**A**

***PROJECT REPORT***

*on*

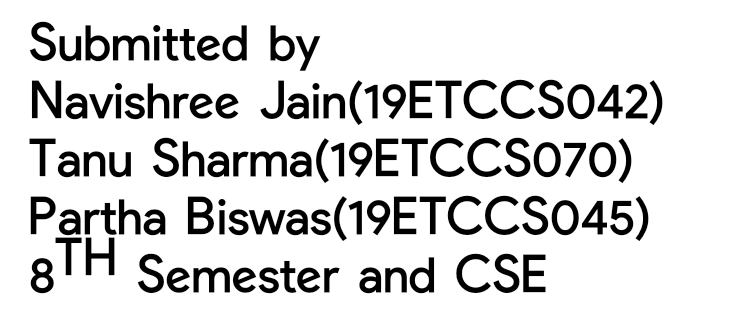
**Enterprise Resource Planning**

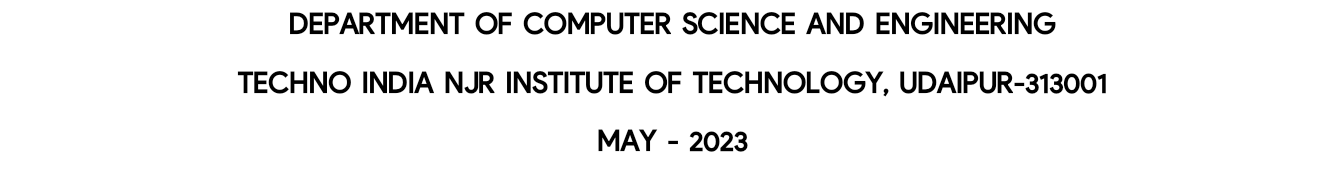
*Submitted in partial fulfilment of the requirements for the degree of*

**BACHELOR OF TECHNOLOGY**



**Session: - Jan-June 2023**







Department of Computer Science and Engineering

Techno India NJR Institute of Technology, Udaipur-313001

**Certificate**

This is to certify that project work titled Enterprise Resource Planning by Navishree Jain,Tanu sharma and Partha Biswas was successfully carried out in the Department of Computer Science and Engineering, TINJRIT and the report is approved for submission in the partial fulfillment of the requirements for award of degree of Bachelor of Technology in Computer Science and Engineering.

Aditya Maheshwari Dr. Rimpy Bishnoi

Head of Industry project HOD

CSE and TINJRIT Dept. of CSE TINJRIT, Udaipur Date...................... Date......................



Department of Computer Science and Engineering

Techno India NJR Institute of Technology, Udaipur-313001

**Examiner Certificate**

This is to certify that the following student Navishree Jain,Tanu sharma and Partha Biswas

of final year B.Tech. (Computer Science and Engineering), was examined for the project work titled

***Enterprise Resource Planning***

during the academic year 2022 – 2023 at Techno India NJR Institute of Technology, Udaipur

**Remarks:**

**Date:**

Signature Signature

(**Internal Examiner**) (**External Examiner**)

Name :- ……………………… Name :- ………………………

Designation:- ……………….. Designation:- ………………..

Department: - ………………. Department: - ……………….

Organization:- ……………… Organization:- ………………

Preface

ERP systems are the kind of software tools which are used to manage the data of an enterprise. ERP system helps different organizations to deal with different departments of an enterprise. Different departments like receiving, inventory management, customer order management, production planning, shipping, accounting, human resource management, and other business functions. Basically, it is the practice of consolidating an enterprise’s planning, its manufacturing, its sales and marketing efforts into one management system. It combines all databases across different departments into a single database which can be easily accessible to all employees of that enterprise.

In this report, we begin by laying the foundation, offering a clear definition of ERP and its fundamental components. We then proceed to examine the key modules and functionalities typically found in ERP systems, exploring how they interconnect and contribute to the overall management of an enterprise's resources. Moreover, we discuss the various deployment models, including on-premises, cloud-based, and hybrid solutions, highlighting their respective advantages and considerations.

While ERP systems bring numerous benefits, their implementation journey is not without challenges. This report addresses the potential hurdles that organizations may encounter during the planning, selection, and implementation phases, providing insights into effective strategies for successful ERP integration. Furthermore, we delve into change management aspects and the importance of organizational readiness, emphasizing the significance of aligning business processes with the chosen ERP solution.

Additionally, this report explores the impact of ERP systems on different functional areas within an organization, including finance, human resources, supply chain management, customer relationship management, and more. We discuss the potential improvements and operational efficiencies that can be achieved through the seamless integration of these departments.

**ACKNOWLEDGMENT**

We take this opportunity to record our sincere thanks to all who helped us to successfully complete this work. Firstly, We are grateful to our **supervisor Aditya Maheshwari** for his invaluable guidance and constant encouragement, support and most importantly for giving us the opportunity to carry out this work.

We would like to express our deepest sense of gratitude and humble regards to our

**Head of Department Dr. Rimpy Bishnoi** for giving invariable encouragement in our endeavors and providing necessary facility for the same. Also a sincere thanks to all faculty members of CSE, TINJRIT for their help in the project directly or indirectly.

Finally, We would like to thank my friends for their support and discussions that have proved very valuable for us. We are indebted to our parents for providing constant support, love and encouragement. We thank them for the sacrifices they made so that we could grow up in a learning environment. They have always stood by us in everything we have done, providing constant support, encouragement and love

**Navishree Jain & 19ETCCS042**

**Tanu Sharma & 19ETCCS070**

**Partha Biswas & 19ETCCS045**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

TECHNO INDIA NJR INSTITUTE OF TECHNOLOGY, UDAIPUR-313001

**CONTENTS**

| **Abstract** | ………………………………………………………………. | i |
| --- | --- | --- |
| **Contents** | ………………………………………………………………. | ii – iii |
| **List of Figures** ……………………………………………………............ | | iv – v |
| **List of Tables** ……………………………………………………………. | | vi |
| **List of Abbreviations Used**………………………………………………. | | vii – viii |
| **Chapter 1** | INTRODUCTION…………………………………………... | 1 |
|  | * 1. Background ………….……………………………. | 1 |
|  | 1.2 Objectives……………………………………………. | 2 |
|  | 1.3 Scope …………………..…………………………….. | 3 |
|  | 1.4 Methodology……………………………………………  1.5 Overview of ERP………………………………………  1.6 Components of ERP……………………………………  1.7 Advantages of ERP…………………………………….. | 4  5-7  8  9-10 |
| **Chapter 2** | SOFTWARE REQUIREMENT SPECIFICATION…………………………………………. | 11 |
|  | * 1. React……………………………………...…............. | 12 |
|  | * 1. Node.js……………………………………………….. | 13-15 |
|  | * 1. PostgresSql ……………………………………………   2. MongoDB…………………………………………….. | 16  17-18 |
| **Chapter 3** | ERP IMPLEMENTATION………………………………... | 19 |
|  | 3.1 Planning and selection…………………………………. | 20 |
|  | 3.2 Deployment Models…………………………………… | 21-22 |
|  | 3.3 Implementation Process………………………………… | 22-24 |
|  |  |  |
|  |  |  |

**LIST OF FIGURES**

| Fig 1-2 | Objectives of ERP…………………………………………....... | 2 |
| --- | --- | --- |
| Fig 1-5-1 | Before ERP…………………………………………………….. | 6 |
| Fig 1-5-2 | After ERP…………………..………………………………….. | 7 |
| Fig 1-6 | ERP components…………………………………………… | 9 |
| Fig 2-2 | Concepts of node.js………………………………………….. | 15 |
| Fig 2-4 | Introduction to MongoDB…………………………..………. | 18 |
| Fig 3-2 | ERP Deployment Models……………………………………... | 22 |
| Fig 3-3 | ERP Implementation Process……………………………..... | 24 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**List of Abbreviations/ Symbols**

| ERP | Enterprise Resource Planning |
| --- | --- |
| UAT | User Acceptance Testing |
| ROI | Return On Investment |
| KPI | Key Performance Indicator |
| CRM | Customer Relationship Management |
| SCM | Supply Chain Management |
| HRM | Human Resource Management |
| UI | User Interface |
| DOM | Document Object Model |
| RFP | Request for Proposal |
| RFI | Request for Information |
| MVCC | multi-version concurrency control |
| JSON | JavaScript Object Notation |
| HTTP | Hyper Text Transfer Protocol |
| RDBMS | Relational Database Management System |
|  |  |
|  |  |

**CHAPTER – I**

**INTRODUCTION**

|  |  |
| --- | --- |

**INTRODUCTION**

**1.1** **Background**

[Enterprise Resource Planning (ERP) systems](https://www.erpadvisorsgroup.com/erp-systems) are defined as “connected business applications used to automate business processes and manage resource data across the entire enterprise.” In other words, ERP systems are a conceptual framework for how organizations can automate processes across an enterprise.

 Different departments like receiving, inventory management, customer order management, production planning, shipping, accounting, human resource management, and other business functions. Basically, it is the practice of consolidating an enterprise’s planning, its manufacturing, its sales and marketing efforts into one management system. It combines all databases across different departments into a single database which can be easily accessible to all employees of that enterprise. It helps in automation of the tasks involved in performing a business process.

**1.2 Core Objectives**

Integration of real-time information:

The most important core objective of ERP is understanding real time information regarding various strategies of your business field. [ERP software](https://www.ebizframe.com/retail-erp/) package helps you integrate date and information in real time. Further, it assists in better planning as well as management of resources according to the needs of your organization. The most important benefits you obtain while using ERP system are better work flow, upgrading the required software modules and improved efficiency.

Controlling and managing establishments are also found to be effective at different locations. Now, ERP has turned out to be a significant tool package being utilized by many a number of multinational organizations with the intention to accomplish their long-term goals. ROI or return on investment is another essential core objective of an [ERP system](https://www.ebizframe.com/). With proper implementation of an ERP package, professionals will gain extra profit by using available resources. In short, ERP is supposed to be an appropriate tool adopted by the company owners to deliver desired output yet at an affordable cost.

If you would like to compete in the present global market field, it is important to have better resource management. Implementation of ERP will control different functions, thereby enhancing company efficiency. For the purpose of controlling and managing the data effectively, some latest technologies will be included in the ERP software package. If ERP is implemented as per the company’s requirements, then it will ensure you a high ROI. Previous to planning project, it becomes essential to design the project realistically as this will help employees maximize ROI practically.

Reducing inventory cost will be assumed to be one among the core objectives of implementing ERP. Knowing business requirements, customer needs and better order tracking by ERP will help in proper resource utilization and management. It will improve operational process and also maximize the rate of Return on investment. Improving accuracy of outcomes, enhancing the operation flexibility and enhancing productivity are some other core objectives of implementing ERP.



Fig 1.2 – Objectives Of ERP

**1.3 Scope**

ERP scope describes the extent of a project. Based on your goals you determine the high level initiatives and functional areas that define the project. ERP systems may have features for managing employee data, payroll processing, benefits administration, performance management, training and development, recruitment, and workforce planning. ERP systems may provide project management capabilities, including project planning, resource allocation, task management, milestone tracking, and project cost management.

**1.4 Methodology**

Project Initiation: In this phase, the organization establishes the project team, defines project objectives, outlines the scope, and identifies key stakeholders. The project team conducts a feasibility study to assess the organization's readiness for ERP implementation, including budgeting and resource allocation.

Planning and Analysis: This phase involves a thorough analysis of the organization's existing processes, systems, and data. The project team identifies requirements, defines business processes, and maps them to the functionalities offered by the ERP system. Gaps between the current state and desired future state are identified, and customization or configuration needs are determined.

System Design: Based on the requirements and process mapping, the project team designs the system architecture and configuration. This includes defining data structures, workflows, security roles, interfaces with other systems, and customizations. The team may also conduct system integration planning if the ERP system needs to be integrated with other software applications.

Development and Configuration: In this phase, the ERP system is set up and configured according to the design specifications. This involves defining master data, creating user interfaces, configuring modules, and customizing the system to meet specific business needs. Data migration plans are also developed to transfer data from legacy systems to the ERP system.

Testing: The ERP system undergoes rigorous testing to ensure its functionality, performance, and reliability. This includes unit testing, integration testing, system testing, and user acceptance testing. Test scripts and scenarios are created, and issues or bugs are identified, resolved, and retested.

Training: Before the ERP system is deployed, training programs are conducted to familiarize end-users with the new system. Training may cover basic system navigation, module-specific functionalities, data entry, and reporting. Training can be conducted through classroom sessions, online courses, user manuals, and hands-on exercises.

Deployment and Go-Live: Once testing and training are complete, the ERP system is deployed to the production environment. Data migration is performed, and the system is made available to end-users. The go-live process may involve parallel operations (running both old and new systems simultaneously), phased implementation (gradual rollout across different departments or locations), or a "big bang" approach (simultaneous switch to the new system).

Post-Implementation and Support: After the go-live, the ERP system enters a stabilization phase. The project team monitors system performance, addresses any post-implementation issues or user concerns, and provides ongoing support. Regular system audits and maintenance activities are performed, and continuous improvement initiatives are identified to optimize system utilization.

**1.5 Overview of ERP**

Here is an overview of ERP and its key characteristics:

Integration: ERP systems integrate multiple business functions, departments, and data sources into a unified platform. It eliminates data silos and enables seamless flow of information across the organization, promoting collaboration and visibility.

Modules and Functionality: ERP systems typically consist of various modules that cater to specific areas of business operations. Common modules include finance and accounting, human resources, supply chain management, manufacturing, customer relationship management, project management, and business intelligence.

Centralized Database: ERP systems maintain a centralized database where all relevant data is stored. This includes master data (such as customer, vendor, and employee information), transactional data, inventory data, and other critical business data. The centralized database ensures data consistency and provides a single source of truth for the organization.

Process Automation: ERP systems automate manual and repetitive tasks, reducing human errors and improving operational efficiency. They provide standardized workflows and process automation capabilities to streamline business processes, such as purchase-to-pay, order-to-cash, and record-to-report.

Reporting and Analytics: ERP systems offer reporting and analytics tools to generate meaningful insights from the organization's data. Users can create customizable reports, dashboards, and visualizations to monitor key performance indicators (KPIs), track business metrics, and make data-driven decisions.

Scalability and Flexibility: ERP systems are designed to accommodate the growth and changing needs of an organization. They can scale up or down to support small businesses as well as large enterprises. ERP systems also provide flexibility through customization options, allowing organizations to tailor the system to their specific requirements.

Compliance and Regulatory Support: ERP systems often include features to ensure compliance with industry-specific regulations, government policies, and financial standards. They facilitate adherence to legal requirements, data privacy regulations, and auditing processes.

Vendor Ecosystem: ERP systems are typically provided by software vendors who offer implementation, customization, training, and support services. These vendors may also have a marketplace of add-on modules, integrations, and third-party applications that enhance the functionality of the ERP system.

Cloud-Based and On-Premises Deployments: ERP systems can be deployed on-premises, where the organization hosts the software and infrastructure internally, or they can be cloud-based, where the ERP system is hosted and maintained by the vendor on remote servers. Cloud-based ERP offers advantages such as lower upfront costs, easier scalability, and accessibility from anywhere with an internet connection.

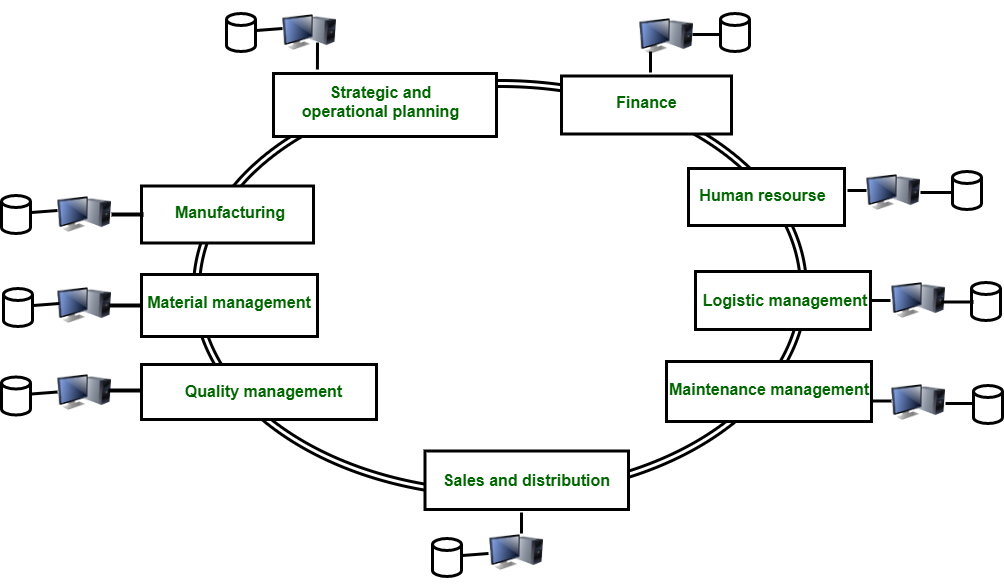


Fig 1.5.1 – Before ERP

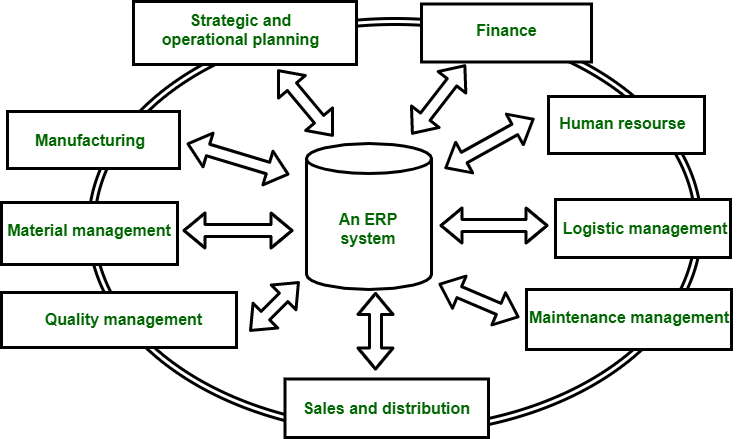


Fig 1.5.2 – After ERP

**1.6 Components of ERP**

Financial Management: This component focuses on managing financial processes such as general ledger, accounts payable and receivable, budgeting, cash management, and financial reporting. It helps track and analyze financial data across the organization.

Supply Chain Management (SCM): SCM involves the management of the entire supply chain, including procurement, inventory management, order fulfillment, logistics, and supplier relationship management. It aims to optimize the flow of goods and services from suppliers to customers.

Customer Relationship Management (CRM): CRM component focuses on managing customer interactions, sales, and marketing activities. It helps organizations maintain customer data, track leads and opportunities, manage marketing campaigns, and enhance customer satisfaction and loyalty.

Human Resource Management (HRM): HRM component handles employee-related processes such as employee data management, payroll, benefits administration, performance management, recruitment, training, and workforce planning. It helps streamline HR processes and supports effective workforce management.

Manufacturing and Production Management: This component deals with managing the manufacturing and production processes, including production planning, scheduling, inventory control, shop floor management, quality control, and product lifecycle management. It helps optimize manufacturing operations and improve efficiency.

Project Management: The project management component assists in planning, organizing, and tracking project activities, resources, timelines, and budgets. It allows organizations to manage projects effectively, allocate resources efficiently, and monitor project progress.

Business Intelligence and Reporting: This component provides tools for data analysis, reporting, and decision-making. It helps organizations extract insights from data collected within the ERP system and generate reports, dashboards, and key performance indicators (KPIs) for informed decision-making.

Integration and Data Management: ERP systems rely on integration capabilities to connect with various internal and external systems, ensuring smooth data flow across different functional areas. Data management ensures data integrity, consistency, and security within the ERP system.

Mobile and E-commerce: Some modern ERP systems offer mobile and e-commerce capabilities, allowing users to access and interact with the ERP system through mobile devices or enabling online transactions with customers and suppliers.

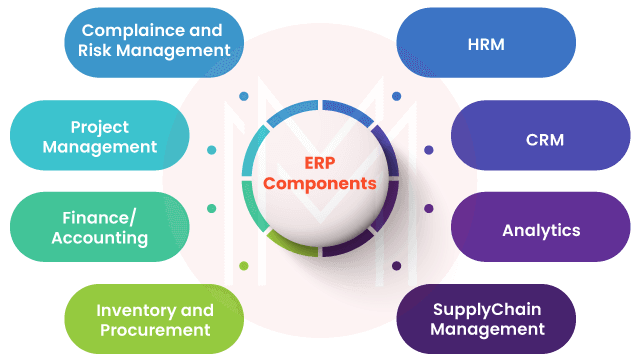


Fig 1.6 – ERP components

**1.7 Advantages of ERP**

Streamlined Processes: ERP integrates and automates various business processes across departments, eliminating manual tasks, redundant data entry, and paper-based processes. It enables streamlined workflows and improves operational efficiency.

Centralized Data: ERP provides a single source of truth for data, consolidating information from different functional areas into a unified database. This centralization enhances data accuracy, consistency, and reliability, allowing users to access real-time, up-to-date information.

Improved Collaboration: ERP promotes collaboration and information sharing across departments by providing a common platform. It enables employees to access and share data, documents, and reports, fostering better communication and collaboration within the organization.

Enhanced Decision-making: ERP systems offer robust reporting and analytics capabilities, providing users with insights and actionable information. It enables data-driven decision-making, helps identify trends, monitor key performance indicators (KPIs), and supports strategic planning.

Better Customer Relationship Management: ERP includes Customer Relationship Management (CRM) functionalities that help organizations manage customer interactions, track sales opportunities, and provide personalized services. It improves customer satisfaction, strengthens relationships, and increases customer retention.

Improved Supply Chain Management: ERP systems facilitate effective supply chain management by optimizing procurement, inventory management, demand forecasting, and order fulfillment. It ensures timely delivery of products or services, minimizes stockouts, and optimizes inventory levels.

Increased Productivity: Automation of manual tasks, streamlined processes, and reduced data entry lead to improved productivity. Employees can focus on value-added activities, while ERP handles routine tasks, reducing errors and improving efficiency.

Regulatory Compliance: ERP systems often include built-in compliance features to help organizations adhere to industry regulations and standards. It simplifies regulatory reporting, facilitates audits, and ensures data security and privacy.

Scalability and Flexibility: ERP systems are designed to accommodate organizational growth and changes. They can scale up to handle increased transaction volumes, additional users, and expanding operations. ERP also offers flexibility to adapt to changing business requirements and integrate with other systems.

Cost Savings: Although implementing an ERP system requires an initial investment, it can result in long-term cost savings. ERP eliminates duplicate systems, reduces manual errors, optimizes inventory, and improves resource utilization, leading to cost efficiencies and improved financial management.

**CHAPTER – II**

**SOFTWARE REQUIREMENT SPECIFICATION**

**SOFTWARE REQUIREMENT SPECIFICATION**

Now we will be describing all the languages, platforms tools and technologies are used in order to complete this project.

**2.1 React**

React is an open-source JavaScript library for building user interfaces (UIs). It was developed by Facebook and is widely used for creating interactive and dynamic web applications.

**Features of React**

Component-Based Architecture: React follows a component-based approach, where UIs are divided into reusable and independent components. Each component manages its own state and can be composed together to build complex user interfaces.

Virtual DOM: React uses a virtual representation of the actual DOM (Document Object Model) called the Virtual DOM. It efficiently updates and renders only the necessary parts of the UI, resulting in improved performance.

Declarative Syntax: React uses a declarative syntax, allowing developers to describe how the UI should look based on its current state. Developers focus on what the UI should be, and React takes care of updating the UI to match the desired state.

Efficient Rendering: React optimizes rendering by performing a diffing algorithm that efficiently updates only the necessary changes in the UI. This minimizes the number of DOM manipulations, resulting in better performance.

Component Lifecycle Methods: React provides lifecycle methods that allow developers to control and perform actions at different stages of a component's lifecycle. This includes mounting, updating, and unmounting components.

State Management: React allows the management of state within components. State represents the data that can change over time, and React provides mechanisms to update and handle state changes, triggering UI updates accordingly.

Component Reusability: React promotes component reusability, allowing developers to create modular components that can be reused across different parts of an application. This leads to code efficiency and easier maintenance.

Ecosystem and Community: React has a vast ecosystem of libraries, tools, and community support. This includes libraries for state management (e.g., Redux, MobX), routing (e.g., React Router), form handling (e.g., Formik), and many more. The active community provides resources, tutorials, and discussions to assist developers.

Cross-platform Development: React can be used for building not only web applications but also mobile applications using React Native. React Native allows developers to write code once and deploy it on multiple platforms, such as iOS and Android.

Integration and Compatibility: React can be easily integrated with existing projects and frameworks. It can coexist with other libraries or frameworks, allowing gradual adoption and migration.

**Who uses React ?**

* Large Technology Companies
* E-commerce Platforms
* Financial Services
* Startups and Small Businesses
* Educational Institutions

2**.2 Node.js**

Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009 and its latest version is v0.10.36 Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux. Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

Node.js = Runtime Environment + JavaScript Library 12

**Features of Node.js:**

Following are some of the important features that make Node.js the first choice of software architects.

Asynchronous and Event Driven- All APIs of Node.js library are asynchronous, that is, nonblocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.

Very Fast- Being built on Google Chrome's V8 JavaScript Engine; Node.js library is very fast in code execution.

Single Threaded but Highly Scalable- Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non- blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.

No Buffering − Node.js applications never buffer any data. These applications simply output the data in chunks.

License − Node.js is released under the MIT license

**Who uses Node.js?**

Node.js is used by eBay, General Electric, Go Daddy, Microsoft, PayPal, Uber, Wiki pins, Yahoo!, and Yammer to name a few. Projects, Applications, and Companies Using Node.

**Concepts:**

The following diagram depicts some important parts of Node.js which we will discuss in detail in the subsequent.

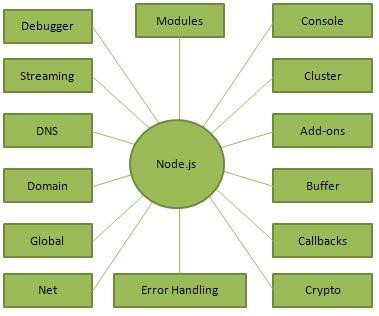


Fig 2.2 – Concepts of Node.js

**Where do we use Node.js?**

Following are the areas where Node.js is proving itself as a perfect technology partner.

I/O bound Applications.

Data Streaming Applications.

Data Intensive Real-time Applications (DIRT).

JSON APIs based Applications.

Single Page Applications.

It is not advisable to use Node.js for CPU intensive applications.

**2.3 PostgreSQL**

PostgreSQL, often referred to as Postgres, is an open-source relational database management system (RDBMS) known for its robustness, scalability, and extensive feature set. It provides support for a wide range of applications and is widely used in various industries.

**Features of PostgreSQL**

Relational Database: PostgreSQL is a relational database system based on the SQL (Structured Query Language) standard. It organizes data into tables with rows and columns, enabling efficient storage and retrieval of structured data.

Open-Source: PostgreSQL is an open-source project, which means it is freely available and can be used, modified, and distributed by anyone. The open-source nature fosters a strong community of developers and users, resulting in regular updates, bug fixes, and enhancements.

Advanced Features: PostgreSQL offers a rich set of advanced features. It supports ACID (Atomicity, Consistency, Isolation, Durability) properties to ensure data integrity and reliability. It provides support for complex queries, joins, views, triggers, stored procedures, and full-text search.

Scalability and Performance: PostgreSQL is designed to handle high-volume and concurrent workloads. It supports multi-version concurrency control (MVCC) to manage concurrent transactions effectively. Additionally, it provides various performance optimization techniques such as indexing, query optimization, and parallel processing.

Data Types and Extensions: PostgreSQL offers a wide range of built-in and user-defined data types, including numeric, string, date/time, array, JSON, and spatial data types. It also supports extensions, allowing users to add custom functionality and extend the capabilities of the database.

Data Integrity and Constraints: PostgreSQL provides robust data integrity features, including primary keys, foreign keys, unique constraints, check constraints, and triggers. These mechanisms ensure data consistency and enforce business rules and relationships.

Extensibility: PostgreSQL allows the creation of custom data types, operators, functions, and procedural languages using various programming languages such as SQL, PL/pgSQL, PL/Python, PL/Perl, and more. This extensibility enables developers to tailor the database to specific application needs.

High Availability and Replication: PostgreSQL supports various replication and high availability mechanisms, including streaming replication, logical replication, and synchronous and asynchronous replication. These features ensure data redundancy, fault tolerance, and availability in distributed and clustered environments.

Security: PostgreSQL emphasizes security features to protect sensitive data. It supports secure communication protocols, user authentication, role-based access control (RBAC), row-level security, and encryption of data at rest and in transit.

Cross-Platform Compatibility: PostgreSQL is cross-platform and runs on various operating systems, including Windows, Linux, macOS, and Unix-like systems. It can be easily integrated with different programming languages and frameworks.

**2.4 MongoDB**

MongoDB is an open-source document database and leading No SQL database. MongoDB is written in C++. This tutorial will give you great understanding on MongoDB concepts needed to create and deploy a highly scalable and performance-oriented database.

**Audience:**

This tutorial is designed for Software Professionals who are willing to learn MongoDB Database in simple and easy steps. It will throw light on MongoDB concepts and after completing this tutorial you will be at an intermediate level of expertise, from where you can take yourself at higher level of expertise.

**Prerequisites:**

Before proceeding with this tutorial, you should have a basic understanding of database, text editor and execution of programs, etc. Because we are going to develop high performance database, so it will be good if you have an understanding on the basic concepts of Database (RDBMS).

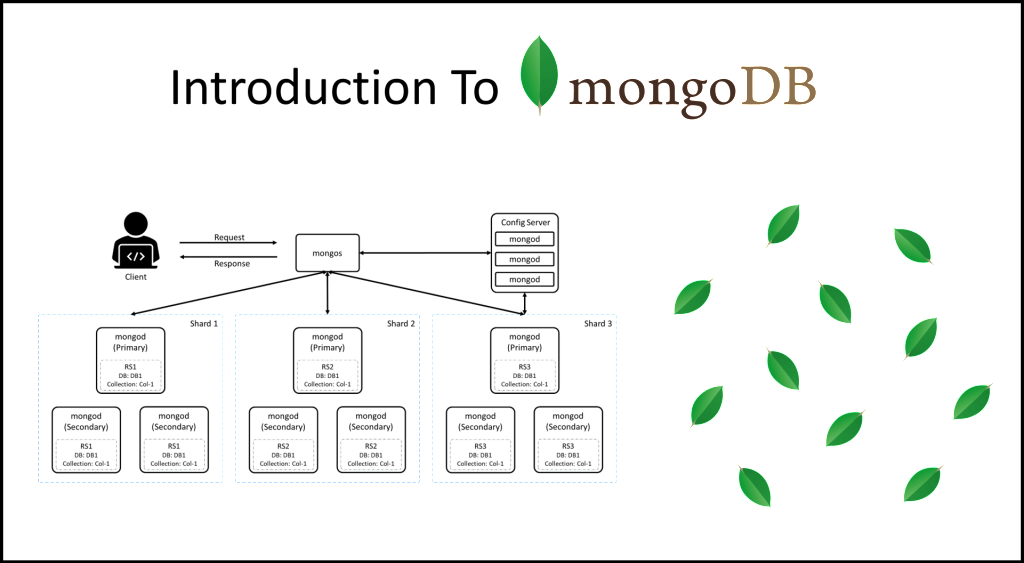
****

Fig 2.4 – Introduction to MongoDB

**CHAPTER – III**

**ERP IMPLEMENTATION**

**ERP IMPLEMENTATION**

ERP implementation refers to the process of deploying an Enterprise Resource Planning (ERP) system within an organization. It involves a series of steps and activities to ensure a successful adoption of the ERP system

**3.1 Planning and Selection**

* Define Objectives and Requirements:

Identify the goals and objectives of implementing an ERP system. Determine what areas of the organization will be impacted and the desired outcomes.

Conduct a thorough analysis of the organization's current business processes, systems, and pain points to define the specific requirements for the ERP system.

* Engage Stakeholders and Form a Project Team:

Involve key stakeholders from different departments in the ERP implementation project. This includes representatives from management, IT, finance, operations, and other relevant departments.

Form a cross-functional project team with members who have a good understanding of the organization's processes and requirements.

* Research and Evaluation:

Conduct market research to identify ERP vendors and solutions that align with the organization's industry, size, and requirements.

Evaluate ERP systems based on factors such as functionality, scalability, customization options, ease of use, vendor support, implementation cost, and ongoing maintenance.

Consider factors like technology compatibility, integration capabilities with existing systems, and the vendor's reputation and track record.

Request for Proposal (RFP) or Request for Information (RFI):

Prepare an RFP or RFI document to gather detailed information from ERP vendors.

Clearly define the organization's requirements, expectations, timelines, and evaluation criteria in the document.

Send the RFP/RFI to selected ERP vendors and review their responses to shortlist potential candidates.

* Vendor Demonstrations and Proof of Concept:

Invite shortlisted ERP vendors to provide live demonstrations of their systems. This allows stakeholders to visualize how the ERP system will address the organization's specific needs.

Consider conducting proof-of-concept exercises, where the ERP vendors demonstrate how their solution can handle real-life scenarios and specific requirements of the organization.

* Vendor Selection:

Evaluate the ERP vendors based on their responses, demonstrations, proof of concept, and their ability to meet the organization's requirements.

Consider factors such as vendor credibility, financial stability, customer references, implementation methodology, and post-implementation support.

Collaborate with the project team and key stakeholders to make an informed decision on selecting the ERP vendor that best fits the organization's needs.

* Create an Implementation Plan:

Develop a comprehensive implementation plan with clear timelines, milestones, and responsibilities.

Identify potential risks and develop mitigation strategies.

Allocate resources, both human and financial, to support the implementation process.

* Change Management and Communication:

Establish a change management strategy to prepare employees for the ERP implementation and address their concerns and resistance.

Communicate the benefits and goals of the ERP system to employees and stakeholders.

Provide regular updates and create a communication plan to keep everyone informed about the implementation progress.

* Data Preparation and Migration:

Analyze the existing data and determine the data cleansing, transformation, and migration requirements.

Develop a data migration plan, ensuring data accuracy and integrity during the transition to the new ERP system.

Test the data migration process and perform trial runs to identify and address any issues or inconsistencies.

* Training and User Adoption:

Develop a comprehensive training program to educate end-users on how to use the new ERP system effectively.

Conduct training sessions, workshops, and provide user documentation to support the learning process.

Encourage user adoption and provide ongoing support to address any challenges or questions.

**3.2 Deployment models**

* On-Premises Deployment:

In an on-premises deployment, the ERP system is installed and maintained locally on the organization's own servers and IT infrastructure.

The organization has full control over the hardware, software, and data security.

On-premises deployments often require substantial upfront investment in hardware, software licenses, and infrastructure.

The organization is responsible for managing system updates, maintenance, backups, and security.

On-premises deployments are suitable for organizations that require complete control over their data, have specific compliance requirements, or have significant IT infrastructure and resources.

* Cloud-Based Deployment:

In a cloud-based deployment, the ERP system is hosted and managed by a third-party cloud service provider.

The organization accesses and uses the ERP system over the internet through web browsers or dedicated applications.

The cloud service provider handles infrastructure management, system updates, maintenance, backups, and security.

Cloud-based deployments typically involve subscription-based pricing models, where organizations pay for the resources and services they use.

Cloud deployments offer scalability, flexibility, and accessibility from anywhere with an internet connection.

Organizations can reduce upfront costs and IT infrastructure requirements but rely on the cloud provider for data security and system availability.

* Hybrid Deployment:

A hybrid deployment combines elements of both on-premises and cloud-based deployments.

In this model, certain components or functionalities of the ERP system are hosted on-premises, while others are hosted in the cloud.

The hybrid approach allows organizations to leverage the benefits of both deployment models, such as maintaining sensitive data on-premises while utilizing cloud services for scalability and flexibility.

It requires integration and coordination between the on-premises and cloud-based components to ensure seamless data flow and functionality.

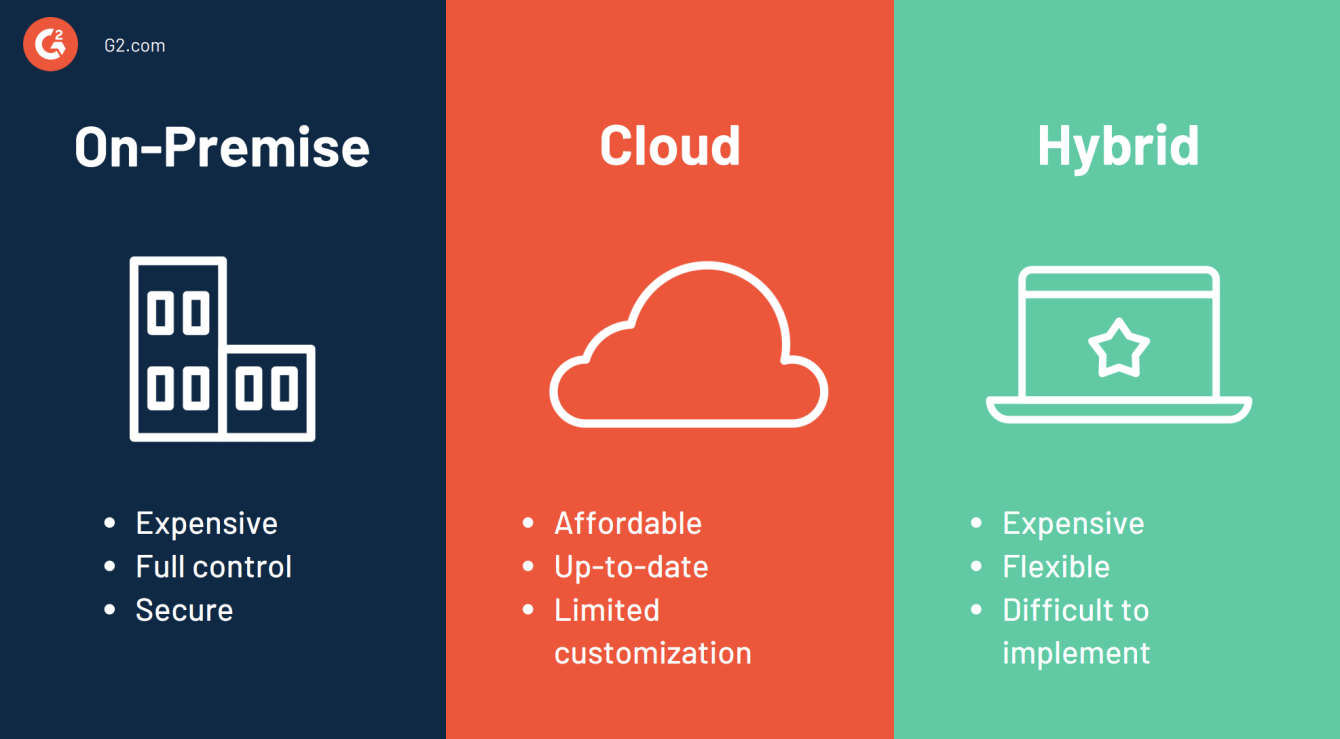
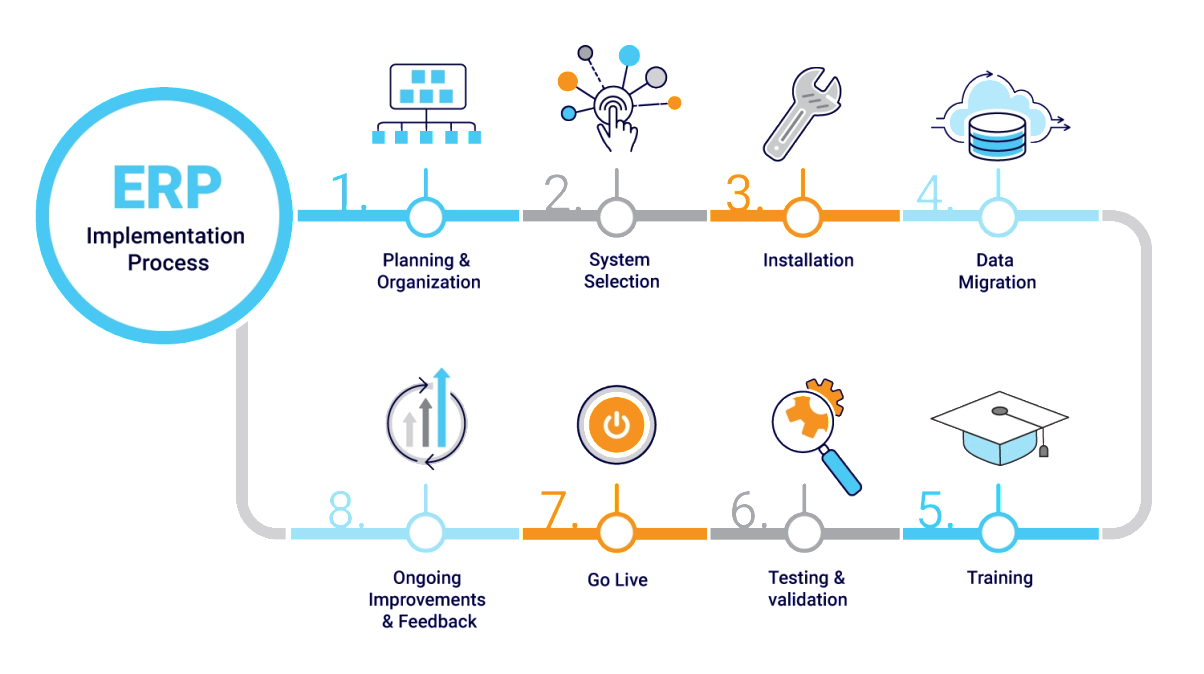


Fig 3.2 – ERP Deployment Models

**3.3 ERP Implementation process**

The ERP implementation process involves several stages that organizations go through to successfully deploy an Enterprise Resource Planning (ERP) system. While the specific steps may vary depending on the organization and ERP solution, here is a general outline of the ERP implementation process:

1. **Project Initiation and Planning**:
   * Define the project scope, objectives, and deliverables.
   * Establish the project team and roles, including project manager, functional leads, technical experts, and end-user representatives.
   * Create a detailed project plan, including timelines, milestones, and resource allocation.
   * Conduct a risk assessment and develop a risk management strategy.
2. **Business Process Analysis and Design**:
   * Analyze existing business processes, workflows, and system requirements.
   * Identify areas for improvement and define future-state processes aligned with ERP capabilities.
   * Determine integration needs with other systems and map data flow across the organization.
3. **System Configuration and Customization**:
   * Configure the ERP system to align with the organization's specific requirements.
   * Customize the system if necessary, considering the impact on future upgrades and system compatibility.
   * Develop and test any required customizations, reports, or interfaces.
4. **Data Migration**:
   * Plan and execute the migration of data from legacy systems to the new ERP system.
   * Cleanse and validate data to ensure accuracy and integrity.
   * Perform data conversion and mapping, ensuring compatibility with the ERP system's data structures.
5. **System Testing**:
   * Conduct various levels of testing, including unit testing, integration testing, and system testing.
   * Validate the ERP system's functionality, data accuracy, and performance.
   * Involve end-users in UAT to ensure the system meets their needs.
6. **Training and Change Management**:
   * Develop a comprehensive training program for end-users, administrators, and IT staff.
   * Conduct training sessions, workshops, and provide user documentation.
   * Implement change management strategies to prepare employees for the transition, address resistance, and ensure user adoption.
7. **Deployment and Go-Live**:
   * Prepare the production environment for the ERP system's launch.
   * Execute the cutover plan to transition from legacy systems to the new ERP system.
   * Perform final data verification and ensure system readiness for production use.
8. **Post-Implementation Support**:
   * Monitor the system's performance and address any immediate issues or concerns.
   * Provide ongoing support to end-users, addressing their questions and providing guidance.
   * Conduct post-implementation reviews to evaluate the success of the implementation, identify areas for improvement, and make necessary adjustments.
9. **System Optimization and Enhancements**:
   * Continuously monitor and optimize system performance.
   * Identify areas for system enhancements or additional functionalities.
   * Plan and implement system upgrades, patches, and new releases as provided by the ERP vendor.

Fig 3.3 – ERP Implementation Process

**Appendices**

**App.js**

class App extends Component {

state = {

data: {},

loading: false,

pass: true,

isLogin: false,

firstTimeAlert: true,

};

componentDidMount() {

this.setState({

data: {

\_id: localStorage.getItem("\_id") || "",

Account: localStorage.getItem("Account") || "",

Name: localStorage.getItem("Name") || ""

},

isLogin: localStorage.getItem("isLogin") == "true"

}, () => {

// this.alertFirstTime()

});

}

render() {

return (

< Router >

<Switch>

<Route exact path="/login" render={props => this.state.data["Account"] == 1 ? ( <Redirect to="/admin" />) :

this.state.data["Account"] == 2 ? (<Redirect to="/hr" />) : this.state.data["Account"] == 3 ? ( <Redirect to="/employee" />) : (

<Login onSubmit={this.handleSubmit}

loading={this.state.loading}

pass={this.state.pass}

/>

)

}

/>

<Route path="/admin" render={props => this.state.data["Account"] == 1 ? (

<DashboardAdmin data={this.state.data} onLogout={this.handleLogout}/>) : (<Redirect to="/login" />)}

/>

<Route

// exact

path="/hr"

render={props =>

this.state.data["Account"] == 2 ? (

<DashboardHR

data={this.state.data}

onLogout={this.handleLogout}

/>

) : (

<Redirect to="/login" />

)

}

/>

<Route

// exact

path="/employee"

render={props =>

this.state.data["Account"] == 3 ? (

<DashboardEmployee

data={this.state.data}

onLogout={this.handleLogout}

/>

) : (

<Redirect to="/login" />

)

}

/>

<Redirect to="/login" />

</Switch>

</Router >

);

}

handleSubmit = event => {

event.preventDefault();

this.setState({ pass: true });

this.setState({ loading: true });

this.login(event.target[0].value, event.target[1].value);

event.target.reset();

};

handleLogout = event => {

console.log("logout");

localStorage.clear();

this.componentDidMount();

};

login = (id, pass) => {

let bodyLogin = {

email: id,

password: pass

};

axios

.post(process.env.REACT\_APP\_API\_URL + "/api/login", bodyLogin)

.then(res => {

console.log(decodedData.Account);

console.log(jwt.decode(res.data));

var decodedData = jwt.decode(res.data);

localStorage.setItem("token", res.data);

if ((res == undefined || res == null || decodedData.Account == undefined || decodedData.Account == null) && !( decodedData.Account == 1 ||

decodedData.Account == 2 || decodedData.Account == 3 )) {

this.setState({ pass: false });

this.setState({ loading: false });

} else {

if (decodedData.Account == 1) {

this.setState({ pass: true });

this.setState({ loading: false });

this.setState({ isLogin: true });

localStorage.setItem("isLogin", true);

localStorage.setItem("Account", 1);

localStorage.setItem("\_id", decodedData["\_id"]);

localStorage.setItem(

"Name",

decodedData["FirstName"] + " " + decodedData["LastName"]

);

this.componentDidMount();

history.push("#/admin/role");

}

if (decodedData.Account == 2) {

this.setState({ pass: true });

this.setState({ loading: false });

this.setState({ isLogin: true });

localStorage.setItem("isLogin", true);

localStorage.setItem("Account", 2);

localStorage.setItem("\_id", decodedData["\_id"]);

localStorage.setItem(

"Name",

decodedData["FirstName"] + " " + decodedData["LastName"]

);

this.componentDidMount();

history.push("#/hr/employee");

}

if (decodedData.Account == 3) {

this.setState({ pass: true });

this.setState({ loading: false });

this.setState({ isLogin: true });

localStorage.setItem("isLogin", true);

localStorage.setItem("Account", 3);

localStorage.setItem("\_id", decodedData["\_id"]);

localStorage.setItem(

"Name",

decodedData["FirstName"] + " " + decodedData["LastName"]

);

this.componentDidMount();

history.push("#/employee/" + decodedData.\_id + "/personal-info");

}

}

})

.catch(err => {

this.setState({ pass: false });

this.setState({ loading: false });

}); }; }

export default App;

**Login.js**

class Login extends Component {

render() {

return (

<div>

<div className="container">

<div id="main-outer-div">

<div id="logo-div">

<img id="logo-img" src={Logo} alt="" />

</div>

<div id="title-div">

<h4 className="title">Sign in</h4>

</div>

<div id="outer-login-form-div">

<form action="" method="" onSubmit={this.props.onSubmit}>

<input className="login-form-input" type="text" placeholder="Email" required="required" name="Username"/>

<input className="login-form-input" type="password" placeholder="Password" required="required"/>

<input className="login-form-input" type="submit" value="Sign in" id="submitBtn" />

{!this.props.pass ? (<p className="alert">Invalid UserName or Password</p>) : ("")}

</form>

</div>

<div className="loading">

<ScaleLoader

css={override}

sizeUnit={"px"}

size={150}

color={"#123abc"}

loading={this.props.loading}

/>

</div>

</div>

</div>

</div>

);

} }

export default Login;

**AdminPortal.js**

class AdminPortal extends Component {

state = {

table: true,

editForm: false,

editData: {},

addFormStatus: "",

editFormStatus: ""

};

render() {

return (

<React.Fragment>

{this.state.table ? (

this.state.editForm ? (

<AdminPortalFormEdit

onPortalEditUpdate={this.handlePortalEditUpdate}

onFormEditClose={this.handleEditFormClose}

editData={this.state.editData}

onStatusChange={this.handleEditFormStatusChange}

/>

) : (

<AdminPortalTable

onAddPortal={this.handleAddPortal}

onEditPortal={this.handleEditPortal}

/>

)

) : (

<AdminPortalForm

onPortalSubmit={this.handlePortalSubmit}

onFormClose={this.handleFormClose}

onStatusChange={this.handleAddFormStatusChange}

/>

)}

</React.Fragment>

);

}

handlePortalSubmit = event => {

event.preventDefault();

console.log("portal", event.target[0].value, event.target[1].value, event.target[2].value);

console.log("portal status", this.state.addFormStatus);

this.setState({ table: true });

let body = {

PortalName: event.target[0].value,

Status: this.state.addFormStatus

};

axios

.post(process.env.REACT\_APP\_API\_URL + "/api/admin/portal", body, {

headers: {

authorization: localStorage.getItem("token") || ""

}

})

.then(res => {

this.setState({ table: false });

this.setState({ table: true });

})

.catch(err => {

console.log(err);

});

};

handleAddPortal = () => {

console.log("clicked1");

this.setState({ table: false });

};

handleEditPortal = e => {

console.log(e);

console.log("clicked6");

this.setState({ editForm: true });

this.setState({ editData: e });

this.setState({ editFormStatus: e["Status"] });

};

handleFormClose = () => {

console.log("clicked1");

this.setState({ table: true });

};

handleEditFormClose = () => {

console.log("clicked5");

this.setState({ editForm: false });

};

handleFormClose = () => {

console.log("clicked1");

this.setState({ table: true });

};

handleAddFormStatusChange = (e) => {

this.setState({

addFormStatus: e.currentTarget.value

});

};

handleEditFormStatusChange = (e) => {

this.setState({

editFormStatus: e.currentTarget.value

});

};

handlePortalEditUpdate = (info, formData1) => {

let body = {

\_id: info["\_id"],

PortalName: formData1,

Status: this.state.editFormStatus,

ID: info["ID"],

};

console.log("update", body);

axios

.put(process.env.REACT\_APP\_API\_URL + "/api/admin/portal/" + info["ID"], body, {

headers: {

authorization: localStorage.getItem("token") || ""

}

})

.then(res => {

this.setState({ table: false });

this.setState({ table: true });

})

.catch(err => {

console.log(err);

});

this.setState({ editForm: false });

};

}

export default AdminPortal;

**DashboardEmployee.js**

import React, { Component } from "react";

import "./DashboardEmployee.css";

import { HashRouter as Router, Route, Link } from "react-router-dom";

import { Switch } from "react-router";

import { Redirect } from "react-router-dom";

import NavBar from "../NavBar.jsx";

import PersonalInfo from "./PersonalInfo.jsx";

import Education from "./Education.jsx";

import FamilyInfo from "./FamilyInfo.jsx";

import WorkExperience from "./WorkExperience.jsx";

import LeaveApplicationEmp from "./LeaveApplicationEmp.jsx";

import NotFound404 from "../NotFound404.jsx";

import { FontAwesomeIcon } from "@fortawesome/react-fontawesome";

import {

faUsers,

faUser,

faFileAlt,

faUniversity,

faBriefcase,

faMale,

} from "@fortawesome/free-solid-svg-icons";

class DashboardHR extends Component {

state = {

redirect: true,

checked: true

};

handleChange = (checked) => {

if (this.state.checked == true) {

document.getElementById("sidebar").setAttribute("class", "display-none");

}

else { document.getElementById("sidebar").setAttribute("class", "display-block"); }

this.setState({ checked });}

render() {

return (

<Router>

<div id="outer-main-div">

<div id="outer-nav">

<NavBar

loginInfo={this.props.data}

checked={this.state.checked}

handleChange={this.handleChange}

onLogout={this.props.onLogout}

/>

</div>

<div id="main-non-nav">

<div id="sidebar">

<div id="sidebar-top-content" />

<div id="main-title" className="main-title-employee">

<FontAwesomeIcon icon={faUsers} className="sidebar-icon" />

Employee

</div>

<ul className="navbar-ul">

<li>

<Link

to={

"/employee/" +

this.props.data["\_id"] +

"/personal-info"

}

>

<FontAwesomeIcon

icon={faUser}

className="sidebar-icon"

/>

Personal Information

</Link>

</li>

<li>

<Link

to={

"/employee/" + this.props.data["\_id"] + "/education" }

> <FontAwesomeIcon

icon={faUniversity}

className="sidebar-icon"

/>

Education

</Link>

</li>

<li>

<Link to={

"/employee/" + this.props.data["\_id"] + "/family-info"

}>

<FontAwesomeIcon

icon={faMale}

className="sidebar-icon"

/>

Dependents

</Link>

</li>

<li>

<Link to={

"/employee/" + this.props.data["\_id"] + "/work-experience"

}>

<FontAwesomeIcon

icon={faBriefcase}

className="sidebar-icon"

/>

WorkExp

</Link>

</li>

<li>

<Link to={

"/employee/" + this.props.data["\_id"] + "/leave-application-emp"

}>

<FontAwesomeIcon

icon={faFileAlt}

className="sidebar-icon"

/>

Leave Application

</Link>

</li>

</ul>

</div>

</div>

</div>

</Router>

);

}

}

export default DashboardHR;