fb.feedback_text }}</td> <td>{{ fb.provided_by.username }}</td> <td>{{
fb.feedback_date.strftime('%Y-%m-%d') }}</td> {% if is_manager and fb.provided_by_id ==
current_user.id %} <td> <div class="btn-group" role="group"> <a href="{{
url_for('feedback.edit_feedback', feedback_id=fb.id) }}" class="btn btn-warning btn-sm"
data-bs-toggle="tooltip" title="Edit Feedback"> <i class="fas fa-edit"></i> </a> <button
type="button" class="btn btn-danger btn-sm confirm-delete" data-bs-toggle="modal"
data-bs-target="#deleteFeedbackModal{{ fb.id }}" title="Delete Feedback"> <i class="fas
fa-trash"></i> </button> <!-- Delete Modal --> <div class="modal fade" id="deleteFeedbackModal{{
fb.id }}" tabindex="-1" aria-labelledby="deleteFeedbackModalLabel" aria-hidden="true"> <div
class="modal-dialog"> <div class="modal-content"> <div class="modal-header"> <h5
class="modal-title" id="deleteFeedbackModalLabel"> Confirm Delete </h5> <button type="button"
class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button> </div> <div
class="modal-body"> Are you sure you want to delete this feedback? This action cannot be undone.
</div> <div class="modal-footer"> <button type="button" class="btn btn-secondary"
data-bs-dismiss="modal">Cancel</button> <form action="{{ url_for('feedback.delete_feedback',
feedback_id=fb.id) }}" method="POST"> <button type="submit" class="btn btn-danger">Delete</button>
</form> </div> </div> </div> </div> </div> </td> {% elif is_manager %} <td>-</td> {% endif %} </tr>
{% endfor %} </tbody> </table> </div> {% else %} <div class="text-center py-5"> <i class="fas
fa-comments fa-4x mb-3 text-gray-300"></i> <p class="lead text-gray-500">No feedback records
found</p> {% if is_manager and employee.manager_id == current_user.id %} <a href="{{
url_for('feedback.create_feedback', employee_id=employee.id) }}" class="btn btn-success mt-3"> <i
class="fas fa-comment-dots mr-1"></i> Provide First Feedback </a> {% endif %} </div> {% endif %}
</div> </div> </div> </div> </div> {% endblock %}
```

templates/employee_form.html

templates/employee_form.html

```
{% extends "base.html" %} {% block title %} {% if action == 'create' %} Add New Employee - Employee
Management System {% else %} Edit Employee - Employee Management System {% endif %} {% endblock %}
{% block content %} <div class="container-fluid"> <div class="d-flex justify-content-between
align-items-center mb-4"> <h1 class="h3 text-gray-800"> {% if action == 'create' %} Add New
Employee {% else %} Edit Employee {% endif %} </h1> <a href="{{ url_for('employee.employee_list')
}}" class="btn btn-secondary"> <i class="fas fa-arrow-left mr-1"></i> Back to List </a> </div> <!--
Employee Form Card --> <div class="row"> <div class="col-lg-12"> <div class="card shadow mb-4">
<div class="card-header py-3"> <h6 class="m-0 font-weight-bold text-primary"> {% if action ==
'create' %} Employee Information {% else %} Update Employee Information {% endif %} </h6> </div>
<div class="card-body"> <form method="POST" action="{% if action == 'create' %}{{
url_for('employee.create_employee') }}{% else %}{{ url_for('employee.edit_employee',
employee_id=employee.id) }}{% endif %}"> <div class="row"> <div class="col-md-6 mb-3"> <label
for="full_name" class="form-label">Full Name</label> <input type="text" class="form-control"
id="full_name" name="full_name" value="{{ employee.full_name if employee else '' }}" required>
</div> <div class="col-md-6 mb-3"> <label for="benzyl" class="form-label">Employee ID
(Benzyl)</label> <input type="text" class="form-control" id="benzyl" name="benzyl" value="{{
employee.benzyl if employee else '' }}" required> </div> </div> <div class="row"> <div
class="col-md-6 mb-3"> <label for="joining_date" class="form-label">Joining Date</label> <input
type="date" class="form-control" id="joining_date" name="joining_date" value="{{
employee.joining_date if employee else '' }}" required> </div> <div class="col-md-6 mb-3"> <label
for="role" class="form-label">Role</label> <input type="text" class="form-control" id="role"
name="role" value="{{ employee.role if employee else '' }}" required> </div> </div> <div
class="row"> <div class="col-md-6 mb-3"> <label for="skill" class="form-label">Skill</label> <input
type="text" class="form-control" id="skill" name="skill" value="{{ employee.skill if employee else
'' }}" required> </div> <div class="col-md-6 mb-3"> <label for="team"
class="form-label">Team</label> <input type="text" class="form-control" id="team" name="team"
value="{{ employee.team if employee else '' }}" required> </div> </div> <div class="row"> <div
class="col-md-6 mb-3"> <label for="grade" class="form-label">Grade</label> <input type="text"
class="form-control" id="grade" name="grade" value="{{ employee.grade if employee else '' }}"
required> </div> <div class="col-md-6 mb-3"> <label for="designation"
class="form-label">Designation</label> <input type="text" class="form-control" id="designation"
name="designation" value="{{ employee.designation if employee else '' }}" required> </div> </div>
<div class="row"> <div class="col-md-6 mb-3"> <label for="location"
class="form-label">Location</label> <input type="text" class="form-control" id="location"
name="location" value="{{ employee.location if employee else '' }}" required> </div> <div
class="col-md-6 mb-3"> <label for="manager_id" class="form-label">Manager</label> <select
class="form-select" id="manager_id" name="manager_id"> <option value="">Select Manager</option> {%
for manager in managers %} <option value="{{ manager.id }}" {% if employee and employee.manager_id
== manager.id %}selected{% endif %} {% if not employee and manager.id == current_user.id
%}selected{% endif %}> {{ manager.username }} </option> {% endfor %} </select> </div> </div> <div
class="mt-4"> <button type="submit" class="btn btn-primary"> {% if action == 'create' %} <i
class="fas fa-plus-circle mr-1"></i> Create Employee {% else %} <i class="fas fa-save mr-1"></i>
Update Employee {% endif %} </button> <a href="{{ url_for('employee.employee_list') }}" class="btn
btn-secondary"> <i class="fas fa-times-circle mr-1"></i> Cancel </a> </div> </form> </div> </div>
</div> </div> </div> {% endblock %}
```

templates/employee_list.html

templates/employee_list.html

```
{% extends "base.html" %} {% block title %}Employees - Employee Management System{% endblock %} {%
block content %} <div class="container-fluid"> <div class="d-flex justify-content-between
align-items-center mb-4"> <h1 class="h3 text-gray-800">Employees</h1> {% if is_manager %} <a
href="{{ url_for('employee.create_employee') }}" class="btn btn-primary"> <i class="fas
fa-user-plus mr-1"></i> Add Employee </a> {% endif %} </div> <!-- Employee List Table --> <div
class="card shadow mb-4"> <div class="card-header py-3"> <h6 class="m-0 font-weight-bold
text-primary">Employee Directory</h6> </div> <div class="card-body"> {% if employees %} <div
class="table-responsive"> <table class="table table-bordered" id="employeeTable" width="100%"
cellspacing="0"> <thead> <tr> <th>Full Name</th> <th>Benzyl</th> <th>Role</th> <th>Skill</th>
<th>Team</th> <th>Grade</th> <th>Location</th> <th>Actions</th> </tr> </thead> <tbody> {% for
employee in employees %} <tr {% if employee.manager_id == current_user.id %}class="table-primary"{%
endif %}> <td>{{ employee.full_name }}</td> <td>{{ employee.benzyl }}</td> <td>{{ employee.role
}}</td> <td>{{ employee.skill }}</td> <td>{{ employee.team }}</td> <td>{{ employee.grade }}</td>
<td>{{ employee.location }}</td> <td> <div class="btn-group" role="group"> <a href="{{
url_for('employee.employee_detail', employee_id=employee.id) }}" class="btn btn-info btn-sm"
data-bs-toggle="tooltip" title="View Details"> <i class="fas fa-eye"></i> </a> {% if is_manager and
employee.manager_id == current_user.id %} <a href="{{ url_for('feedback.create_feedback',
employee_id=employee.id) }}" class="btn btn-success btn-sm" data-bs-toggle="tooltip" title="Provide
Feedback"> <i class="fas fa-comment-dots"></i> </a> <a href="{{ url_for('employee.edit_employee',
employee_id=employee.id) }}" class="btn btn-warning btn-sm" data-bs-toggle="tooltip" title="Edit
Employee"> <i class="fas fa-edit"></i> </a> <button type="button" class="btn btn-danger btn-sm
confirm-delete" data-bs-toggle="modal" data-bs-target="#deleteModal{{ employee.id }}" title="Delete
Employee"> <i class="fas fa-trash"></i> </button> <!-- Delete Modal --> <div class="modal fade"
id="deleteModal{{ employee.id }}" tabindex="-1" aria-labelledby="deleteModalLabel"
aria-hidden="true"> <div class="modal-dialog"> <div class="modal-content"> <div
class="modal-header"> <h5 class="modal-title" id="deleteModalLabel"> Confirm Delete </h5> <button
type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button> </div> <div
class="modal-body"> Are you sure you want to delete {{ employee.full_name }}? This action cannot be
undone. </div> <div class="modal-footer"> <button type="button" class="btn btn-secondary"
data-bs-dismiss="modal">Cancel</button> <form action="{{ url_for('employee.delete_employee',
employee_id=employee.id) }}" method="POST"> <button type="submit" class="btn
btn-danger">Delete</button> </form> </div> </div> </div> </div> {% endif %} </div> </td> </tr> {%
endfor %} </tbody> </table> </div> {% else %} <div class="text-center py-5"> <i class="fas fa-users
fa-4x mb-3 text-gray-300"></i> <p class="lead text-gray-500">No employees found</p> {% if
is_manager %} <a href="{{ url_for('employee.create_employee') }}" class="btn btn-primary mt-3"> <i
class="fas fa-user-plus mr-1"></i> Add First Employee </a> {% endif %} </div> {% endif %} </div>
</div> </div> {% endblock %}
```

templates/errors/404.html

templates/errors/404.html

```
<!DOCTYPE html> <html lang="en" data-bs-theme="dark"> <head> <meta charset="UTF-8"> <meta
name="viewport" content="width=device-width, initial-scale=1.0"> <title>404 Not Found - Employee
Management System</title> <!-- Bootstrap CSS from Replit CDN --> <link rel="stylesheet"
href="https://cdn.replit.com/agent/bootstrap-agent-dark-theme.min.css"> <!-- Font Awesome --> <link
rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
</head> <body> <div class="container mt-5"> <div class="text-center"> <div
class="display-1">404</div> <p class="lead mb-5">Page Not Found</p> <p class="text-muted mb-4">It
looks like you found a glitch in the matrix...</p> <a href="/" class="btn btn-primary">&larr; Back
to Home</a> </div> </div> <!-- Bootstrap JS Bundle --> <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body> </html>
```

templates/errors/500.html

templates/errors/500.html

```
{% extends "base.html" %} {% block title %}500 Server Error - Employee Management System{% endblock
%} {% block content %} <div class="container-fluid"> <div class="text-center"> <div class="error
mx-auto" data-text="500">500</div> <p class="lead text-gray-800 mb-5">Internal Server Error</p> <p
class="text-gray-500 mb-0">Something went wrong. Please try again later.</p> <a href="{{
url_for('employee.dashboard') }}">&larr; Back to Dashboard</a> </div> </div> {% endblock %}
```

templates/errors/base_error.html

> templates/errors/base_error.html

```
<!DOCTYPE html> <html lang="en" data-bs-theme="dark"> <head> <meta charset="UTF-8"> <meta
name="viewport" content="width=device-width, initial-scale=1.0"> <title>{% block title %}Error -
Employee Management System{% endblock %}</title> <!-- Bootstrap CSS from Replit CDN --> <link
rel="stylesheet" href="https://cdn.replit.com/agent/bootstrap-agent-dark-theme.min.css"> <!-- Font
Awesome --> <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css"> </head> <body>
<div class="container mt-5"> {% block content %}{% endblock %} </div> <!-- Bootstrap JS Bundle -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body> </html>
```

templates/feedback_form.html

templates/feedback_form.html

```
{% extends "base.html" %} {% block title %} {% if action == 'create' %} Provide Feedback - Employee
Management System {% else %} Edit Feedback - Employee Management System {% endif %} {% endblock %}
{% block content %} <div class="container-fluid"> <div class="d-flex justify-content-between
align-items-center mb-4"> <h1 class="h3 text-gray-800"> {% if action == 'create' %} Provide
Feedback for {{ employee.full_name }} {% else %} Edit Feedback for {{ employee.full_name }} {%
endif %} </h1> <a href="{{ url_for('feedback.employee_feedback', employee_id=employee.id) }}"
class="btn btn-secondary"> <i class="fas fa-arrow-left mr-1"></i> Back to Feedback </a> </div> <!--
Feedback Form Card --> <div class="row"> <div class="col-lg-12"> <div class="card shadow mb-4">
<div class="card-header py-3"> <h6 class="m-0 font-weight-bold text-primary"> {% if action ==
'create' %} New Feedback {% else %} Update Feedback {% endif %} </h6> </div> <div
class="card-body"> <form method="POST" action="{% if action == 'create' %}{{
url_for('feedback.create_feedback', employee_id=employee.id) }}{% else %}{{
url_for('feedback.edit_feedback', feedback_id=feedback.id) }}{% endif %}"> <div class="row"> <div
class="col-md-6 mb-3"> <label for="quarter" class="form-label">Quarter</label> <select
class="form-select" id="quarter" name="quarter" required> <option value="">Select Quarter</option>
{% set years = [2023, 2024, 2025] %} {% for year in years %} {% for q in range(1, 5) %} <option
value="Q{{ q }}-{{ year }}" {% if (feedback and feedback.quarter == 'Q' ~ q ~ '-' ~ year) or (not
feedback and current_quarter == 'Q' ~ q ~ '-' ~ year) %} selected {% endif %}> Q{{ q }}-{{ year }}
</option> {% endfor %} {% endfor %} </select> </div> <div class="col-md-6 mb-3"> <label
for="rating" class="form-label">Rating</label> <input type="hidden" id="rating" name="rating"
value="{{ feedback.rating if feedback else 0 }}" required> <div class="rating-container"> <div
class="rating-stars"> {% for i in range(1, 6) %} <i class="fas fa-star rating-star {% if feedback
and i <= feedback.rating %}rated{% endif %}" data-value="{{ i }}"></i> {% endfor %} </div> <span
class="ms-2 rating-label"> {% if feedback %} {{ feedback.rating }} out of 5 {% else %} Select a
rating {% endif %} </span> </div> </div> </div> <div class="mb-3"> <label for="feedback_text"
class="form-label">Feedback</label> <textarea class="form-control" id="feedback_text"
name="feedback_text" rows="6" required>{{ feedback.feedback_text if feedback else '' }}</textarea>
</div> <div class="mt-4"> <button type="submit" class="btn btn-primary"> {% if action == 'create'
%} <i class="fas fa-paper-plane mr-1"></i> Submit Feedback {% else %} <i class="fas fa-save
mr-1"></i> Update Feedback {% endif %} </button> <a href="{{ url_for('feedback.employee_feedback',
employee_id=employee.id) }}" class="btn btn-secondary"> <i class="fas fa-times-circle mr-1"></i>
Cancel </a> </div> </form> </div> </div> </div> </div> </div> {% endblock %} {% block extra_js %}
<script> document.addEventListener('DOMContentLoaded', function() { const ratingInput =
document.getElementById('rating'); const ratingStars = document.querySelectorAll('.rating-star');
const ratingLabel = document.querySelector('.rating-label'); ratingStars.forEach(star => {
star.addEventListener('click', function() { const value = this.getAttribute('data-value');
ratingInput.value = value; // Update star display ratingStars.forEach(s => { const starValue =
s.getAttribute('data-value'); if (starValue <= value) { s.classList.add('rated'); } else {
s.classList.remove('rated'); } }); // Update label ratingLabel.textContent = `${value} out of 5`;
}); }); }); </script> {% endblock %}
```

templates/feedback_list.html

templates/feedback_list.html

```
{% extends "base.html" %} {% block title %}Feedback - Employee Management System{% endblock %} {%
block content %} <div class="container-fluid"> <div class="d-flex justify-content-between
align-items-center mb-4"> <h1 class="h3 text-gray-800"> {% if view_type == 'provided' %} Feedback
Provided {% elif view_type == 'received' %} Feedback Received {% elif view_type == 'for_employee'
%} Feedback for {{ employee.full_name }} {% endif %} </h1> {% if is_manager and view_type ==
'for_employee' and employee.manager_id == current_user.id %} <a href="{{
url_for('feedback.create_feedback', employee_id=employee.id) }}" class="btn btn-success"> <i
class="fas fa-comment-dots mr-1"></i> Provide Feedback </a> {% endif %} </div> <!-- Feedback List
Card --> <div class="card shadow mb-4"> <div class="card-header py-3"> <h6 class="m-0
font-weight-bold text-primary"> {% if view_type == 'provided' %} Feedback You Have Provided {% elif
view_type == 'received' %} Feedback You Have Received {% elif view_type == 'for_employee' %} All
Feedback for {{ employee.full_name }} {% endif %} </h6> </div> <div class="card-body"> {% if
feedback %} <div class="table-responsive"> <table class="table table-bordered" id="feedbackTable"
width="100%" cellspacing="0"> <thead> <tr> {% if view_type != 'for_employee' %} <th>Employee</th>
{% endif %} <th>Quarter</th> <th>Rating</th> <th>Feedback</th> {% if view_type == 'received' or
view_type == 'for_employee' %} <th>Provided By</th> {% endif %} <th>Date</th> {% if is_manager and
view_type != 'received' %} <th>Actions</th> {% endif %} </tr> </thead> <tbody> {% for fb in
feedback %} <tr> {% if view_type != 'for_employee' %} <td> <a href="{{
url_for('employee.employee_detail', employee_id=fb.employee.id) }}"> {{ fb.employee.full_name }}
</a> </td> {% endif %} <td>{{ fb.quarter }}</td> <td> <div class="feedback-rating"> {% for i in
range(fb.rating) %} <i class="fas fa-star feedback-star"></i> {% endfor %} {% for i in range(5 -
fb.rating) %} <i class="far fa-star feedback-star"></i> {% endfor %} </div> </td> <td>{{
fb.feedback_text }}</td> {% if view_type == 'received' or view_type == 'for_employee' %} <td>{{
fb.provided_by.username }}</td> {% endif %} <td>{{ fb.feedback_date.strftime('%Y-%m-%d') }}</td> {%
if is_manager and view_type != 'received' and fb.provided_by_id == current_user.id %} <td> <div
class="btn-group" role="group"> <a href="{{ url_for('feedback.edit_feedback', feedback_id=fb.id)
}}" class="btn btn-warning btn-sm" data-bs-toggle="tooltip" title="Edit Feedback"> <i class="fas
fa-edit"></i> </a> <button type="button" class="btn btn-danger btn-sm confirm-delete"
data-bs-toggle="modal" data-bs-target="#deleteFeedbackModal{{ fb.id }}" title="Delete Feedback"> <i
class="fas fa-trash"></i> </button> <!-- Delete Modal --> <div class="modal fade"
id="deleteFeedbackModal{{ fb.id }}" tabindex="-1" aria-labelledby="deleteFeedbackModalLabel"
aria-hidden="true"> <div class="modal-dialog"> <div class="modal-content"> <div
class="modal-header"> <h5 class="modal-title" id="deleteFeedbackModalLabel"> Confirm Delete </h5>
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button> </div>
<div class="modal-body"> Are you sure you want to delete this feedback? This action cannot be
undone. </div> <div class="modal-footer"> <button type="button" class="btn btn-secondary"
data-bs-dismiss="modal">Cancel</button> <form action="{{ url_for('feedback.delete_feedback',
feedback_id=fb.id) }}" method="POST"> <button type="submit" class="btn btn-danger">Delete</button>
</form> </div> </div> </div> </div> </div> </td> {% elif is_manager and view_type != 'received' %}
<td>-</td> {% endif %} </tr> {% endfor %} </tbody> </table> </div> {% else %} <div
class="text-center py-5"> <i class="fas fa-comments fa-4x mb-3 text-gray-300"></i> <p class="lead
text-gray-500">No feedback records found</p> {% if is_manager and view_type == 'for_employee' and
employee.manager_id == current_user.id %} <a href="{{ url_for('feedback.create_feedback',
employee_id=employee.id) }}" class="btn btn-success mt-3"> <i class="fas fa-comment-dots mr-1"></i>
Provide First Feedback </a> {% endif %} </div> {% endif %} </div> </div> </div> {% endblock %}
```

templates/import_export.html

templates/import_export.html

```
{% extends "base.html" %} {% block title %}Import/Export - Employee Management System{% endblock %}
{% block content %} <div class="container-fluid"> <div class="d-flex justify-content-between
align-items-center mb-4"> <h1 class="h3 text-gray-800">Import/Export Employee Data</h1> <a href="{{
url_for('employee.dashboard') }}" class="btn btn-secondary"> <i class="fas fa-arrow-left mr-1"></i>
Back to Dashboard </a> </div> <div class="row"> <!-- Import Card --> <div class="col-lg-6"> <div
class="card shadow mb-4"> <div class="card-header py-3"> <h6 class="m-0 font-weight-bold
text-primary">Import Employee Data</h6> </div> <div class="card-body"> <p>Upload an Excel file
(.xlsx, .xls) or CSV file containing employee data.</p> <form method="POST" action="{{
url_for('employee.import_export') }}" enctype="multipart/form-data"> <input type="hidden"
name="action" value="import"> <div class="mb-3"> <label for="file" class="form-label">Select
File</label> <input class="form-control" type="file" id="file" name="file" accept=".xlsx,.xls,.csv"
required> </div> <div class="alert alert-info"> <h6 class="alert-heading">Expected Column
Headers</h6> <p class="mb-0"> The following columns are required: "Full Name", "Joining Date",
"Benzyl", "Role", "Skill", "Team", "Manager Name", "Grade", "Designation", "Location" </p> </div>
<button type="submit" class="btn btn-success"> <i class="fas fa-file-import mr-1"></i> Import Data
</button> </form> </div> </div> </div> <!-- Export Card --> <div class="col-lg-6"> <div class="card
shadow mb-4"> <div class="card-header py-3"> <h6 class="m-0 font-weight-bold text-primary">Export
Employee Data</h6> </div> <div class="card-body"> <p>Export all employee data to an Excel file
(.xlsx) for backup or reporting.</p> <form method="POST" action="{{
url_for('employee.import_export') }}"> <input type="hidden" name="action" value="export"> <div
class="mb-3"> <div class="alert alert-info"> <p class="mb-0"> This will export all employee data
including names, roles, skills, teams, etc. The exported file will not include feedback records.
</p> </div> </div> <button type="submit" class="btn btn-info"> <i class="fas fa-file-export
mr-1"></i> Export Data </button> </form> </div> </div> </div> </div> </div> {% endblock %}
```

templates/login.html

templates/login.html

```
{% extends "base.html" %} {% block title %}Login - Employee Management System{% endblock %} {%
block content %} <div class="container"> <div class="row justify-content-center"> <div
class="col-xl-10 col-lg-12 col-md-9"> <div class="card o-hidden border-0 shadow-lg my-5"> <div
class="card-body p-0"> <div class="row"> <div class="col-lg-6 d-none d-lg-block bg-primary"> <div
class="p-5 text-center text-white d-flex align-items-center justify-content-center h-100"> <div>
<h1 class="display-4 mb-4"> <i class="fas fa-users fa-2x mb-3"></i><br> Employee Management System
</h1> <p class="lead">Manage your team, track feedback, and improve performance</p> </div> </div>
</div> <div class="col-lg-6"> <div class="p-5"> <div class="text-center"> <h1 class="h4
mb-4">Welcome Back!</h1> </div> <form class="user" method="POST" action="{{ url_for('auth.login')
}}"> <div class="form-group mb-3"> <input type="text" class="form-control form-control-user"
id="username" name="username" placeholder="Username" required> </div> <div class="form-group mb-3">
<input type="password" class="form-control form-control-user" id="password" name="password"
placeholder="Password" required> </div> <button type="submit" class="btn btn-primary btn-user
btn-block w-100"> Login </button> </form> <hr> <div class="text-center"> <p>Not registered? Contact
your system administrator.</p> </div> <!-- Link to initialize users if none exist --> <div
class="text-center"> <a href="{{ url_for('auth.create_initial_users') }}" class="small"> Initialize
system </a> </div> </div> </div> </div> </div> </div> </div> </div> </div> {% endblock %}
```

templates/profile.html

templates/profile.html

```
{% extends "base.html" %} {% block title %}Profile - Employee Management System{% endblock %} {%
block content %} <div class="container-fluid"> <div class="d-flex justify-content-between
align-items-center mb-4"> <h1 class="h3 text-gray-800">Your Profile</h1> <a href="{{
url_for('employee.dashboard') }}" class="btn btn-secondary"> <i class="fas fa-arrow-left mr-1"></i>
Back to Dashboard </a> </div> <!-- User Profile Card --> <div class="row"> <div class="col-lg-12">
<div class="card shadow mb-4"> <div class="card-header py-3"> <h6 class="m-0 font-weight-bold
text-primary">User Information</h6> </div> <div class="card-body"> <div class="row"> <div
class="col-md-3 text-center"> <div class="employee-avatar"> {{ current_user.username[0]|upper }}
</div> <h4 class="mt-3">{{ current_user.username }}</h4> <p class="text-muted">{{ 'Manager' if
current_user.is_manager else 'Employee' }}</p> </div> <div class="col-md-9"> <div
class="employee-info"> <div class="employee-info-item"> <div
class="employee-info-label">Username</div> <div class="employee-info-value">{{
current_user.username }}</div> </div> <div class="employee-info-item"> <div
class="employee-info-label">Email</div> <div class="employee-info-value">{{ current_user.email
}}</div> </div> <div class="employee-info-item"> <div class="employee-info-label">Role</div> <div
class="employee-info-value">{{ 'Manager' if current_user.is_manager else 'Employee' }}</div> </div>
{% if employee %} <div class="employee-info-item"> <div class="employee-info-label">Employee
ID</div> <div class="employee-info-value">{{ employee.benzyl }}</div> </div> <div
class="employee-info-item"> <div class="employee-info-label">Department</div> <div
class="employee-info-value">{{ employee.team }}</div> </div> <div class="employee-info-item"> <div
class="employee-info-label">Position</div> <div class="employee-info-value">{{ employee.designation
}}</div> </div> {% endif %} </div> </div> </div> <hr> <div class="text-center mt-3"> <a href="{{
url_for('auth.logout') }}" class="btn btn-danger"> <i class="fas fa-sign-out-alt mr-1"></i> Logout
</a> </div> </div> </div> </div> </div> </div> {% endblock %}
```

./static/js/datatables-config.js

./static/js/datatables-config.js

```
document.addEventListener('DOMContentLoaded', function() { // Initialize DataTables for employee
list const employeeTable = document.getElementById('employeeTable'); if (employeeTable) { new
DataTable('#employeeTable', { responsive: true, dom: 'Bfrtip', buttons: [ 'copy', 'csv', 'excel',
'pdf', 'print' ], order: [[0, 'asc']], // Sort by name by default pageLength: 10, language: {
search: "Filter:", lengthMenu: "Show _MENU_ employees per page", info: "Showing _START_ to _END_ of
_TOTAL_ employees", infoEmpty: "Showing 0 to 0 of 0 employees", infoFiltered: "(filtered from _MAX_
total employees)" } }); } // Initialize DataTables for feedback list const feedbackTable =
document.getElementById('feedbackTable'); if (feedbackTable) { new DataTable('#feedbackTable', {
responsive: true, order: [[3, 'desc']], // Sort by date by default pageLength: 10, language: {
search: "Filter:", lengthMenu: "Show _MENU_ feedback entries per page", info: "Showing _START_ to
_END_ of _TOTAL_ feedback entries", infoEmpty: "Showing 0 to 0 of 0 feedback entries",
infoFiltered: "(filtered from _MAX_ total feedback entries)" } }); } // For dashboard stats,
initialize simple charts const teamSkillsChart = document.getElementById('teamSkillsChart'); if
(teamSkillsChart) { // Extract data from the page const labels = []; const data = [];
document.querySelectorAll('.skill-stat').forEach(stat => {
labels.push(stat.getAttribute('data-skill')); data.push(parseInt(stat.getAttribute('data-count')));
}); // Create chart new Chart(teamSkillsChart, { type: 'doughnut', data: { labels: labels,
datasets: [{ data: data, backgroundColor: [ '#4e73df', '#1cc88a', '#36b9cc', '#f6c23e', '#e74a3b',
'#5a5c69', '#858796', '#2e59d9', '#17a673', '#2c9faf' ], hoverBackgroundColor: [ '#2e59d9',
'#17a673', '#2c9faf', '#f6c23e', '#e74a3b', '#5a5c69', '#858796', '#2e59d9', '#17a673', '#2c9faf'
], hoverBorderColor: "rgba(234, 236, 244, 1)", }] }, options: { responsive: true,
maintainAspectRatio: false, plugins: { legend: { position: 'bottom', }, title: { display: true,
text: 'Team Skills Distribution' } } } }); } // Feedback trends chart (for managers) const
feedbackTrendsChart = document.getElementById('feedbackTrendsChart'); if (feedbackTrendsChart) { //
Extract data from the page (this would be populated by the backend) const quarters = []; const
avgRatings = []; document.querySelectorAll('.feedback-trend').forEach(trend => {
quarters.push(trend.getAttribute('data-quarter'));
avgRatings.push(parseFloat(trend.getAttribute('data-avg-rating'))); }); // Create chart new
Chart(feedbackTrendsChart, { type: 'line', data: { labels: quarters, datasets: [{ label: 'Average
Rating', data: avgRatings, lineTension: 0.3, backgroundColor: "rgba(78, 115, 223, 0.05)",
borderColor: "rgba(78, 115, 223, 1)", pointRadius: 3, pointBackgroundColor: "rgba(78, 115, 223,
1)", pointBorderColor: "rgba(78, 115, 223, 1)", pointHoverRadius: 5, pointHoverBackgroundColor:
"rgba(78, 115, 223, 1)", pointHoverBorderColor: "rgba(78, 115, 223, 1)", pointHitRadius: 10,
pointBorderWidth: 2, fill: true }] }, options: { responsive: true, maintainAspectRatio: false,
scales: { y: { min: 0, max: 5, ticks: { stepSize: 1 } } }, plugins: { legend: { display: false },
title: { display: true, text: 'Feedback Rating Trends by Quarter' } } } }); } });
```

./static/js/main.js

./static/js/main.js

```
// Wait for the DOM to be fully loaded document.addEventListener('DOMContentLoaded', function() {
// Initialize tooltips var tooltipTriggerList =
[].slice.call(document.querySelectorAll('[data-bs-toggle="tooltip"]')); var tooltipList =
tooltipTriggerList.map(function(tooltipTriggerEl) { return new bootstrap.Tooltip(tooltipTriggerEl);
}); // Initialize popovers var popoverTriggerList =
[].slice.call(document.querySelectorAll('[data-bs-toggle="popover"]')); var popoverList =
popoverTriggerList.map(function(popoverTriggerEl) { return new bootstrap.Popover(popoverTriggerEl);
}); // Handle flash message dismissal setTimeout(function() { let alerts =
document.querySelectorAll('.alert-auto-dismiss'); alerts.forEach(function(alert) { let bsAlert =
new bootstrap.Alert(alert); bsAlert.close(); }); }, 5000); // Confirm delete operations
document.querySelectorAll('.confirm-delete').forEach(function(button) {
button.addEventListener('click', function(event) { if (!confirm('Are you sure you want to delete
this item? This action cannot be undone.')) { event.preventDefault(); } }); }); // Star rating
system for feedback const ratingInputs = document.querySelectorAll('.rating-input'); const
ratingStars = document.querySelectorAll('.rating-star'); if (ratingStars.length > 0) {
ratingStars.forEach(star => { star.addEventListener('click', function() { const value =
this.getAttribute('data-value'); document.getElementById('rating').value = value; // Update star
display ratingStars.forEach(s => { const starValue = s.getAttribute('data-value'); if (starValue <=
value) { s.classList.add('rated'); } else { s.classList.remove('rated'); } }); }); }); } // Toggle
side navigation for mobile const sidebarToggle = document.getElementById('sidebarToggle'); if
(sidebarToggle) { sidebarToggle.addEventListener('click', function() {
document.body.classList.toggle('sidebar-toggled');
document.querySelector('.sidebar').classList.toggle('toggled'); }); } // Handle search
functionality const searchInput = document.getElementById('searchInput'); if (searchInput) {
searchInput.addEventListener('keyup', function() { const searchTerm = this.value.toLowerCase();
const table = document.querySelector('table'); const rows = table.querySelectorAll('tbody tr');
rows.forEach(row => { const text = row.textContent.toLowerCase(); if(text.includes(searchTerm)) {
row.style.display = ''; } else { row.style.display = 'none'; } }); }); } });
```

static/js/datatables-config.js

static/js/datatables-config.js

```
document.addEventListener('DOMContentLoaded', function() { // Initialize DataTables for employee
list const employeeTable = document.getElementById('employeeTable'); if (employeeTable) { new
DataTable('#employeeTable', { responsive: true, dom: 'Bfrtip', buttons: [ 'copy', 'csv', 'excel',
'pdf', 'print' ], order: [[0, 'asc']], // Sort by name by default pageLength: 10, language: {
search: "Filter:", lengthMenu: "Show _MENU_ employees per page", info: "Showing _START_ to _END_ of
_TOTAL_ employees", infoEmpty: "Showing 0 to 0 of 0 employees", infoFiltered: "(filtered from _MAX_
total employees)" } }); } // Initialize DataTables for feedback list const feedbackTable =
document.getElementById('feedbackTable'); if (feedbackTable) { new DataTable('#feedbackTable', {
responsive: true, order: [[3, 'desc']], // Sort by date by default pageLength: 10, language: {
search: "Filter:", lengthMenu: "Show _MENU_ feedback entries per page", info: "Showing _START_ to
_END_ of _TOTAL_ feedback entries", infoEmpty: "Showing 0 to 0 of 0 feedback entries",
infoFiltered: "(filtered from _MAX_ total feedback entries)" } }); } // For dashboard stats,
initialize simple charts const teamSkillsChart = document.getElementById('teamSkillsChart'); if
(teamSkillsChart) { // Extract data from the page const labels = []; const data = [];
document.querySelectorAll('.skill-stat').forEach(stat => {
labels.push(stat.getAttribute('data-skill')); data.push(parseInt(stat.getAttribute('data-count')));
}); // Create chart new Chart(teamSkillsChart, { type: 'doughnut', data: { labels: labels,
datasets: [{ data: data, backgroundColor: [ '#4e73df', '#1cc88a', '#36b9cc', '#f6c23e', '#e74a3b',
'#5a5c69', '#858796', '#2e59d9', '#17a673', '#2c9faf' ], hoverBackgroundColor: [ '#2e59d9',
'#17a673', '#2c9faf', '#f6c23e', '#e74a3b', '#5a5c69', '#858796', '#2e59d9', '#17a673', '#2c9faf'
], hoverBorderColor: "rgba(234, 236, 244, 1)", }] }, options: { responsive: true,
maintainAspectRatio: false, plugins: { legend: { position: 'bottom', }, title: { display: true,
text: 'Team Skills Distribution' } } } }); } // Feedback trends chart (for managers) const
feedbackTrendsChart = document.getElementById('feedbackTrendsChart'); if (feedbackTrendsChart) { //
Extract data from the page (this would be populated by the backend) const quarters = []; const
avgRatings = []; document.querySelectorAll('.feedback-trend').forEach(trend => {
quarters.push(trend.getAttribute('data-quarter'));
avgRatings.push(parseFloat(trend.getAttribute('data-avg-rating'))); }); // Create chart new
Chart(feedbackTrendsChart, { type: 'line', data: { labels: quarters, datasets: [{ label: 'Average
Rating', data: avgRatings, lineTension: 0.3, backgroundColor: "rgba(78, 115, 223, 0.05)",
borderColor: "rgba(78, 115, 223, 1)", pointRadius: 3, pointBackgroundColor: "rgba(78, 115, 223,
1)", pointBorderColor: "rgba(78, 115, 223, 1)", pointHoverRadius: 5, pointHoverBackgroundColor:
"rgba(78, 115, 223, 1)", pointHoverBorderColor: "rgba(78, 115, 223, 1)", pointHitRadius: 10,
pointBorderWidth: 2, fill: true }] }, options: { responsive: true, maintainAspectRatio: false,
scales: { y: { min: 0, max: 5, ticks: { stepSize: 1 } } }, plugins: { legend: { display: false },
title: { display: true, text: 'Feedback Rating Trends by Quarter' } } } }); } });
```

static/js/main.js

static/js/main.js

```
// Wait for the DOM to be fully loaded document.addEventListener('DOMContentLoaded', function() {
// Initialize tooltips var tooltipTriggerList =
[].slice.call(document.querySelectorAll('[data-bs-toggle="tooltip"]')); var tooltipList =
tooltipTriggerList.map(function(tooltipTriggerEl) { return new bootstrap.Tooltip(tooltipTriggerEl);
}); // Initialize popovers var popoverTriggerList =
[].slice.call(document.querySelectorAll('[data-bs-toggle="popover"]')); var popoverList =
popoverTriggerList.map(function(popoverTriggerEl) { return new bootstrap.Popover(popoverTriggerEl);
}); // Handle flash message dismissal setTimeout(function() { let alerts =
document.querySelectorAll('.alert-auto-dismiss'); alerts.forEach(function(alert) { let bsAlert =
new bootstrap.Alert(alert); bsAlert.close(); }); }, 5000); // Confirm delete operations
document.querySelectorAll('.confirm-delete').forEach(function(button) {
button.addEventListener('click', function(event) { if (!confirm('Are you sure you want to delete
this item? This action cannot be undone.')) { event.preventDefault(); } }); }); // Star rating
system for feedback const ratingInputs = document.querySelectorAll('.rating-input'); const
ratingStars = document.querySelectorAll('.rating-star'); if (ratingStars.length > 0) {
ratingStars.forEach(star => { star.addEventListener('click', function() { const value =
this.getAttribute('data-value'); document.getElementById('rating').value = value; // Update star
display ratingStars.forEach(s => { const starValue = s.getAttribute('data-value'); if (starValue <=
value) { s.classList.add('rated'); } else { s.classList.remove('rated'); } }); }); }); } // Toggle
side navigation for mobile const sidebarToggle = document.getElementById('sidebarToggle'); if
(sidebarToggle) { sidebarToggle.addEventListener('click', function() {
document.body.classList.toggle('sidebar-toggled');
document.querySelector('.sidebar').classList.toggle('toggled'); }); } // Handle search
functionality const searchInput = document.getElementById('searchInput'); if (searchInput) {
searchInput.addEventListener('keyup', function() { const searchTerm = this.value.toLowerCase();
const table = document.querySelector('table'); const rows = table.querySelectorAll('tbody tr');
rows.forEach(row => { const text = row.textContent.toLowerCase(); if(text.includes(searchTerm)) {
row.style.display = ''; } else { row.style.display = 'none'; } }); }); } });
```

./app.py

> ./app.py

```
import os import logging from flask import Flask, render_template from flask_sqlalchemy import
SQLAlchemy from sqlalchemy.orm import DeclarativeBase from flask_login import LoginManager from
werkzeug.middleware.proxy_fix import ProxyFix # Configure logging
logging.basicConfig(level=logging.DEBUG) logger = logging.getLogger(__name__) # Setup base for
SQLAlchemy models class Base(DeclarativeBase): pass # Initialize extensions db =
SQLAlchemy(model_class=Base) login_manager = LoginManager() # Create the Flask application app =
Flask(__name__) app.secret_key = os.environ.get("SESSION_SECRET", "dev-secret-key") app.wsgi_app =
ProxyFix(app.wsgi_app, x_proto=1, x_host=1) # Configure the database
app.config["SQLALCHEMY_DATABASE_URI"] = os.environ.get( "DATABASE_URL",
"sqlite:///employee_management.db" ) app.config["SQLALCHEMY_TRACK_MODIFICATIONS"] = False
app.config["SQLALCHEMY_ENGINE_OPTIONS"] = { "pool_recycle": 300, "pool_pre_ping": True, } #
Initialize extensions with the app db.init_app(app) login_manager.init_app(app)
login_manager.login_view = 'auth.login' login_manager.login_message_category = 'info' # Handle
error pages @app.errorhandler(404) def page_not_found(e): return
render_template('errors/404.html'), 404 @app.errorhandler(500) def internal_server_error(e): return
render_template('errors/500.html'), 500 # Setup login manager user loader
@login_manager.user_loader def load_user(user_id): from models import User # Import here to avoid
circular imports return User.query.get(int(user_id)) # Create all tables and import routes with
app.app_context(): # Import models (moved to inside app context to avoid circular imports) import
models # Create database tables db.create_all() # Import and register blueprints from auth import
auth_bp from employee import employee_bp from feedback import feedback_bp from docs import docs_bp
app.register_blueprint(auth_bp) app.register_blueprint(employee_bp)
app.register_blueprint(feedback_bp) app.register_blueprint(docs_bp) logger.info("Application
initialized successfully")
```

./auth.py

./auth.py

```
from flask import Blueprint, render_template, redirect, url_for, flash, request from flask_login
import login_user, logout_user, login_required, current_user from werkzeug.security import
generate_password_hash from app import db from models import User, Employee import logging logger =
logging.getLogger(__name__) auth_bp = Blueprint('auth', __name__, url_prefix='/auth')
@auth_bp.route('/login', methods=['GET', 'POST']) def login(): if current_user.is_authenticated:
return redirect(url_for('employee.dashboard')) if request.method == 'POST': username =
request.form.get('username') password = request.form.get('password') # Validate form inputs if not
username or not password: flash('Please provide both username and password', 'danger') return
render_template('login.html') # Find user by username user =
User.query.filter_by(username=username).first() # Check if user exists and password is correct if
user and user.check_password(password): login_user(user) logger.info(f"User {username} logged in
successfully") # Redirect to the page user was trying to access or dashboard next_page =
request.args.get('next') if next_page: return redirect(next_page) return
redirect(url_for('employee.dashboard')) else: flash('Invalid username or password. Please try
again.', 'danger') logger.warning(f"Failed login attempt for username: {username}") return
render_template('login.html') @auth_bp.route('/logout') @login_required def logout(): logout_user()
flash('You have been logged out successfully.', 'success') return redirect(url_for('auth.login'))
@auth_bp.route('/profile') @login_required def profile(): # Find employee record related to this
user if it exists employee = Employee.query.filter_by(user_id=current_user.id).first() return
render_template('profile.html', employee=employee) # Admin function to create initial users
@auth_bp.route('/create_initial_users') def create_initial_users(): # Check if any users exist
already if User.query.count() > 0: flash('Initial users already created', 'info') return
redirect(url_for('auth.login')) try: # Create admin user admin = User( username='admin',
email='admin@example.com', is_manager=True ) admin.set_password('admin123') # Create a regular
employee user employee = User( username='employee', email='employee@example.com', is_manager=False
) employee.set_password('employee123') db.session.add(admin) db.session.add(employee)
db.session.commit() flash('Initial users created successfully. You can now login with username
"admin" and password "admin123".', 'success') logger.info("Initial users created") except Exception
as e: db.session.rollback() logger.error(f"Error creating initial users: {str(e)}") flash(f'Error
creating users: {str(e)}', 'danger') return redirect(url_for('auth.login'))
```

./create_functionality_docs.py

> ./create_functionality_docs.py

```
""" Documentation Generator for EMS Application This script generates comprehensive documentation
of the EMS application including functionality details, class descriptions, and screenshots. """
import os import inspect import importlib from datetime import datetime from xhtml2pdf import pisa
from flask import Flask import models import auth import employee import feedback import utils def
get_module_details(module): """Extract class and function details from a module""" module_name =
module.__name__ module_doc = module.__doc__ or "No documentation available." classes = [] functions
= [] for name, obj in inspect.getmembers(module): # Get classes if inspect.isclass(obj) and
obj.__module__ == module_name: class_info = { 'name': name, 'doc': obj.__doc__ or "No documentation
available.", 'methods': [] } for method_name, method in inspect.getmembers(obj): if
method_name.startswith('_'): continue if inspect.isfunction(method) or inspect.ismethod(method):
method_info = { 'name': method_name, 'doc': method.__doc__ or "No documentation available.",
'signature': str(inspect.signature(method)) } class_info['methods'].append(method_info)
classes.append(class_info) # Get functions elif inspect.isfunction(obj) and obj.__module__ ==
module_name: function_info = { 'name': name, 'doc': obj.__doc__ or "No documentation available.",
'signature': str(inspect.signature(obj)) } functions.append(function_info) return { 'name':
module_name, 'doc': module_doc, 'classes': classes, 'functions': functions } def
generate_html_documentation(): """Generate HTML documentation for all modules""" modules = [models,
auth, employee, feedback, utils] modules_data = [get_module_details(module) for module in modules]
# Start building HTML html = f""" <!DOCTYPE html> <html> <head> <meta charset="UTF-8"> <title>EMS
Application Functionality Documentation</title> <style> body {{ font-family: Arial, sans-serif;
margin: 20px; }} h1 {{ color: #2c3e50; text-align: center; }} h2 {{ color: #3498db; border-bottom:
1px solid #3498db; padding-bottom: 5px; }} h3 {{ color: #2980b9; }} h4 {{ color: #16a085; }} pre {{
background-color: #f8f9fa; padding: 10px; border-radius: 4px; }} .signature {{ font-family:
monospace; background-color: #f0f0f0; padding: 3px; }} .module {{ margin-bottom: 30px; border: 1px
solid #ddd; padding: 15px; border-radius: 5px; }} .class {{ margin-bottom: 20px; background-color:
#f8f9fa; padding: 10px; border-radius: 4px; }} .method {{ margin-left: 20px; margin-bottom: 10px;
}} .function {{ margin-bottom: 15px; background-color: #f8f9fa; padding: 10px; border-radius: 4px;
}} .toc {{ background-color: #f8f9fa; padding: 10px; margin-bottom: 20px; }} .toc ul {{
padding-left: 20px; }} .screenshot {{ width: 95%; border: 1px solid #ddd; margin: 10px 0; }}
.page-break {{ page-break-before: always; }} @page {{ size: A4; margin: 2cm; }} </style> </head>
<body> <h1>Employee Management System Functionality Documentation</h1> <p style="text-align:
center;">Generated on: {datetime.now().strftime('%Y-%m-%d %H:%M:%S')}</p> <div class="toc">
<h2>Table of Contents</h2> <ul> <li><a href="#overview">Application Overview</a></li> <li><a
href="#functionality">Key Functionalities</a> <ul> <li><a href="#auth">Authentication
System</a></li> <li><a href="#employee-management">Employee Management</a></li> <li><a
href="#feedback-system">Feedback System</a></li> <li><a href="#import-export">Import/Export
Functionality</a></li> </ul> </li> <li><a href="#class-details">Class Details</a> <ul> """ # Add
module links to TOC for module_data in modules_data: html += f'<li><a
href="#{module_data["name"]}">{module_data["name"]}</a></li>\n' html += """ </ul> </li> <li><a
href="#usage-guide">Usage Guide</a></li> </ul> </div> <!-- Application Overview --> <div
class="page-break"></div> <h2 id="overview">Application Overview</h2> <p> The Employee Management
System (EMS) is a comprehensive web application built with Flask and PostgreSQL that allows line
managers to maintain employee details and provide feedback throughout the year. The system provides
full CRUD operations for employee data management, feedback tracking, and reporting. </p> <p> The
application is organized with a modular structure, separating concerns into different components:
authentication, employee management, feedback system, and utility functions. </p> <!-- Key
Functionalities with Screenshots --> <div class="page-break"></div> <h2 id="functionality">Key
Functionalities</h2> <h3 id="auth">Authentication System</h3> <p>The authentication system handles
user login, logout, and profile management with the following features:</p> <ul> <li>User login
with username and password</li> <li>Role-based access control (manager vs. employee)</li>
<li>Profile management</li> <li>Password security using bcrypt hashing</li> </ul> <p>Key
Classes/Functions:</p> <ul> <li><code>User</code> model in models.py</li> <li><code>login()</code>,
<code>logout()</code>, <code>profile()</code> functions in auth.py</li> </ul> <h4>Authentication
Screenshots</h4> <p><strong>Login Page</strong></p> <div class="screenshot-placeholder"> [Login
Page Screenshot] </div> <p><strong>User Profile Page</strong></p> <div
class="screenshot-placeholder"> [Profile Page Screenshot] </div> <h3
id="employee-management">Employee Management</h3> <p>The employee management module handles all
operations related to employee data:</p> <ul> <li>Dashboard overview of team and statistics</li>
<li>Employee listing with sorting and filtering</li> <li>Employee details view</li> <li>Create,
update, and delete employee records</li> </ul> <p>Key Classes/Functions:</p> <ul>
<li><code>Employee</code> model in models.py</li> <li><code>dashboard()</code>,
<code>employee_list()</code>, <code>create_employee()</code> functions in employee.py</li> </ul>
<h4>Employee Management Screenshots</h4> <p><strong>Dashboard</strong></p> <div
class="screenshot-placeholder"> [Dashboard Screenshot] </div> <p><strong>Employee List</strong></p>
<div class="screenshot-placeholder"> [Employee List Screenshot] </div> <p><strong>Employee
Details</strong></p> <div class="screenshot-placeholder"> [Employee Details Screenshot] </div> <h3
id="feedback-system">Feedback System</h3> <p>The feedback system allows managers to provide
structured feedback to their reportees:</p> <ul> <li>Create quarterly feedback with ratings and
comments</li> <li>View feedback history for each employee</li> <li>Edit and delete feedback</li>
</ul> <p>Key Classes/Functions:</p> <ul> <li><code>Feedback</code> model in models.py</li>
```

```
<li><code>feedback_list()</code>, <code>create_feedback()</code> functions in feedback.py</li>
</ul> <h4>Feedback System Screenshots</h4> <p><strong>Feedback List</strong></p> <div
class="screenshot-placeholder"> [Feedback List Screenshot] </div> <p><strong>Create
Feedback</strong></p> <div class="screenshot-placeholder"> [Create Feedback Screenshot] </div> <h3
id="import-export">Import/Export Functionality</h3> <p>The import/export module enables data
exchange with Excel spreadsheets:</p> <ul> <li>Import employee data from Excel
(manager_toolkit.xlsx)</li> <li>Export employee data to Excel format</li> </ul> <p>Key
Classes/Functions:</p> <ul> <li><code>import_export()</code> function in employee.py</li>
<li><code>export_employees_to_excel()</code> function in utils.py</li> </ul> <h4>Import/Export
Screenshots</h4> <p><strong>Import/Export Interface</strong></p> <div
class="screenshot-placeholder"> [Import/Export Screenshot] </div> <!-- Class Details Section -->
<div class="page-break"></div> <h2 id="class-details">Class Details</h2> """ # Add module details
for module_data in modules_data: html += f""" <div class="module page-break"
id="{module_data['name']}"> <h3>{module_data['name']}.py</h3> <pre>{module_data['doc']}</pre>
<h4>Classes:</h4> """ if not module_data['classes']: html += "<p>No classes defined in this
module.</p>" else: for class_data in module_data['classes']: html += f""" <div class="class">
<h4>{class_data['name']}</h4> <pre>{class_data['doc']}</pre> <h5>Methods:</h5> """ if not
class_data['methods']: html += "<p>No methods defined in this class.</p>" else: for method in
class_data['methods']: html += f""" <div class="method"> <h6>{method['name']}<span
class="signature">{method['signature']}</span></h6> <pre>{method['doc']}</pre> </div> """ html +=
"</div>" html += """ <h4>Functions:</h4> """ if not module_data['functions']: html += "<p>No
functions defined in this module.</p>" else: for function in module_data['functions']: html += f"""
<div class="function"> <h5>{function['name']}<span
class="signature">{function['signature']}</span></h5> <pre>{function['doc']}</pre> </div> """ html
+= "</div>" # Add Usage Guide html += """ <div class="page-break"></div> <h2 id="usage-guide">Usage
Guide</h2> <h3>Setup and Installation</h3> <ol> <li>Extract the ZIP file to a local directory</li>
<li>Create a Python virtual environment: <code>python -m venv venv</code></li> <li>Activate the
virtual environment: <ul> <li>Windows: <code>venv\\Scripts\\activate</code></li> <li>Linux/Mac:
<code>source venv/bin/activate</code></li> </ul> </li> <li>Install dependencies: <code>pip install
flask flask-login flask-sqlalchemy gunicorn openpyxl pandas psycopg2-binary pygments sqlalchemy
werkzeug xhtml2pdf</code> </li> <li>Configure database connection: <code>export
DATABASE_URL="postgresql://username:password@localhost:5432/employee_management"</code></li>
<li>Run the application: <code>gunicorn --bind 0.0.0.0:5000 main:app</code></li> </ol> <h3>Initial
Setup</h3> <ol> <li>Navigate to <code>http://localhost:5000</code> in your browser</li> <li>Click
"Initialize system" to create admin user</li> <li>Log in with username "admin" and password
"admin123"</li> <li>Change the default password immediately</li> </ol> <h3>Using the
Application</h3> <ol> <li>Import employee data via the Import/Export section</li> <li>Add, edit, or
delete employees as needed</li> <li>Provide feedback to employees on a quarterly basis</li> <li>Use
the dashboard to monitor employee performance</li> </ol> </body> </html> """ return html def
create_pdf(html_content, output_path): """Convert HTML to PDF and save to file""" with
open(output_path, "w+b") as pdf_file: pisa_status = pisa.CreatePDF(html_content, dest=pdf_file) if
pisa_status.err: return f"Error creating PDF: {pisa_status.err}" else: return f"PDF documentation
created successfully at {output_path}" if __name__ == "__main__": html_content =
generate_html_documentation() result = create_pdf(html_content,
"EMS_Functionality_Documentation.pdf") print(result)
```

./docs.py

> ./docs.py

```
from flask import Blueprint, send_file, flash, redirect, url_for from flask_login import
login_required, current_user import os from utils import require_manager import subprocess # Create
a Blueprint for documentation docs_bp = Blueprint('docs', __name__)
@docs_bp.route('/documentation') @login_required @require_manager def generate_and_download_docs():
"""Generate and download the documentation PDF""" try: # Run the documentation generator script
result = subprocess.run(['python', 'generate_documentation.py'], capture_output=True, text=True,
check=True) # Check if the file was created if os.path.exists('EMS_Documentation.pdf'): return
send_file('EMS_Documentation.pdf', as_attachment=True) else: flash('Failed to generate
documentation: ' + result.stdout, 'danger') return redirect(url_for('employee.dashboard')) except
subprocess.CalledProcessError as e: flash(f'Error generating documentation: {e.stderr}', 'danger')
return redirect(url_for('employee.dashboard')) except Exception as e: flash(f'Unexpected error:
{str(e)}', 'danger') return redirect(url_for('employee.dashboard'))
```

./employee.py

> ./employee.py

```
from flask import Blueprint, render_template, redirect, url_for, flash, request, jsonify from
flask_login import login_required, current_user from werkzeug.utils import secure_filename from app
import db from models import Employee, User, Feedback from datetime import datetime import pandas
as pd import os import logging from utils import require_manager, export_employees_to_excel logger
= logging.getLogger(__name__) employee_bp = Blueprint('employee', __name__) @employee_bp.route('/')
@login_required def dashboard(): # Basic statistics for the dashboard total_employees =
Employee.query.count() # Add current date/time for dashboard display now = datetime.now() # For
managers, show their team stats if current_user.is_manager: managed_employees =
Employee.query.filter_by(manager_id=current_user.id).count() recent_feedback =
Feedback.query.filter_by(provided_by_id=current_user.id)\
.order_by(Feedback.feedback_date.desc()).limit(5).all() # Team distribution by skill team_skills =
db.session.query(Employee.skill, db.func.count(Employee.id))\
.filter_by(manager_id=current_user.id)\ .group_by(Employee.skill).all() context = {
'total_employees': total_employees, 'managed_employees': managed_employees, 'recent_feedback':
recent_feedback, 'team_skills': team_skills, 'is_manager': True, 'now': now } else: # For regular
employees, show their feedback employee = Employee.query.filter_by(user_id=current_user.id).first()
if employee: recent_feedback = Feedback.query.filter_by(employee_id=employee.id)\
.order_by(Feedback.feedback_date.desc()).limit(5).all() context = { 'total_employees':
total_employees, 'employee': employee, 'recent_feedback': recent_feedback, 'is_manager': False,
'now': now } else: context = { 'total_employees': total_employees, 'is_manager': False, 'now': now
} return render_template('dashboard.html', **context) @employee_bp.route('/employees')
@login_required def employee_list(): # Filter employees based on user's role if
current_user.is_manager: # Managers see all employees, but highlight their team employees =
Employee.query.all() else: # Regular employees only see themselves employee =
Employee.query.filter_by(user_id=current_user.id).first() employees = [employee] if employee else
[] # Get all managers for the dropdown in templates managers =
User.query.filter_by(is_manager=True).all() return render_template('employee_list.html',
employees=employees, managers=managers, is_manager=current_user.is_manager)
@employee_bp.route('/employees/<int:employee_id>') @login_required def
employee_detail(employee_id): employee = Employee.query.get_or_404(employee_id) # Regular employees
can only view their own details if not current_user.is_manager and not employee.user_id ==
current_user.id: flash('You do not have permission to view this employee.', 'danger') return
redirect(url_for('employee.employee_list')) # Get feedback for the employee feedback =
Feedback.query.filter_by(employee_id=employee.id)\ .order_by(Feedback.feedback_date.desc()).all() #
Get manager info manager = User.query.get(employee.manager_id) if employee.manager_id else None
return render_template('employee_detail.html', employee=employee, feedback=feedback,
manager=manager, is_manager=current_user.is_manager) @employee_bp.route('/employees/new',
methods=['GET', 'POST']) @login_required @require_manager def create_employee(): if request.method
== 'POST': try: # Get form data full_name = request.form.get('full_name') joining_date_str =
request.form.get('joining_date') benzyl = request.form.get('benzyl') role =
request.form.get('role') skill = request.form.get('skill') team = request.form.get('team') grade =
request.form.get('grade') designation = request.form.get('designation') location =
request.form.get('location') manager_id = request.form.get('manager_id') # Validate required fields
if not all([full_name, joining_date_str, benzyl, role, skill, team, grade, designation, location]):
flash('All fields are required', 'danger') return redirect(url_for('employee.create_employee')) #
Convert joining date string to date object joining_date = datetime.strptime(joining_date_str,
'%Y-%m-%d').date() # Create new employee new_employee = Employee( full_name=full_name,
joining_date=joining_date, benzyl=benzyl, role=role, skill=skill, team=team, grade=grade,
designation=designation, location=location, manager_id=manager_id if manager_id else
current_user.id ) db.session.add(new_employee) db.session.commit() flash('Employee created
successfully', 'success') return redirect(url_for('employee.employee_detail',
employee_id=new_employee.id)) except Exception as e: db.session.rollback() logger.error(f"Error
creating employee: {str(e)}") flash(f'Error creating employee: {str(e)}', 'danger') # Get all
managers for the dropdown managers = User.query.filter_by(is_manager=True).all() return
render_template('employee_form.html', employee=None, managers=managers, action='create')
@employee_bp.route('/employees/<int:employee_id>/edit', methods=['GET', 'POST']) @login_required
@require_manager def edit_employee(employee_id): employee = Employee.query.get_or_404(employee_id)
if request.method == 'POST': try: # Update employee data from form employee.full_name =
request.form.get('full_name') employee.joining_date =
datetime.strptime(request.form.get('joining_date'), '%Y-%m-%d').date() employee.benzyl =
request.form.get('benzyl') employee.role = request.form.get('role') employee.skill =
request.form.get('skill') employee.team = request.form.get('team') employee.grade =
request.form.get('grade') employee.designation = request.form.get('designation') employee.location
= request.form.get('location') employee.manager_id = request.form.get('manager_id')
db.session.commit() flash('Employee updated successfully', 'success') return
redirect(url_for('employee.employee_detail', employee_id=employee.id)) except Exception as e:
db.session.rollback() logger.error(f"Error updating employee: {str(e)}") flash(f'Error updating
employee: {str(e)}', 'danger') # Get all managers for the dropdown managers =
User.query.filter_by(is_manager=True).all() return render_template('employee_form.html',
employee=employee, managers=managers, action='edit')
```

```
@employee_bp.route('/employees/<int:employee_id>/delete', methods=['POST']) @login_required
@require_manager def delete_employee(employee_id): employee =
Employee.query.get_or_404(employee_id) try: # Also delete related feedback
Feedback.query.filter_by(employee_id=employee.id).delete() db.session.delete(employee)
db.session.commit() flash('Employee deleted successfully', 'success') except Exception as e:
db.session.rollback() logger.error(f"Error deleting employee: {str(e)}") flash(f'Error deleting
employee: {str(e)}', 'danger') return redirect(url_for('employee.employee_list'))
@employee_bp.route('/import-export', methods=['GET', 'POST']) @login_required @require_manager def
import_export(): if request.method == 'POST': action = request.form.get('action') if action ==
'import': # Check if file was uploaded if 'file' not in request.files: flash('No file part',
'danger') return redirect(request.url) file = request.files['file'] # Check if file was selected if
file.filename == '': flash('No file selected', 'danger') return redirect(request.url) # Check file
extension if file and file.filename.endswith(('.xlsx', '.xls', '.csv')): try: # Read Excel/CSV file
if file.filename.endswith('.csv'): df = pd.read_csv(file) else: df = pd.read_excel(file) # Process
data imported_count = 0 for _, row in df.iterrows(): # Check if employee exists existing_employee =
Employee.query.filter_by(benzyl=row.get('Benzyl')).first() # Convert joining date joining_date =
None joining_date_str = row.get('Joining Date') if isinstance(joining_date_str, str): try:
joining_date = datetime.strptime(joining_date_str, '%d-%m-%Y').date() except ValueError: try:
joining_date = datetime.strptime(joining_date_str, '%Y-%m-%d').date() except ValueError:
logger.warning(f"Could not parse date: {joining_date_str}") elif isinstance(joining_date_str,
pd.Timestamp): joining_date = joining_date_str.date() if not joining_date: continue if
existing_employee: # Update existing employee existing_employee.full_name = row.get('Full Name')
existing_employee.joining_date = joining_date existing_employee.role = row.get('Role')
existing_employee.skill = row.get('Skill') existing_employee.team = row.get('Team')
existing_employee.grade = row.get('Grade') existing_employee.designation = row.get('Designation')
existing_employee.location = row.get('Location') else: # Create new employee manager_name =
row.get('Manager Name') manager = User.query.filter_by(username=manager_name).first() new_employee
= Employee( full_name=row.get('Full Name'), joining_date=joining_date, benzyl=row.get('Benzyl'),
role=row.get('Role'), skill=row.get('Skill'), team=row.get('Team'), grade=row.get('Grade'),
designation=row.get('Designation'), location=row.get('Location'), manager_id=manager.id if manager
else current_user.id ) db.session.add(new_employee) imported_count += 1 db.session.commit()
flash(f'Successfully imported {imported_count} employees', 'success') except Exception as e:
db.session.rollback() logger.error(f"Error importing employees: {str(e)}") flash(f'Error importing
employees: {str(e)}', 'danger') else: flash('File format not supported. Please upload an Excel or
CSV file.', 'danger') elif action == 'export': try: # Export all employees to Excel excel_file =
export_employees_to_excel() flash('Employee data exported successfully', 'success') return
excel_file except Exception as e: logger.error(f"Error exporting employees: {str(e)}")
flash(f'Error exporting employees: {str(e)}', 'danger') return
render_template('import_export.html')
```

./feedback.py

./feedback.py

```
from flask import Blueprint, render_template, redirect, url_for, flash, request from flask_login
import login_required, current_user from app import db from models import Employee, Feedback from
datetime import datetime import logging from utils import require_manager, get_current_quarter
logger = logging.getLogger(__name__) feedback_bp = Blueprint('feedback', __name__,
url_prefix='/feedback') @feedback_bp.route('/') @login_required def feedback_list(): # For
managers, show feedback they've given if current_user.is_manager: feedback =
Feedback.query.filter_by(provided_by_id=current_user.id)\
.order_by(Feedback.feedback_date.desc()).all() context = { 'feedback': feedback, 'is_manager':
True, 'view_type': 'provided' } else: # For employees, show feedback they've received employee =
Employee.query.filter_by(user_id=current_user.id).first() if employee: feedback =
Feedback.query.filter_by(employee_id=employee.id)\ .order_by(Feedback.feedback_date.desc()).all()
context = { 'feedback': feedback, 'is_manager': False, 'view_type': 'received' } else: context = {
'feedback': [], 'is_manager': False, 'view_type': 'received' } return
render_template('feedback_list.html', **context) @feedback_bp.route('/employee/<int:employee_id>')
@login_required def employee_feedback(employee_id): employee =
Employee.query.get_or_404(employee_id) # Regular employees can only view their own feedback if not
current_user.is_manager and not employee.user_id == current_user.id: flash('You do not have
permission to view this feedback.', 'danger') return redirect(url_for('feedback.feedback_list')) #
Get feedback for the employee feedback = Feedback.query.filter_by(employee_id=employee.id)\
.order_by(Feedback.feedback_date.desc()).all() return render_template('feedback_list.html',
feedback=feedback, employee=employee, is_manager=current_user.is_manager, view_type='for_employee')
@feedback_bp.route('/new/<int:employee_id>', methods=['GET', 'POST']) @login_required
@require_manager def create_feedback(employee_id): employee =
Employee.query.get_or_404(employee_id) # Make sure the employee is managed by the current user if
employee.manager_id != current_user.id: flash('You can only provide feedback for employees you
manage.', 'danger') return redirect(url_for('employee.employee_list')) if request.method == 'POST':
try: # Get form data rating = int(request.form.get('rating')) feedback_text =
request.form.get('feedback_text') quarter = request.form.get('quarter') # Validate required fields
if not all([rating, feedback_text, quarter]): flash('All fields are required', 'danger') return
redirect(url_for('feedback.create_feedback', employee_id=employee_id)) # Validate rating range if
rating < 1 or rating > 5: flash('Rating must be between 1 and 5', 'danger') return
redirect(url_for('feedback.create_feedback', employee_id=employee_id)) # Create new feedback
new_feedback = Feedback( employee_id=employee_id, provided_by_id=current_user.id, rating=rating,
feedback_text=feedback_text, quarter=quarter, feedback_date=datetime.utcnow() )
db.session.add(new_feedback) db.session.commit() flash('Feedback submitted successfully',
'success') return redirect(url_for('feedback.employee_feedback', employee_id=employee_id)) except
Exception as e: db.session.rollback() logger.error(f"Error creating feedback: {str(e)}")
flash(f'Error submitting feedback: {str(e)}', 'danger') # Get current quarter for default selection
current_quarter = get_current_quarter() return render_template('feedback_form.html',
employee=employee, current_quarter=current_quarter, feedback=None, action='create')
@feedback_bp.route('/edit/<int:feedback_id>', methods=['GET', 'POST']) @login_required
@require_manager def edit_feedback(feedback_id): feedback = Feedback.query.get_or_404(feedback_id)
# Make sure the feedback was provided by the current user if feedback.provided_by_id !=
current_user.id: flash('You can only edit feedback you provided.', 'danger') return
redirect(url_for('feedback.feedback_list')) employee = Employee.query.get(feedback.employee_id) if
request.method == 'POST': try: # Update feedback data from form feedback.rating =
int(request.form.get('rating')) feedback.feedback_text = request.form.get('feedback_text')
feedback.quarter = request.form.get('quarter') # Validate rating range if feedback.rating < 1 or
feedback.rating > 5: flash('Rating must be between 1 and 5', 'danger') return
redirect(url_for('feedback.edit_feedback', feedback_id=feedback_id)) db.session.commit()
flash('Feedback updated successfully', 'success') return
redirect(url_for('feedback.employee_feedback', employee_id=employee.id)) except Exception as e:
db.session.rollback() logger.error(f"Error updating feedback: {str(e)}") flash(f'Error updating
feedback: {str(e)}', 'danger') return render_template('feedback_form.html', employee=employee,
feedback=feedback, action='edit') @feedback_bp.route('/delete/<int:feedback_id>', methods=['POST'])
@login_required @require_manager def delete_feedback(feedback_id): feedback =
Feedback.query.get_or_404(feedback_id) # Make sure the feedback was provided by the current user if
feedback.provided_by_id != current_user.id: flash('You can only delete feedback you provided.',
'danger') return redirect(url_for('feedback.feedback_list')) employee_id = feedback.employee_id
try: db.session.delete(feedback) db.session.commit() flash('Feedback deleted successfully',
'success') except Exception as e: db.session.rollback() logger.error(f"Error deleting feedback:
{str(e)}") flash(f'Error deleting feedback: {str(e)}', 'danger') return
redirect(url_for('feedback.employee_feedback', employee_id=employee_id))
```

./generate_complete_documentation.py

./generate_complete_documentation.py

```
""" Complete Documentation Generator for EMS Application This script: 1. Fixes template issues 2.
Creates placeholder images for documentation 3. Generates both functionality and code documentation
PDFs """ import os import subprocess import inspect import importlib from datetime import datetime
from xhtml2pdf import pisa # Create screenshots directory os.makedirs('screenshots', exist_ok=True)
def fix_template_issues(): """Fix the template issues with request.endpoint checks"""
template_files = [ 'templates/base.html', 'templates/errors/404.html', 'templates/errors/500.html'
] print("Fixing template issues...") for file_path in template_files: if os.path.exists(file_path):
try: with open(file_path, 'r', encoding='utf-8') as file: content = file.read() # Fix the
request.endpoint checks content = content.replace( "{% if 'employee_list' in request.endpoint %}",
"{% if request.endpoint and 'employee_list' in request.endpoint %}" ) content = content.replace(
"{% if 'feedback' in request.endpoint %}", "{% if request.endpoint and 'feedback' in
request.endpoint %}" ) content = content.replace( "{% if 'import_export' in request.endpoint %}",
"{% if request.endpoint and 'import_export' in request.endpoint %}" ) content = content.replace(
"{% if request.endpoint == 'employee.dashboard' %}", "{% if request.endpoint and request.endpoint
== 'employee.dashboard' %}" ) with open(file_path, 'w', encoding='utf-8') as file:
file.write(content) print(f" ✓ Fixed {file_path}") except Exception as e: print(f" ✗ Error fixing
{file_path}: {str(e)}") else: print(f" ■ File not found: {file_path}") print("Template fixes
complete.") def generate_class_documentation(): """Generate class-based documentation for the EMS
application""" print("Generating class and functionality documentation...") try: # Run the existing
script subprocess.run(['python', 'create_functionality_docs.py'], check=True) print(" ✓
Functionality documentation generated") except Exception as e: print(f" ✗ Error generating
functionality documentation: {str(e)}") def generate_code_documentation(): """Generate code-level
documentation for the EMS application""" print("Generating code documentation...") try: # Run the
existing script subprocess.run(['python', 'generate_documentation.py'], check=True) print(" ✓ Code
documentation generated") except Exception as e: print(f" ✗ Error generating code documentation:
{str(e)}") def main(): """Run the complete documentation generation process""" print("Starting
documentation generation process...") # Step 1: Fix template issues fix_template_issues() # Step 2:
Generate class and functionality documentation generate_class_documentation() # Step 3: Generate
code documentation generate_code_documentation() print("\nDocumentation generation complete!")
print("Files created:") print("- EMS_Functionality_Documentation.pdf (functionality details with
class documentation)") print("- EMS_Documentation.pdf (complete code listing)") print("\nNOTE: To
include actual screenshots, you will need to:") print("1. Run the application") print("2. Capture
screenshots of each page") print("3. Replace the placeholder images in the documentation") if
__name__ == "__main__": main()
```

./main.py

> ./main.py

```python
from app import app if __name__ == "__main__": app.run(host="0.0.0.0", port=5000, debug=True)
```

./models.py

> ./models.py

```
from datetime import datetime from app import db from flask_login import UserMixin from
werkzeug.security import generate_password_hash, check_password_hash class User(UserMixin,
db.Model): """User model for authentication and authorization""" __tablename__ = 'users' id =
db.Column(db.Integer, primary_key=True) username = db.Column(db.String(64), unique=True,
nullable=False) email = db.Column(db.String(120), unique=True, nullable=False) password_hash =
db.Column(db.String(256), nullable=False) is_manager = db.Column(db.Boolean, default=False) #
Relationships employees_managed = db.relationship('Employee', backref='manager_user',
lazy='dynamic', foreign_keys='Employee.manager_id') def set_password(self, password):
self.password_hash = generate_password_hash(password) def check_password(self, password): return
check_password_hash(self.password_hash, password) def __repr__(self): return f'<User
{self.username}>' class Employee(db.Model): """Employee model to store employee details"""
__tablename__ = 'employees' id = db.Column(db.Integer, primary_key=True) full_name =
db.Column(db.String(100), nullable=False) joining_date = db.Column(db.Date, nullable=False) benzyl
= db.Column(db.String(50), unique=True, nullable=False) # Employee ID or username role =
db.Column(db.String(100), nullable=False) skill = db.Column(db.String(100), nullable=False) team =
db.Column(db.String(100), nullable=False) grade = db.Column(db.String(20), nullable=False)
designation = db.Column(db.String(100), nullable=False) location = db.Column(db.String(100),
nullable=False) # User account related to this employee (if any) user_id = db.Column(db.Integer,
db.ForeignKey('users.id'), nullable=True) user = db.relationship('User', foreign_keys=[user_id]) #
Manager relationship manager_id = db.Column(db.Integer, db.ForeignKey('users.id'), nullable=True) #
Relationships feedback_received = db.relationship('Feedback', backref='employee', lazy='dynamic',
foreign_keys='Feedback.employee_id') def __repr__(self): return f'<Employee {self.full_name}>'
class Feedback(db.Model): """Feedback model to store manager feedback for employees"""
__tablename__ = 'feedback' id = db.Column(db.Integer, primary_key=True) employee_id =
db.Column(db.Integer, db.ForeignKey('employees.id'), nullable=False) provided_by_id =
db.Column(db.Integer, db.ForeignKey('users.id'), nullable=False) provided_by =
db.relationship('User', foreign_keys=[provided_by_id]) rating = db.Column(db.Integer,
nullable=False) # e.g., 1-5 rating scale feedback_text = db.Column(db.Text, nullable=False)
feedback_date = db.Column(db.DateTime, default=datetime.utcnow) quarter = db.Column(db.String(10),
nullable=False) # e.g., "Q1-2023" def __repr__(self): return f'<Feedback for Employee
#{self.employee_id} by {self.provided_by.username}>'
```

./simple_documentation_generator.py

./simple_documentation_generator.py

```
""" Simplified Documentation Generator for EMS Application This script generates documentation
without requiring the application to be running, avoiding circular import issues. """ import os
import inspect import importlib from datetime import datetime from xhtml2pdf import pisa import ast
def safe_import(module_name): """Try to import a module safely without crashing on circular
imports""" try: return importlib.import_module(module_name) except ImportError as e:
print(f"Warning: Could not import {module_name}: {str(e)}") return None def
extract_docstrings(file_path): """Extract docstrings and function signatures from Python files"""
with open(file_path, 'r', encoding='utf-8') as f: content = f.read() tree = ast.parse(content)
module_doc = ast.get_docstring(tree) classes = [] functions = [] for node in tree.body: if
isinstance(node, ast.ClassDef): class_info = { 'name': node.name, 'doc': ast.get_docstring(node) or
"No documentation available.", 'methods': [] } for item in node.body: if isinstance(item,
ast.FunctionDef): if item.name.startswith('_') and item.name != '__init__': continue args = [] for
arg in item.args.args: args.append(arg.arg) method_info = { 'name': item.name, 'doc':
ast.get_docstring(item) or "No documentation available.", 'signature': f"({', '.join(args)})" }
class_info['methods'].append(method_info) classes.append(class_info) elif isinstance(node,
ast.FunctionDef): args = [] for arg in node.args.args: args.append(arg.arg) function_info = {
'name': node.name, 'doc': ast.get_docstring(node) or "No documentation available.", 'signature':
f"({', '.join(args)})" } functions.append(function_info) return { 'doc': module_doc or "No
documentation available.", 'classes': classes, 'functions': functions } def
generate_html_documentation(): """Generate HTML documentation by parsing Python files""" # List of
Python files to document files = [ ('models.py', 'Data Models'), ('auth.py', 'Authentication
Module'), ('employee.py', 'Employee Management Module'), ('feedback.py', 'Feedback System Module'),
('utils.py', 'Utility Functions') ] modules_data = [] for file_name, description in files: if
os.path.exists(file_name): module_data = extract_docstrings(file_name) module_data['name'] =
file_name.replace('.py', '') module_data['description'] = description
modules_data.append(module_data) else: print(f"Warning: File {file_name} not found") # Start
building HTML html = f""" <!DOCTYPE html> <html> <head> <meta charset="UTF-8"> <title>EMS
Application Functionality Documentation</title> <style> body {{ font-family: Arial, sans-serif;
margin: 20px; }} h1 {{ color: #2c3e50; text-align: center; }} h2 {{ color: #3498db; border-bottom:
1px solid #3498db; padding-bottom: 5px; }} h3 {{ color: #2980b9; }} h4 {{ color: #16a085; }} pre {{
background-color: #f8f9fa; padding: 10px; border-radius: 4px; }} .signature {{ font-family:
monospace; background-color: #f0f0f0; padding: 3px; }} .module {{ margin-bottom: 30px; border: 1px
solid #ddd; padding: 15px; border-radius: 5px; }} .class {{ margin-bottom: 20px; background-color:
#f8f9fa; padding: 10px; border-radius: 4px; }} .method {{ margin-left: 20px; margin-bottom: 10px;
}} .function {{ margin-bottom: 15px; background-color: #f8f9fa; padding: 10px; border-radius: 4px;
}} .toc {{ background-color: #f8f9fa; padding: 10px; margin-bottom: 20px; }} .toc ul {{
padding-left: 20px; }} .screenshot {{ width: 95%; border: 1px solid #ddd; margin: 10px 0; }}
.page-break {{ page-break-before: always; }} @page {{ size: A4; margin: 2cm; }} </style> </head>
<body> <h1>Employee Management System Functionality Documentation</h1> <p style="text-align:
center;">Generated on: {datetime.now().strftime('%Y-%m-%d %H:%M:%S')}</p> <div class="toc">
<h2>Table of Contents</h2> <ul> <li><a href="#overview">Application Overview</a></li> <li><a
href="#functionality">Key Functionalities</a> <ul> <li><a href="#auth">Authentication
System</a></li> <li><a href="#employee-management">Employee Management</a></li> <li><a
href="#feedback-system">Feedback System</a></li> <li><a href="#import-export">Import/Export
Functionality</a></li> </ul> </li> <li><a href="#class-details">Module Details</a> <ul> """ # Add
module links to TOC for module_data in modules_data: html += f'<li><a
href="#{module_data["name"]}">{module_data["description"]}</a></li>\n' html += """ </ul> </li>
<li><a href="#usage-guide">Usage Guide</a></li> </ul> </div> <!-- Application Overview --> <div
class="page-break"></div> <h2 id="overview">Application Overview</h2> <p> The Employee Management
System (EMS) is a comprehensive web application built with Flask and PostgreSQL that allows line
managers to maintain employee details and provide feedback throughout the year. The system provides
full CRUD operations for employee data management, feedback tracking, and reporting. </p> <p> The
application is organized with a modular structure, separating concerns into different components:
authentication, employee management, feedback system, and utility functions. </p> <!-- Key
Functionalities with Screenshots --> <div class="page-break"></div> <h2 id="functionality">Key
Functionalities</h2> <h3 id="auth">Authentication System</h3> <p>The authentication system handles
user login, logout, and profile management with the following features:</p> <ul> <li>User login
with username and password</li> <li>Role-based access control (manager vs. employee)</li>
<li>Profile management</li> <li>Password security using bcrypt hashing</li> </ul> <p>Key
Classes/Functions:</p> <ul> <li><code>User</code> model in models.py</li> <li><code>login()</code>,
<code>logout()</code>, <code>profile()</code> functions in auth.py</li> </ul> <h4>Authentication
Screenshots</h4> <p><strong>Login Page</strong></p> <div class="screenshot-placeholder"> [Login
Page Screenshot] </div> <p><strong>User Profile Page</strong></p> <div
class="screenshot-placeholder"> [Profile Page Screenshot] </div> <h3
id="employee-management">Employee Management</h3> <p>The employee management module handles all
operations related to employee data:</p> <ul> <li>Dashboard overview of team and statistics</li>
<li>Employee listing with sorting and filtering</li> <li>Employee details view</li> <li>Create,
update, and delete employee records</li> </ul> <p>Key Classes/Functions:</p> <ul>
<li><code>Employee</code> model in models.py</li> <li><code>dashboard()</code>,
<code>employee_list()</code>, <code>create_employee()</code> functions in employee.py</li> </ul>
```

```
<h4>Employee Management Screenshots</h4> <p><strong>Dashboard</strong></p> <div
class="screenshot-placeholder"> [Dashboard Screenshot] </div> <p><strong>Employee List</strong></p>
<div class="screenshot-placeholder"> [Employee List Screenshot] </div> <p><strong>Employee
Details</strong></p> <div class="screenshot-placeholder"> [Employee Details Screenshot] </div> <h3
id="feedback-system">Feedback System</h3> <p>The feedback system allows managers to provide
structured feedback to their reportees:</p> <ul> <li>Create quarterly feedback with ratings and
comments</li> <li>View feedback history for each employee</li> <li>Edit and delete feedback</li>
</ul> <p>Key Classes/Functions:</p> <ul> <li><code>Feedback</code> model in models.py</li>
<li><code>feedback_list()</code>, <code>create_feedback()</code> functions in feedback.py</li>
</ul> <h4>Feedback System Screenshots</h4> <p><strong>Feedback List</strong></p> <div
class="screenshot-placeholder"> [Feedback List Screenshot] </div> <p><strong>Create
Feedback</strong></p> <div class="screenshot-placeholder"> [Create Feedback Screenshot] </div> <h3
id="import-export">Import/Export Functionality</h3> <p>The import/export module enables data
exchange with Excel spreadsheets:</p> <ul> <li>Import employee data from Excel
(manager_toolkit.xlsx)</li> <li>Export employee data to Excel format</li> </ul> <p>Key
Classes/Functions:</p> <ul> <li><code>import_export()</code> function in employee.py</li>
<li><code>export_employees_to_excel()</code> function in utils.py</li> </ul> <h4>Import/Export
Screenshots</h4> <p><strong>Import/Export Interface</strong></p> <div
class="screenshot-placeholder"> [Import/Export Screenshot] </div> <!-- Module Details Section -->
<div class="page-break"></div> <h2 id="class-details">Module Details</h2> """ # Add module details
for module_data in modules_data: html += f""" <div class="module page-break"
id="{module_data['name']}"> <h3>{module_data['name']}.py - {module_data['description']}</h3>
<pre>{module_data['doc']}</pre> <h4>Classes:</h4> """ if not module_data['classes']: html += "<p>No
classes defined in this module.</p>" else: for class_data in module_data['classes']: html += f"""
<div class="class"> <h4>{class_data['name']}</h4> <pre>{class_data['doc']}</pre> <h5>Methods:</h5>
""" if not class_data['methods']: html += "<p>No methods defined in this class.</p>" else: for
method in class_data['methods']: html += f""" <div class="method"> <h6>{method['name']}<span
class="signature">{method['signature']}</span></h6> <pre>{method['doc']}</pre> </div> """ html +=
"</div>" html += """ <h4>Functions:</h4> """ if not module_data['functions']: html += "<p>No
functions defined in this module.</p>" else: for function in module_data['functions']: html += f"""
<div class="function"> <h5>{function['name']}<span
class="signature">{function['signature']}</span></h5> <pre>{function['doc']}</pre> </div> """ html
+= "</div>" # Add Usage Guide html += """ <div class="page-break"></div> <h2 id="usage-guide">Usage
Guide</h2> <h3>Setup and Installation</h3> <ol> <li>Extract the ZIP file to a local directory</li>
<li>Create a Python virtual environment: <code>python -m venv venv</code></li> <li>Activate the
virtual environment: <ul> <li>Windows: <code>venv\\Scripts\\activate</code></li> <li>Linux/Mac:
<code>source venv/bin/activate</code></li> </ul> </li> <li>Install dependencies: <code>pip install
flask flask-login flask-sqlalchemy gunicorn openpyxl pandas psycopg2-binary pygments sqlalchemy
werkzeug xhtml2pdf</code> </li> <li>Configure database connection: <code>export
DATABASE_URL="postgresql://username:password@localhost:5432/employee_management"</code></li>
<li>Run the application: <code>gunicorn --bind 0.0.0.0:5000 main:app</code></li> </ol> <h3>Initial
Setup</h3> <ol> <li>Navigate to <code>http://localhost:5000</code> in your browser</li> <li>Click
"Initialize system" to create admin user</li> <li>Log in with username "admin" and password
"admin123"</li> <li>Change the default password immediately</li> </ol> <h3>Using the
Application</h3> <ol> <li>Import employee data via the Import/Export section</li> <li>Add, edit, or
delete employees as needed</li> <li>Provide feedback to employees on a quarterly basis</li> <li>Use
the dashboard to monitor employee performance</li> </ol> </body> </html> """ return html def
create_pdf(html_content, output_path): """Convert HTML to PDF and save to file""" with
open(output_path, "w+b") as pdf_file: pisa_status = pisa.CreatePDF(html_content, dest=pdf_file) if
pisa_status.err: return f"Error creating PDF: {pisa_status.err}" else: return f"PDF documentation
created successfully at {output_path}" if __name__ == "__main__": html_content =
generate_html_documentation() result = create_pdf(html_content,
"EMS_Functionality_Documentation.pdf") print(result)
```

./utils.py

> ./utils.py

```
from functools import wraps from flask import flash, redirect, url_for, Response from flask_login
import current_user from datetime import datetime import pandas as pd import io from models import
Employee, User def require_manager(f): """Decorator to require manager role for a view""" @wraps(f)
def decorated_function(*args, **kwargs): if not current_user.is_manager: flash('This action
requires manager privileges', 'danger') return redirect(url_for('employee.dashboard')) return
f(*args, **kwargs) return decorated_function def get_current_quarter(): """Get the current quarter
in format 'Q1-YYYY'""" now = datetime.now() quarter = (now.month - 1) // 3 + 1 return
f"Q{quarter}-{now.year}" def export_employees_to_excel(): """Export employee data to Excel""" #
Query all employees employees = Employee.query.all() # Create dataframe data = [] for employee in
employees: manager = User.query.get(employee.manager_id) if employee.manager_id else None
manager_name = manager.username if manager else "" data.append({ 'Full Name': employee.full_name,
'Joining Date': employee.joining_date, 'Benzyl': employee.benzyl, 'Role': employee.role, 'Skill':
employee.skill, 'Team': employee.team, 'Manager Name': manager_name, 'Grade': employee.grade,
'Designation': employee.designation, 'Location': employee.location }) # Convert to DataFrame df =
pd.DataFrame(data) # Create Excel file in memory output = io.BytesIO() with pd.ExcelWriter(output,
engine='openpyxl') as writer: df.to_excel(writer, index=False, sheet_name='Employees')
output.seek(0) # Create response with Excel file return Response( output,
mimetype='application/vnd.openxmlformats-officedocument.spreadsheetml.sheet', headers={
'Content-Disposition': 'attachment; filename=employee_data.xlsx' } )
```