

EMPLOYEE MANAGEMENT SYSTEM PROJECT DOCUMENTATION Ebook

Employee Management System Project Documentation

In today's dynamic business environment, efficient employee management is crucial for organizational success. Developing a comprehensive Employee Management System (EMS) can streamline HR processes, improve data accuracy, and enhance overall productivity. A well-structured project documentation for an employee management system serves as a blueprint guiding developers, stakeholders, and users through the system's architecture, functionalities, and implementation strategies. This article provides an in-depth overview of employee management system project documentation, covering essential components, best practices, and key considerations to ensure the successful development and deployment of such a system.

Overview of Employee Management System

An Employee Management System is a software application designed to manage all employee-related information and processes within an organization. It automates tasks such as employee data management, attendance tracking, payroll processing, leave management, performance evaluation, and more.

Key Objectives of an Employee Management System:

- Centralize employee data for easy access and management
- Automate routine HR tasks to reduce manual effort
- Improve data accuracy and consistency
- Facilitate better decision-making through analytics
- Enhance employee engagement and communication

Importance of Project Documentation in Employee Management System

Project documentation plays a vital role throughout the software development lifecycle. It ensures that all stakeholders have a clear understanding of the system's purpose, scope, functionalities, and technical details. Proper documentation helps in:

- Clarifying project requirements and objectives
- Guiding developers during system design and implementation
- Providing maintenance and upgrade references
- Ensuring compliance with organizational and legal standards
- Facilitating onboarding of new team members

Components of Employee Management System Project Documentation

A comprehensive project documentation for an employee management system encompasses several key sections:

1. Introduction

- Purpose: Define the primary goal of the system.
- Scope: Outline the functionalities covered and limitations.
- Audience: Identify who will use the documentation (developers, testers, end-users).

2. System Overview

- Description of the overall system architecture.
- Core modules and their interactions.
- Technologies and platforms used.

3. Functional Requirements

- Detailed description of features such as:
 - Employee registration and profile management
 - Attendance and leave management
 - Payroll processing
 - Performance appraisal
 - Department and role management
 - Reporting and analytics

4. Non-Functional Requirements

- Performance benchmarks

- Security standards
- Usability considerations
- Scalability and maintainability

5. System Design and Architecture

- High-Level Design: Overview of system components and their interactions.
- Detailed Design: Class diagrams, flowcharts, database schemas, and UI wireframes.
- Technology Stack: Programming languages, frameworks, databases, and third-party tools.

6. Database Design

- Entity-Relationship (ER) diagrams
- Tables, attributes, and relationships
- Data validation rules

7. User Interface Design

- Mockups and wireframes for different modules
- Navigation flow
- Accessibility considerations

8. Implementation Plan

- Development milestones
- Task allocation
- Testing strategies

9. Testing and Validation

- Types of testing (unit, integration, system, acceptance)
- Test cases and expected results
- Bug tracking and resolution process

10. Deployment and Maintenance

- Deployment environment setup
- Data migration procedures
- Support and maintenance plan

11. Appendices

- Glossary of terms
- References and resources
- Change log

Best Practices for Creating Effective Employee Management System Documentation

To ensure the documentation is useful and effective, consider the following best practices:

- **Clarity and Conciseness:** Use clear language and avoid ambiguity.
- **Organization:** Structure the document logically with a table of contents.
- **Visual Aids:** Incorporate diagrams, flowcharts, and wireframes to illustrate concepts.
- **Version Control:** Maintain version history to track updates.
- **Stakeholder Involvement:** Get feedback from end-users, developers, and managers.
- **Regular Updates:** Keep documentation current with system changes.

Tools and Technologies for Employee Management System Development

Choosing the right tools and technologies is essential for a successful project. Some popular options include:

- **Programming Languages:** Java, C, Python, PHP
- **Frameworks:** Spring Boot, ASP.NET, Django, Laravel
- **Databases:** MySQL, PostgreSQL, MongoDB
- **Front-End Technologies:** Angular, React, Vue.js
- **Deployment Platforms:** Cloud services like AWS, Azure, or on-premises servers

Security Considerations in Employee Management System

Given the sensitive nature of employee data, security must be a priority. Key security measures include:

- User authentication and role-based access control
- Data encryption during storage and transmission
- Regular security audits
- Backup and disaster recovery plans
- Compliance with data protection regulations like GDPR or HIPAA

Conclusion

Developing a robust employee management system requires meticulous planning, clear documentation, and adherence to best practices. The project documentation serves as a roadmap that aligns stakeholders, guides development, and ensures system quality. By thoroughly documenting system requirements, design, implementation plans, and maintenance strategies, organizations can streamline the development process, reduce errors, and create a scalable and secure solution that meets their HR management needs effectively. Whether for small businesses or large enterprises, comprehensive project documentation forms the foundation for a successful employee management system that enhances operational efficiency and supports organizational growth.

Employee Management System Project Documentation: A Comprehensive Guide to Structuring and Implementing an Effective System

In today's dynamic business landscape, managing human resources efficiently is paramount for organizational success. The backbone of this efficiency often lies in the implementation of a robust employee management system (EMS)—a centralized platform that streamlines HR processes, enhances data accuracy, and fosters better decision-making. However, the foundation of any successful EMS project is thorough and well-structured project documentation. This documentation acts as a blueprint, guiding developers, stakeholders, and users through every phase of the project—from conception to deployment and maintenance.

This article delves into the essentials of employee management system project documentation, exploring its purpose, key components, best practices, and how to craft comprehensive documents that facilitate smooth project execution.

The Significance of Project Documentation in Employee Management System Projects

Before exploring the components, it's vital to understand why project documentation is indispensable for EMS projects.

- **Clarity of Scope and Objectives:** Clear documentation helps define what the system will achieve, aligning stakeholder expectations and preventing scope creep.

- Guidance for Development and Implementation: Well-structured documents serve as a roadmap, guiding developers through system design, coding standards, and testing protocols.
- Facilitation of Communication: Documentation bridges the gap between technical teams, management, and end-users, ensuring everyone has a shared understanding.
- Risk Management: Detailed specifications help identify potential challenges early, allowing for mitigation strategies.
- Maintenance and Future Enhancements: Proper documentation ensures ongoing support, updates, and scalability of the system over time.

Core Components of Employee Management System Project Documentation

Creating comprehensive project documentation involves multiple interconnected sections. Each plays a vital role in ensuring project clarity, scope management, and successful delivery.

- Project Overview and Introduction

Purpose and Scope

Begin with a clear statement explaining the motivation behind developing the EMS. Describe the problems it intends to solve, such as inefficient manual record-keeping or inconsistent data management.

Objectives

Specify what the project aims to accomplish. For example:

- Centralize employee data management
- Automate attendance and leave tracking
- Streamline payroll processing
- Facilitate performance appraisals

Stakeholders

Identify key stakeholders involved, including:

- HR managers
- IT department
- Finance team

- End-users (employees, managers)
- External vendors (if any)

- Requirements Specification

This section details what the system must do, serving as the foundation for design and development.

Functional Requirements

Features and functionalities, such as:

- Employee onboarding and offboarding
- Personal and professional data management
- Attendance and leave management
- Payroll calculation and processing
- Performance evaluation modules
- Role-based access control

Non-Functional Requirements

Quality attributes, including:

- Security standards (encryption, authentication)
- System performance (response time, uptime)
- Scalability considerations
- Usability and user experience
- Data backup and disaster recovery

Constraints and Assumptions

Document any limitations, like budget constraints, technology preferences, or organizational policies.

- System Design and Architecture

System Architecture Diagram

Visual representation of the system's structure, illustrating components such as:

- Database servers
- Application servers
- Client interfaces (web or mobile)
- Integration points with other systems (e.g., payroll, accounting)

Technology Stack

Specify programming languages, frameworks, database systems, and tools chosen for development.

Data Flow and Process Models

Describe how data moves through the system, including:

- Data input methods
 - Processing logic
 - Data output and reports
-
- Database Design

Entity-Relationship Diagrams (ERD)

Visualize data entities such as Employee, Department, Attendance, Payroll, and their relationships.

Database Schemas

Define tables, fields, data types, constraints, and indexes for efficient data retrieval.

Data Security Measures

Outline encryption, access controls, and backup procedures to safeguard sensitive information.

- User Interface and User Experience (UI/UX) Design

Wireframes and Mockups

Provide visual sketches of key screens, such as login pages, dashboards, and data entry forms.

Navigation Flow

Explain how users interact with the system, emphasizing ease of use and accessibility.

Accessibility Considerations

Ensure compliance with standards for users with disabilities.

- Implementation Plan

Development Timeline

Break down the project into phases?planning, design, development, testing, deployment?with estimated durations.

Resource Allocation

Detail team roles, responsibilities, and required tools or environments.

Testing Strategies

Define unit testing, integration testing, user acceptance testing, and criteria for success.

- Deployment and Maintenance

Deployment Procedures

Steps for deploying the system into production, including data migration, user training, and documentation.

Post-Deployment Support

Plans for monitoring, bug fixing, and updating the system as organizational needs evolve.

Documentation and User Manuals

Guides for end-users and administrators to maximize system utilization.

Best Practices for Effective EMS Project Documentation

Crafting thorough documentation is as much about quality as quantity. Here are best practices to ensure your documentation adds value:

- Maintain Clarity and Simplicity: Use clear language, avoid jargon, and ensure explanations are accessible.
- Ensure Consistency: Standardize terminology, formatting, and templates across all documents.
- Involve Stakeholders: Gather input from end-users, technical staff, and management to ensure requirements are accurately captured.
- Update Regularly: Keep documentation current, especially after changes in scope or design.
- Use Visual Aids: Incorporate diagrams, flowcharts, and mockups to clarify complex processes.
- Prioritize Security and Confidentiality: Protect sensitive information within documentation.

Challenges in Documenting Employee Management System Projects

Despite its importance, project documentation faces common hurdles:

- Incomplete Requirements Gathering: Missing details can lead to gaps and rework.
- Over-documentation: Excessive or overly detailed documents can become cumbersome.
- Lack of Stakeholder Engagement: Insufficient input can result in misaligned expectations.
- Rapid Technological Changes: Keeping documentation up-to-date amidst evolving technologies requires discipline.

Overcoming these challenges demands a structured approach, continuous communication, and dedicated documentation practices.

Conclusion: The Pillar of Successful EMS Projects

A well-crafted employee management system project documentation is not merely a formality but a strategic asset that underpins the entire project lifecycle. It ensures clarity of purpose, aligns stakeholder expectations, guides technical development, and facilitates ongoing support. As organizations increasingly rely on digital solutions to manage their human capital, investing time and effort into comprehensive documentation becomes critical.

By adhering to best practices, involving stakeholders throughout, and maintaining updated records, organizations can develop EMS solutions that are robust, scalable, and aligned with organizational goals. Ultimately, thorough project documentation transforms a complex technical endeavor into an organized, manageable, and successful project, delivering tangible benefits in efficiency, accuracy,

and employee satisfaction.

QuestionAnswer What are the essential components to include in an employee management system project documentation? Key components include project overview, objectives, system architecture, database design, user roles and permissions, implementation details, testing strategies, and deployment guidelines. How does comprehensive documentation benefit the development and maintenance of an employee management system? Comprehensive documentation ensures clear understanding among developers, facilitates easier maintenance and updates, aids in onboarding new team members, and provides a reference for troubleshooting and future enhancements. What best practices should be followed when creating project documentation for an employee management system? Best practices include maintaining clarity and conciseness, using standardized formats, including diagrams and flowcharts, documenting all features and APIs, and regularly updating the documentation to reflect system changes. Which tools are commonly used for writing and managing employee management system project documentation? Popular tools include Microsoft Word, Google Docs, Markdown editors, Confluence, and version control systems like Git for collaborative documentation management. How can testing and validation procedures be effectively documented in an employee management system project? Testing procedures should be documented with detailed test cases, expected outcomes, testing environments, and validation steps, along with records of test results to ensure system reliability and compliance.

Related keywords: employee management system, project documentation, HR software, employee database, system design, implementation plan, user manual, technical specifications, development phases, project report

Openerp documentation

Understanding Openerp Documentation: Your Guide to Mastering Odoo

Openerp documentation is an essential resource for developers, business analysts, and users who wish to leverage the full potential of Odoo, formerly known as OpenERP. This comprehensive documentation provides detailed insights into the system's architecture, modules, customization options, and best practices for implementation and maintenance. Whether you are a beginner starting your journey or an experienced professional seeking advanced configurations, the Openerp documentation serves as your roadmap to understanding and optimizing the platform.

In this article, we will explore the various aspects of Openerp documentation, how to access and utilize it effectively, and why it is a crucial component for successful Odoo deployment.

What is OpenERP Documentation?

OpenERP documentation encompasses all the official and community-generated resources that elucidate how to install, configure, customize, and troubleshoot Odoo. It includes detailed guides, API references, tutorials, and FAQs designed to facilitate smooth implementation and operation of the software.

This documentation is typically organized into several key sections:

- Installation Guides
- User Manuals
- Developer Guides
- Module Documentation
- API References
- Troubleshooting and FAQs

The goal of OpenERP documentation is to empower users with the knowledge necessary to make informed decisions, customize the system to meet unique business needs, and maintain their Odoo environment efficiently.

Accessing OpenERP Documentation

The official OpenERP documentation is available through various online platforms:

Official Odoo Documentation Website

- The primary source for up-to-date and comprehensive documentation.
- Offers guides for different versions of Odoo.
- Includes tutorials, developer guides, and technical references.

Community Resources

- Forums such as Odoo Community Association (OCA).
- Blogs, tutorials, and YouTube channels dedicated to Odoo.

Third-Party Guides and Books

- Published books and e-books providing in-depth analysis.
- Paid courses and webinars.

Using Openerp Documentation Effectively

To maximize the benefits of Openerp documentation, consider the following approaches:

Identify Your Role and Needs

- End User: Focus on user manuals and basic workflows.
- Implementation Specialist: Refer to installation guides, configuration tutorials, and module documentation.
- Developer: Dive into API references, development guides, and customization tutorials.

Stay Updated

- Odoo frequently releases updates; always consult the latest documentation corresponding to your version.
- Join community forums to stay informed about common issues and solutions.

Leverage Tutorials and Examples

- Follow step-by-step tutorials to grasp complex configurations.
- Experiment in test environments before applying changes to production.

Key Sections of Openerp Documentation

Understanding the structure of the documentation helps in navigating efficiently. Here are the main sections:

1. Installation and Setup

- System requirements.
- Step-by-step installation procedures for different operating systems.
- Post-installation configuration.

2. User Guides

- How to navigate the interface.
- Managing sales, purchases, accounting, inventory, and other core modules.
- Tips for optimizing workflows.

3. Developer Documentation

- Custom module creation.
- Extending existing modules.
- API usage and best practices.

4. Module Documentation

- Detailed descriptions of each module.
- Configuration options.
- Integration points with other modules.

5. API Reference

- REST API endpoints.
- XML-RPC and JSON-RPC interfaces.
- Data models and methods.

6. Troubleshooting and FAQs

- Common issues and their solutions.
- Performance tuning.
- Backup and disaster recovery.

Best Practices for Reading and Using Openerp Documentation

To use Openerp documentation effectively, consider these best practices:

- Start with the Official Guides: They are the most reliable source of information and cover foundational concepts.
- Use Version-Specific Documentation: Always verify that the documentation matches your Odoo version to avoid discrepancies.

- Participate in Community Forums: Real-world tips and solutions often come from community discussions.
- Document Your Customizations: Keep track of changes made based on the documentation for future reference and troubleshooting.
- Practice in a Sandbox Environment: Before applying new configurations or custom modules, test thoroughly in a controlled environment.

Common Challenges and How the Documentation Helps

Implementing Odoo can present challenges such as configuring complex workflows, integrating third-party modules, or troubleshooting errors. OpenERP documentation addresses these issues by providing:

- Step-by-step troubleshooting guides
- Examples of common configurations
- Performance optimization tips
- Security best practices

By leveraging this information, users can resolve issues more efficiently and maintain system stability.

Contributing to OpenERP Documentation

The open-source nature of Odoo encourages community contributions to its documentation. If you develop new modules, fix existing documentation, or create tutorials, consider submitting your work to:

- Odoo Community Association (OCA)
- Official Odoo documentation repositories

Contributions help improve accuracy, cover new features, and assist other users worldwide.

Conclusion: Why OpenERP Documentation Is Indispensable

In summary, OpenERP documentation is a vital resource that unlocks the full potential of Odoo. It provides detailed instructions, best practices, and technical references necessary for successful deployment, customization, and maintenance. Whether you are a beginner or a seasoned developer, understanding how to navigate and utilize this documentation ensures your Odoo experience is efficient, scalable, and aligned with your business goals.

Investing time in mastering the OpenERP documentation not only accelerates your learning curve but also enhances your ability to troubleshoot, innovate, and optimize your Odoo environment. As the platform continues to evolve, staying engaged with the latest documentation ensures you remain at the forefront of Odoo implementation and development.

Remember: Regularly consult the official documentation and active community forums to stay updated with new features, security patches, and best practices. Your success with Odoo begins with a solid understanding of its documentation.

OpenERP Documentation: A Comprehensive Guide to Mastering the Open Source ERP Platform

In the rapidly evolving landscape of business management software, OpenERP?now known as Odoo?stands out as a powerful, flexible, and open-source Enterprise Resource Planning (ERP) solution. One of the critical factors contributing to its widespread adoption and success is its comprehensive and well-structured documentation. Whether you're a developer, a business analyst, or an end-user, understanding OpenERP's documentation is essential to harness its full potential. In this article, we'll delve into the various facets of OpenERP documentation, exploring its structure, features, and how it empowers users to deploy, customize, and optimize the platform effectively.

Understanding OpenERP Documentation: An Overview

OpenERP's documentation serves as the backbone for users seeking to understand, implement, and extend the platform. It encompasses a wide array of resources designed to cater to different user roles, including administrators, developers, and end-users.

Key features of OpenERP documentation include:

- Comprehensive coverage: From installation guides to advanced customization.
- Structured organization: Clear categorization by modules and user roles.
- Multi-format resources: Textual guides, tutorials, API references, and videos.
- Community-driven content: Contributions from a global community of developers and users.

This multi-layered approach ensures that users at all levels can find relevant, accurate, and up-to-date information.

The Structure of OpenERP Documentation

A well-organized documentation hierarchy is vital for ease of navigation and efficient learning. OpenERP's documentation is typically divided into several core sections:

- Installation and Setup Guides

These documents provide step-by-step instructions for deploying OpenERP in various environments?be it local servers, cloud-hosted solutions, or containerized setups like Docker. They include prerequisites, configurations, and troubleshooting tips.

- User Manuals

Targeted at end-users and business managers, these manuals explain how to operate different modules?such as Sales, Inventory, Accounting, and HR. They often feature screenshots, workflows, and best practices to ensure users can perform daily tasks effectively.

- Developer Documentation

This is a more technical section aimed at developers interested in customizing or extending OpenERP. It covers:

- Module development: Creating new modules and integrating them into the system.
- API references: Detailed documentation of the Open Object API (ORM), web controllers, and XML-RPC/JSON-RPC interfaces.
- Code architecture: Understanding the underlying framework, models, views, and workflows.

- API and Integration Guides

OpenERP supports integration with third-party systems through APIs and connectors. Documentation in this area explains how to connect OpenERP with other platforms, including REST APIs, payment gateways, and external data sources.

- Community Contributions and Tutorials

Given OpenERP?s open-source nature, community-contributed tutorials, blogs, and forums are integral to the ecosystem. These resources often provide practical insights, custom modules, and real-world use cases.

Key Features of OpenERP Documentation

To truly appreciate the value of OpenERP's documentation, it's important to understand its standout features:

- Clarity and Detail

The documentation strives for clarity, breaking down complex concepts into digestible steps. Diagrams, screenshots, and code snippets aid comprehension and practical application.

- Up-to-Date Content

Open source projects evolve rapidly. OpenERP's documentation is regularly updated to reflect new features, security patches, and platform improvements, ensuring users have access to current information.

- Multilingual Support

Given its global user base, OpenERP documentation is often translated into multiple languages, broadening accessibility and usability across different regions.

- Searchability and Indexing

A robust search function enables users to quickly locate relevant topics, while comprehensive indexes and cross-references facilitate seamless navigation across related content.

- Developer-Focused Resources

Detailed API references, code samples, and development workflows make it easier for developers to customize and extend the platform without extensive trial-and-error.

How OpenERP Documentation Facilitates Implementation and Customization

Implementing an ERP system like OpenERP can be a complex undertaking, but thorough documentation simplifies this process by providing:

- Step-by-Step Installation Guides

Clear instructions for deploying OpenERP on various environments, including Linux, Windows, and cloud platforms. These guides often include troubleshooting sections to address common pitfalls.

- Configuration and Setup Instructions

Guides on configuring databases, security settings, user roles, and module activation help tailor the system to specific organizational needs.

- Workflow and Process Documentation

Detailed descriptions of key business processes within modules enable users to understand how to optimize workflows and leverage system features fully.

- Custom Module Development

For organizations with unique requirements, the developer documentation provides comprehensive tutorials on creating custom modules, overriding default behaviors, and integrating third-party libraries.

- Data Migration and Integration

Guides on importing legacy data, synchronizing with external systems, and setting up APIs support seamless data flow and business continuity.

Utilizing OpenERP API Documentation for Developers

The API documentation is a cornerstone for technical customization. It provides:

- Model and Field References: Details on core data models, relationships, and field types.
- Methods and Functions: Information on available operations, including create, read, write, delete, and custom methods.
- Web Services: Guidance on exposing functionalities via XML-RPC, JSON-RPC, or RESTful APIs.
- Security and Access Control: Best practices for implementing user permissions and data

security within custom modules.

By leveraging this documentation, developers can create bespoke features, automate processes, and integrate OpenERP with other enterprise systems effectively.

Community and External Resources

While OpenERP's official documentation is comprehensive, the vibrant community surrounding the platform significantly enhances the learning and implementation experience.

Community Forums and Q&A

Platforms like Odoo's official forums, Stack Overflow, and Reddit host active discussions where users share solutions, tutorials, and best practices.

Tutorials and Blogs

Numerous online blogs and YouTube channels provide step-by-step tutorials on specific modules, customization techniques, and advanced configurations.

Add-on Modules and Plugins

The community regularly develops additional modules, which are documented on repositories like GitHub. These modules often come with their own documentation, expanding OpenERP's capabilities.

Conclusion: The Significance of Robust OpenERP Documentation

In the realm of enterprise software, where complexity and customization are often barriers to adoption, OpenERP's detailed and well-structured documentation plays a pivotal role. It not only accelerates deployment and onboarding but also empowers organizations to tailor the platform to their unique needs without extensive reliance on external consultants.

By providing clarity, up-to-date information, and extensive technical resources, OpenERP documentation transforms a complex ERP system into an accessible and adaptable tool for businesses of all sizes. Whether you're installing your first instance, customizing modules, or developing integrations, thorough documentation is your most valuable resource, guiding you every step of the way toward maximizing your investment in OpenERP.

In summary, OpenERP documentation is an essential pillar that supports the platform's flexibility, scalability, and widespread adoption. Its comprehensive nature, combined with active community

contributions, ensures that users—from novices to seasoned developers—have the resources necessary to leverage OpenERP's full potential, making it a standout solution in the open-source ERP landscape.

Question What is OpenERP documentation and where can I find it? OpenERP documentation provides comprehensive guides and references for installing, configuring, and customizing OpenERP (now Odoo). It is available on the official Odoo documentation website at <https://www.odoo.com/documentation>. How do I access the OpenERP documentation for different versions? You can access documentation for various OpenERP (Odoo) versions by selecting the version from the version dropdown menu on the official documentation site or visiting specific version URLs, such as <https://www.odoo.com/documentation/14.0/> for version 14. Does OpenERP documentation include developer guides? Yes, the OpenERP documentation includes developer guides covering module development, API usage, customization, and deployment to help developers build and extend the system effectively. What are the main topics covered in OpenERP documentation? The main topics include installation, configuration, user guides, module development, API reference, troubleshooting, and best practices for deploying and customizing OpenERP. How can I contribute to OpenERP (Odoo) documentation? You can contribute by submitting improvements or corrections via the official Odoo GitHub repository or community forums, following contribution guidelines provided on the Odoo documentation site. Is there a community forum or support for OpenERP documentation queries? Yes, the Odoo community forums and the official support channels are active platforms where users can ask questions and find help related to OpenERP documentation and usage. Are there video tutorials or online courses linked to OpenERP documentation? Yes, many online learning platforms and YouTube channels offer tutorials that complement the OpenERP documentation, providing visual guides for installation, customization, and development. How often is the OpenERP (Odoo) documentation updated? The documentation is regularly updated with new releases and features, typically aligned with new versions of Odoo, ensuring users have access to current and accurate information.

Related keywords: Odoo documentation, OpenERP guide, ERP manual, OpenERP tutorials, OpenERP setup, Odoo user guide, OpenERP developer documentation, OpenERP installation, Odoo modules, ERP system documentation

Read the full article online: [openerp documentation](#)