**Key Features of the Organizational Welfare Portal**

**1. Staff Dashboard**

* **View Contributions**: Staff can track their monthly contributions to the welfare pool.
* **Deposit Funds**: Staff can make additional deposits beyond their regular contributions.
* **Loan Access**: Staff qualify for loans up to **10 times their total deposits**.

**2. Loan System (Two Components)**

* **Actual Loans**: Traditional loans where staff receive cash directly.
* **Hire Purchase**: Allows staff to acquire goods/assets, with payments made in installments.
* **Loan Tracking**: Tracks the **loan start date** and the **repayment start date** for each loan.
* **Repayment Schedule**: Displays a clear breakdown of repayment timelines and amounts.

**3. Interest Rate Management**

* **Unfixed Interest Rate**: Interest is calculated based on the organization's profit for a period.
* **Dynamic Rate Application**: Interest is applied to all loans, with flexibility for changes in profit.

**4. Admin/Management Controls**

* **Profit Input**: Admins set the organization’s profit for a period, which determines the interest rate.
* **Loan Rules**: Admins can define rules for **Actual Loans** and **Hire Purchase** (e.g., repayment periods, maximum limits).
* **Reports & Analytics**: View reports on contributions, deposits, loans, and outstanding balances.

**5. Security & Role-Based Access**

* **User Roles**: Role-based access for staff, admins, and managers.
* **Audit Trail**: Tracks all user actions for transparency and accountability.

**6. Notifications & Alerts**

* **Deposit Confirmations**: Notifies staff when deposits are successfully made.
* **Loan Status Updates**: Alerts staff about loan approvals, rejections, and repayment reminders.
* **Monthly Summaries**: Sends summaries of contributions, deposits, loans, and interest changes.

**Technology Stack**

* **Backend**: PHP (Core PHP or with a framework like **Laravel** for faster development, better security, and cleaner code)
* **Frontend**: HTML, CSS, JavaScript (with frameworks like **Vue.js** or **React** if needed for interactivity)
* **Database**: MySQL or MariaDB to store user data, contributions, loans, and system logs.
* **Server**: Apache or Nginx (you can use **XAMPP** for local development)
* **Security**: Implement secure login, role-based access, data encryption, and audit trails.