**Core Features Breakdown**

1. **User Roles and Permissions:**
   * **Administrator:** Full access, manage all aspects.
   * **Manager:** Approve/deny requests, view subordinate data, and oversee performance.
   * **Employee:** View personal data, request leave/advances, check attendance and salary.
2. **Employee Management:**
   * **Employee List and Detail:** View and manage employee information.
   * **Employee Creation and Editing:** Admin functionality to add/edit employees.
3. **Time and Attendance:**
   * **Timesheet & Attendance Management:** Track work hours, and attendance. Integrate clock-in/out features.
   * **Leave Management:** Employees can request leave, and managers/admins can approve.
4. **Performance Management:**
   * **Goal Setting and Performance Review:** Managers set goals and conduct reviews.
5. **Payroll and Benefits:**
   * **Payroll Management:** View and manage payroll data.
   * **Benefits Management:** Employees view benefits, enroll, and update information.
6. **Additional Features:**
   * **Document Management:** Store and manage employee-related documents.
   * **Internal Communication:** Messaging system for communication.
   * **Announcements and Notifications:** Share company updates and alerts.
   * **Reports:** Generate and view various reports (attendance, payroll, etc.).

**Essential Angular Components**

1. **Authentication Components:**
   * LogInComponent, RegisterComponent, ForgotPasswordComponent
2. **Employee Management Components:**
   * EmployeeListComponent, EmployeeDetailComponent, EmployeeCreateComponent, EmployeeEditComponent
3. **Attendance Management Components:**
   * AttendanceListComponent, AttendanceCreateComponent, AttendanceEditComponent
4. **Leave Management Components:**
   * LeaveListComponent, LeaveDetailComponent, LeaveCreateComponent, LeaveEditComponent
5. **Performance Management Components:**
   * PerformanceListComponent, PerformanceDetailComponent, PerformanceCreateComponent, PerformanceEditComponent
6. **Payroll Management Components:**
   * PayrollListComponent, PayrollDetailComponent, PayrollCreateComponent, PayrollEditComponent
7. **Common & Utility Components:**
   * NavbarComponent, SidebarComponent, DashboardComponent, ProfileComponent, SettingsComponent, NotificationComponent, PaginationComponent, SearchComponent, FilterComponent, SortComponent
8. **Security & Error Handling Components:**
   * RoleManagementComponent, AccessControlComponent, NotFoundComponent, ErrorComponent
9. **Reports Components:**
   * ReportListComponent, ReportDetailComponent, ReportCreateComponent, ReportEditComponent, ReportDeleteComponent

**Step-by-Step Development Guide**

1. **Initial Setup:**
   * Set up Angular project using Angular CLI.
   * Implement routing and set up a basic layout with NavbarComponent and SidebarComponent.
2. **Authentication Module:**
   * Develop components for login, registration, and password recovery.
   * Integrate with a backend authentication service (to be developed with Spring Boot).
3. **Role-Based Access Control:**
   * Implement a service to manage user roles and permissions.
   * Protect routes and components based on roles.
4. **Employee Management:**
   * Develop components for listing, creating, editing, and viewing employee details.
   * Ensure data binding and validation in forms.
5. **Attendance and Leave Management:**
   * Build components for attendance and leave management.
   * Implement form validations and date pickers for leave applications.
6. **Performance and Payroll Management:**
   * Develop performance review and goal-setting components.
   * Implement payroll management, ensuring data security.
7. **Additional Functionalities:**
   * Add document management, internal communication, and announcement features.
   * Develop reporting functionalities for generating various reports.
8. **Testing and Optimization:**
   * Perform unit and integration testing.
   * Optimize components for performance and responsiveness.
9. **Backend Integration:**
   * Connect the Angular frontend with a Spring Boot backend for data persistence and business logic.
10. **Deployment:**
    * Prepare the application for deployment, including configuration for production builds.
    * Deploy using a suitable hosting service and configure CI/CD pipelines if needed.

**Further Considerations**

* **Scalability:** Ensure the architecture can handle scaling as the user base grows.
* **Security:** Implement robust authentication, authorization, and data protection measures.
* **Usability:** Focus on creating an intuitive and user-friendly interface.

**Payroll or Employee Management System**

**There are 3 users in the system**

* Administrator
* Manager
* Employee

**The Role of Administrator is**

1. Create a new Bonus.
2. View all the Bonus, Salary Advance requests, and Leave request and grant requests.
3. Create an Employee.
4. View all the Employees.
5. Create an employee’s salary.

**The Role of Manager is**

1. Check

**The Role of Employee is**

1. Check in and see the attendance list.
2. View all the Bonus, Salary Advance requests, and Leave request and approval status.
3. Create a new salary advance request.
4. Create a new Leave request.
5. Check salary according to month Employee's salary.

**Full Feature of My Project**

**Core Employee Data Management:**

* **Employee List:** This component would display a list of all employees with basic information like name, ID, department, and designation. It could offer functionalities like searching, sorting, and filtering.
* **Employee Profile:** This component would show detailed information about a specific employee, including contact details, emergency contacts, job description, skills, and work experience. It might allow authorized users to edit this information.

**Time and Attendance:**

* **Timesheet:** This component would allow employees to submit their daily work hours, breaks, and leave requests. It could integrate with a clock-in/out system for automatic data capture. (This might require additional services)
* **Leave Management:** This component would allow employees to request various types of leave (sick leave, vacation, etc.), track their remaining leave balance, and view the approval status of their requests. Managers could approve or reject leave requests within this component.

**Performance Management:**

* **Goal Setting:** This component allows managers to set performance goals for their employees and track their progress throughout the review period.
* **Performance Review:** This component facilitates conducting performance reviews. It could include pre-defined templates for evaluation and feedback.

**Payroll and Benefits:**

* **Payroll Summary:** This component (restricted to authorized users) could display a summary of employee salaries, deductions, and net pay. (Integration with payroll systems might be needed)
* **Benefits Management:** This component would allow employees to view their benefits package details, enroll in different benefits, and update their beneficiary information.

**Additional Components:**

* **Document Management:** This component allows storing and managing employee-related documents like resumes, contracts, and certificates.
* **Announcement Board:** This component provides a platform for HR or management to share company announcements and updates with all employees.
* **Internal Communication:** This component could facilitate internal communication between employees and managers through messages or group discussions.

**Important Considerations:**

* **User Roles and Permissions:** Implement different user roles (admin, manager, employee) with varying access levels to specific functionalities.
* **Data Security:** Ensure secure storage and transmission of sensitive employee data using encryption and secure authentication methods.
* **Integrations:** Consider integrating with existing HR systems or third-party services for functionalities like payroll or timekeeping.

**All eligible Component**

**Authentication Components**

1. **LogInComponent**: Handles user login.
2. **LogOutComponent:** Handles user logout.
3. **RegisterComponent**: Allows new users to register.
4. **ForgotPasswordComponent**: Manages password recovery.

**Employee Management Components**

1. **EmployeeListComponent**: Displays a list of employees.
2. **EmployeeDetailComponent**: Shows detailed information about a single employee.
3. **EmployeeCreateComponent**: Form to add a new employee.
4. **EmployeeEditComponent**: Form to edit existing employee details.
5. **EmployeeDeleteComponent**: Confirms and handles the deletion of an employee.

**Department Management Components**

1. **DepartmentListComponent**: Displays a list of departments.
2. **DepartmentDetailComponent**: Shows detailed information about a single department.
3. **DepartmentCreateComponent**: Form to add a new department.
4. **DepartmentEditComponent**: Form to edit existing department details.
5. **DepartmentDeleteComponent**: Confirms and handles the deletion of a department.

**Attendance Management Components**

1. **AttendanceListComponent**: Displays attendance records.
2. **AttendanceDetailComponent**: Shows detailed attendance information for a single employee.
3. **AttendanceCreateComponent**: Allows for the addition of new attendance records.
4. **AttendanceEditComponent**: Form to edit existing attendance records.
5. **AttendanceDeleteComponent**: Confirms and handles the deletion of attendance records.

**Leave Management Components**

1. **LeaveListComponent**: Displays a list of leave applications.
2. **LeaveDetailComponent**: Shows detailed information about a single leave application.
3. **LeaveCreateComponent**: Form to apply for leave.
4. **LeaveEditComponent**: Form to edit existing leave applications.
5. **LeaveDeleteComponent**: Confirms and handles the deletion of a leave application.

**Performance Management Components**

1. **PerformanceListComponent**: Displays performance reviews.
2. **PerformanceDetailComponent**: Shows detailed performance review for a single employee.
3. **PerformanceCreateComponent**: Form to add a new performance review.
4. **PerformanceEditComponent**: Form to edit existing performance reviews.
5. **PerformanceDeleteComponent**: Confirms and handles the deletion of a performance review.

**Payroll Management Components**

1. **PayrollListComponent**: Displays payroll records.
2. **PayrollDetailComponent**: Shows detailed payroll information for a single employee.
3. **PayrollCreateComponent**: Form to add a new payroll record.
4. **PayrollEditComponent**: Form to edit existing payroll records.
5. **PayrollDeleteComponent**: Confirms and handles the deletion of a payroll record.

**Common Components**

1. **NavbarComponent**: Navigation bar for the application.
2. **SidebarComponent**: Sidebar menu for quick access to different parts of the application.
3. **DashboardComponent**: Provides an overview and summary of key metrics.
4. **ProfileComponent**: Displays and allows editing of the logged-in user's profile.
5. **SettingsComponent**: Application settings and preferences.
6. **NotificationComponent**: Handles and displays system notifications.

**Utility Components**

1. **PaginationComponent**: Handles pagination for lists.
2. **SearchComponent**: Provides search functionality across various components.
3. **FilterComponent**: Provides filtering options for lists.
4. **SortComponent**: Handles sorting of list data.
5. **ModalComponent**: Generic modal dialog for confirmation and other interactions.

**Error Handling Components**

1. **NotFoundComponent**: Displays a 404 not found page.
2. **ErrorComponent**: Generic error page for handling unexpected errors.

**Security Components**

1. **RoleManagementComponent**: Manages user roles and permissions.
2. **AccessControlComponent**: Controls access to various parts of the application based on roles and permissions.

**Reports Components**

1. **ReportListComponent**: Displays various reports.
2. **ReportDetailComponent**: Shows detailed information about a single report.
3. **ReportCreateComponent**: Form to create a new report.
4. **ReportEditComponent**: Form to edit existing reports.
5. **ReportDeleteComponent**: Confirms and handles the deletion of a report.

I want to make a frontend project using angular. Base on this article feature which part should I remove or add? Then I will make backend using spring boot. Now suggest me and guide me step by step process to fulfill my payroll employee management system based on this article using angular.

For my Angular project, I have created a payroll or employee management system project based on above article. Then Bootstrap connected. There I have created many components, services, models etc. First of all, I created attendance, dashboard, department, employee, leave, login, logout, payroll, performance, profile as subfolders in src/app/components. Within this subfolder, I have created 5 subfolders such as attendance, department, employee, leave, performance, such as: attendance-create, attendance-delete, attendance-detail, attendance-edit, attendance-list. That is, the common subfolder of each subfolder is create, delete, details, edit, list. I have created 2 files named auth.guard.ts and role.guard.ts in a folder named Guards. I have created model.ts file named attendance, compliance-tracking, custom-reports, department, employee, heath-programs, hr-analytics, leave, login, logout, onboarding, payroll, performance in a folder called models. In a folder called Pages, compliance-tracking, custom-reports, employee-feedback, expense-claims, heath-programs, hr-analytics, offboarding, policy-documents, profile-update, skill-development, surveys, training-programs, wellness- Created components called resources. Create a folder called services, put all the services files in there, such as: attendance, auth, department, employee, leave, payroll, performance etc.