VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590018

A Angular JS and Node JS Mini Project Report on

"ADMIN PANEL SYSTEM"

Submitted in Partial fulfillment of the Requirements for the V Semester of the Degree of

Bachelor of Engineering in

information Science & Engineering

By

ADITI GARG(1CR21IS007)

AMISHA THAKUR(1CR21IS017)

Under the Guidance of,

Komala Devi Assistant Professor, Dept of ISE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

Affiliated to VTU, Approved by AICTE, Accredited by NBA and NAAC with "A++" Grade

ITPL MAIN ROAD, BROOKFIELD, BENGALURU-560037, KARNATAKA, INDIA

2023-24

CMR INSTITUTE OF TECHNOLOGY

Affiliated to VTU, Approved by AICTE, Accredited by NBA and NAAC with "A++" Grade

ITPL MAIN ROAD, BROOKFIELD, BENGALURU-560037, KARNATAKA, INDIA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the **Angular JS and Node JS Project work** entitled has been carried out by **Aditi Garg(1CR21IS007) ,Amisha Thakur(1CR21IS017)** bonafide students of CMR Institute of Technology, Bengaluru in partial fulfillment for the award of the Degree of **Bachelor of Engineering in Information Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year **2023-2024**. It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the report deposited in the departmental library. This Angular JS and Node JS Project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

Signature of Guide	**Signature of HOD**
Komala Devi	**Dr. Jagadishwari V**
Assistant professor	**Associate Professor & HoD**
Dept. of ISE, CMRIT	**Dept. of ISE, CMRIT**

External Viva

Name of the Examiners	Signature with date
1.	
2.	

DECLARATION

We, the students of V semester from Department of Computer Science and Engineering, CMR Institute of Technology, Bangalore declare that the project work entitled **"Admin Panel System"** has been successfully completed under the guidance of Komala Devi Assistant professor Dept. of Information Science and Engineering, CMR Institute of technology, Bengaluru. This project work is submitted in partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in Computer Science and Engineering during the academic year 2023-2024. The matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place: Bangalore

Date:

Team members:

Aditi Garg(1CR21IS007)	
Amisha Thakur(1CR21IS017)	

ABSTRACT

An Admin System with react involves designing a modular architecture with reusable components, managing state using react context, configuring routing for seamless navigation, implementing authentication with jwt, integrating apis for data management, api integration is crucial for interacting with backend systems, allowing for crud operations on various data entities.

an admin panel system, often referred to as an admin dashboard or back-office system, is a web-based interface designed for managing and controlling various aspects of a software application or website. It serves as a centralized hub where administrators or authorized users can perform administrative tasks, monitor performance, analyze data, and make decisions to support the overall operation of the system.

Administrators can create, edit, and delete user accounts, manage permissions, roles, and access levels.Control over the content displayed on the website or application, including creating, editing, and deleting content such as articles, pages, products, or multimedia and Provide insights into the performance and usage metrics of the system through analytics tools and customizable reports.

Key Points:

1.Security and Access control.
2.Reliability and Scalability
3.Cross-Platform Compatibility
4.Community Support
5.flexible Architecture:

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing me a platform to pursue my studies and carry out the Database Management System Project.

It gives me an immense pleasure to express my deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bengaluru, for his constant encouragement.

I would like to extend my sincere gratitude to **Dr. Jagadishwari, HOD,** Department of information Science and Engineering, CMRIT, Bengaluru, who has been a constant support and encouragement throughout the course of this project.

I would like to thank my guide **Komala Devi, Assistant Professor,** Department of Computer Science and Engineering, for the valuable guidance throughout the tenure of the project work.

I would also like to thank all the faculty members of the Department of Information Science and Engineering who directly or indirectly encouraged me.

Finally, I thank my parents and friends for all the moral support they have given me during the completion of this work.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

An admin panel system, often referred to as an admin dashboard or back-office system, is a web-based interface designed for managing and controlling various aspects of a software application or website. It serves as a centralized hub where administrators or authorized users can perform administrative tasks, monitor performance, analyze data, and make decisions to support the overall operation of the system.

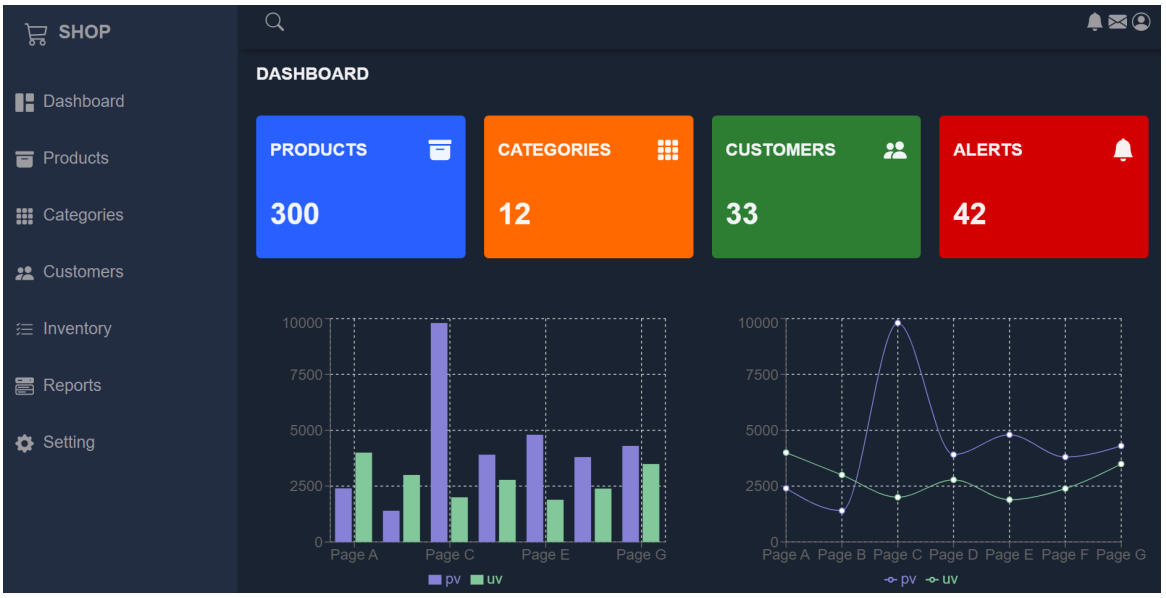


Fig 1.1 Admin -Panel System

Through the admin panel, administrators can create, edit, and delete user accounts, manage permissions and access levels, and monitor user activity. They can also control the content displayed on the platform, including articles, pages, products, or multimedia elements. Additionally, the admin panel offers insights into the performance and usage metrics of the system through analytics tools and customizable reports. It allows administrators to configure system settings, preferences, and parameters to tailor the platform to specific requirements. Security features, such as authentication, authorization, and encryption, are also integral parts of the admin panel, ensuring the protection of sensitive data and preventing unauthorized access. Overall, the admin panel provides administrators with the tools and capabilities they need to efficiently manage and maintain the software application or website.

1.1 Objectives

Here are key components and features typically found in an admin panel system:

- **User Management:** Administrators can create, edit, and delete user accounts, manage permissions, roles, and access levels.
- **Content Management:** Control over the content displayed on the website or application, including creating, editing, and deleting content such as articles, pages, products, or multimedia.
- **Data Analytics and Reporting:** Provide insights into the performance and usage metrics of the system through analytics tools and customizable reports.
- **Settings and Configuration:** Configure various settings related to the system's behavior, appearance, security, and other parameters to customize it according to specific requirements.
- **Dashboard and Overview:** A centralized dashboard provides a snapshot of key metrics, recent activities, and important notifications for quick decision-making.
- **Security and Access Control:** Implement security measures such as authentication, authorization, and encryption to safeguard sensitive data and prevent unauthorized access.
- **Integration with External Services:** Integration with third-party services or APIs for additional functionality such as payment processing, email marketing, or customer relationship management (CRM).
- **Communication and Collaboration:** Features for internal communication and collaboration among administrators, such as messaging systems, task assignments, and activity logs.
- **Customization and Extensibility:** Allow customization and extension of the admin panel's functionality through plugins, themes, or APIs to adapt it to evolving business needs.
- **Centralized Control:** An admin panel provides a centralized interface for managing and controlling various aspects of a system or application.
- **Task Management:** Facilitates the assignment, tracking, and management of tasks or activities related to system administration and maintenance.
- **Notification Systems:** Provides notification features to alert administrators about important system events, updates, or issues.

Overall, an admin panel system plays a crucial role in streamlining administrative tasks, improving efficiency, ensuring data accuracy, and providing insights for informed decision-making in various types of web applications and software systems.

1.1 Scope of the project

The scope of an admin panel system encompasses a wide range of functionalities and responsibilities aimed at facilitating the efficient management and administration of a software application or website.

The key aspects of its scope include:

- **User Management:** The admin panel allows administrators to create, edit, and delete user accounts, manage user roles, permissions, and access levels. It provides features for user authentication, password management, and user activity tracking.
- **Content Management:** Administrators can control the content displayed on the application or website through the admin panel. This includes creating, editing, and deleting various types of content such as articles, pages, products, images, videos, or other multimedia elements.
- **Data Analytics and Reporting:** The admin panel offers insights into the performance and usage metrics of the application through analytics tools and customizable reports. Administrators can track key metrics, monitor user behavior, and analyze trends to make informed decisions.
- **Settings Configuration:** Administrators can configure various settings related to the behavior, appearance, and functionality of the application through the admin panel. This includes setting up preferences, adjusting system parameters, and customizing features to suit specific requirements.
- **Security Management:** The admin panel includes features for managing security aspects such as authentication, authorization, and encryption. Administrators can control user access, enforce security policies, and monitor security-related events to ensure the protection of sensitive data and prevent security breaches.
- **Communication and Collaboration:** Some admin panels provide features for internal communication and collaboration among administrators. This may include messaging systems, task assignments, activity logs, or other collaboration tools to facilitate teamwork and coordination.
- **Customization and Extensibility:** Administrators may have options to customize and extend the admin panel's functionality according to their needs. This may include adding custom modules, plugins, themes, or other extensions to adapt the admin panel to specific requirements or preferences.

Overall, the scope of an admin panel system is to provide administrators with the necessary tools, features, and capabilities to efficiently manage and administer a software application or website, ensuring its smooth operation, security, and performance.

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 Front End Tools

- HTML (Hypertext Markup Language) is the standard markup language for creating the structure and content of web pages, providing the foundation for building interactive and accessible web experiences.
- CSS (Cascading Style Sheets) is used to style and format HTML elements, enabling developers to control the appearance and layout of web pages, including colors, fonts, spacing, and responsiveness.
- JavaScript is a versatile programming language commonly used for adding interactivity, dynamic behavior, and functionality to web pages, allowing developers to create interactive user experiences and manipulate the content of web pages in real-time.

2.2 Back End Tools

- Node.js: Like the frontend, the backend also requires Node.js installed on the server. Ensure you're using a version compatible with your application's dependencies and runtime environment.
- React is a JavaScript library for building user interfaces, known for its component-based architecture, virtual DOM, and efficient rendering. It simplifies the process of creating interactive web applications by providing declarative syntax and reusable UI components.

CHAPTER 3

IMPLEMENTATION

Designing an admin-panel system involves creating a user-friendly interface with intuitive navigation for efficient management and monitoring

3.1 Installation To install and run the program successfully,

follow the steps below:

• Install VS Code from:

https://code.visualstudio.com/download

• Install NodeJs from:

https://nodejs.org/en/download/

• Download the project from:

https://github.com/Aditigarg18/adminpanel

• After opening the program in VS Code, open terminal and command npm start.

3.2 Execution

```
.close_icon {
  color: red;
  margin-left: 30px;
  margin-top: 10px;
  cursor: pointer;
}

.grid-container {
  display: grid;
  grid-template-columns: 260px 1fr 1fr 1fr;
  grid-template-rows: 0.2fr 3fr;
  grid-template-areas:
    'sidebar header header header'
    'sidebar main main main';
  height: 100vh;
}

/* Header  */
.header {
  grid-area: header;
  height: 60px;
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 0 30px 0 30px;
  box-shadow: 0 6px 7px -3px rgba(0, 0, 0, 0.35);
}
```

3.1 css style sheet

```jsx
import { useState } from 'react'
import './App.css'
import Header from './Header'
import Sidebar from './Sidebar'
import Home from './Home'

function App() {
  const [openSidebarToggle, setOpenSidebarToggle] = useState(false)

  const OpenSidebar = () => {
    setOpenSidebarToggle(!openSidebarToggle)
  }

  return (
    <div className='grid-container'>
      <Header OpenSidebar={OpenSidebar}/>
      <Sidebar openSidebarToggle={openSidebarToggle} OpenSidebar={OpenSidebar}/>
      <Home />
    </div>
  )
}

export default App
```

3.2 app jsx code snippet

```jsx
import React from 'react'
import
 {BsFillB  function Header({ OpenSidebar }: {  nCircle, BsSearch, BsJustify}
 from 're      OpenSidebar: any;
           }): React.JSX.Element
function Header({OpenSidebar}) {
  return (
    <header className='header'>
        <div className='menu-icon'>
            <BsJustify className='icon' onClick={OpenSidebar}/>
        </div>
        <div className='header-left'>
            <BsSearch  className='icon'/>
        </div>
        <div className='header-right'>
            <BsFillBellFill className='icon'/>
            <BsFillEnvelopeFill className='icon'/>
            <BsPersonCircle className='icon'/>
        </div>
    </header>
  )
```

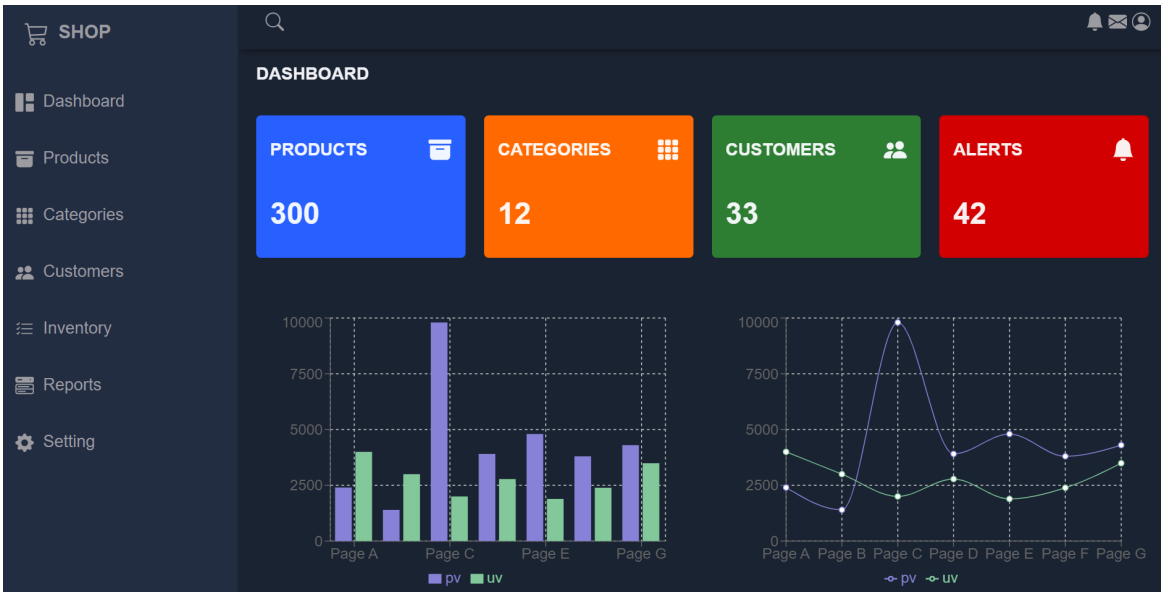
3.3 Header jsx code snippet

```
                <BsFillGrid3X3GapFill className='card_icon'/>
            </div>
            <h1>12</h1>
        </div>
        <div className='card'>
            <div className='card-inner'>
                <h3>CUSTOMERS</h3>
                <BsPeopleFill className='card_icon'/>
            </div>
            <h1>33</h1>
        </div>
        <div className='card'>
            <div className='card-inner'>
                <h3>ALERTS</h3>
                <BsFillBellFill className='card_icon'/>
            </div>
            <h1>42</h1>
        </div>
    </div>

    <div className='charts'>
        <ResponsiveContainer width="100%" height="100%">
        <BarChart
        width={500}
        height={300}
        data={data}
        margin={{
            top: 5,
            right: 30,
            left: 20,
            bottom: 5,
        }}
        >
            <CartesianGrid strokeDasharray="3 3" />
            <XAxis dataKey="name" />
            <YAxis />
            <Tooltip />
            <Legend />
            <Bar dataKey="pv" fill="#8884d8" />
            <Bar dataKey="uv" fill="#82ca9d" />
            </BarChart>
        </ResponsiveContainer>
```

3.4 home jsx code snippet

```
        <BsCart3  className='icon_header'/> SHOP
    </div>
    <span className='icon close_icon' onClick={OpenSidebar}>X</span>
</div>

<ul className='sidebar-list'>
    <li className='sidebar-list-item'>
        <a href="">
            <BsGrid1X2Fill className='icon'/> Dashboard
        </a>
    </li>
    <li className='sidebar-list-item'>
        <a href="">
            <BsFillArchiveFill className='icon'/> Products
        </a>
    </li>
    <li className='sidebar-list-item'>
        <a href="">
            <BsFillGrid3X3GapFill className='icon'/> Categories
        </a>
    </li>
    <li className='sidebar-list-item'>
        <a href="">
            <BsPeopleFill className='icon'/> Customers
        </a>
    </li>
    <li className='sidebar-list-item'>
        <a href="">
            <BsListCheck className='icon'/> Inventory
        </a>
    </li>
    <li className='sidebar-list-item'>
        <a href="">
            <BsMenuButtonWideFill className='icon'/> Reports
        </a>
    </li>
    <li className='sidebar-list-item'>
        <a href="">
            <BsFillGearFill className='icon'/> Setting
        </a>
```

3.5 main jsx code snippet



3.6 the Admin panel page

CHAPTER 4

CONCLUSION AND FUTURE SCOPE

The admin panel page stands as a cornerstone in the realm of system administration, providing a robust and centralized platform for overseeing and managing the diverse intricacies of a software application or website.

At its essence, this interface embodies principles of user-centric design, featuring an intuitive layout and seamless navigation pathways that empower administrators to effortlessly traverse through the multifaceted landscape of user management, content administration, analytics tracking, and system configurations.

From the granular tasks of user authentication and access control to the strategic endeavors of content creation, curation, and publication, administrators wield unparalleled control with every interaction.

Through its clean and visually engaging design, the admin panel page fosters an environment of efficiency and productivity, offering administrators swift access to critical tools, insights, and data points essential for informed decision-making and strategic planning.

By prioritizing functionality and accessibility, this interface serves as the bedrock of system integrity, optimizing performance, fueling innovation, and ultimately ensuring the sustained success and growth of the application or website it governs.

REFERENCES

[1] https://jwt.io/introduction

[2] https://angular.io/docs

[3] https://reactrouter.com/web/guides/quick-start

[4] https://getbootstrap.com/docs/5.0/getting-started/introduction/

[5] https://nodejs.org/en/docs/