COLLEGE CODE : 9203

COLLEGE NAME :CHRISTIAN COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT : B.TECH[INFORMATION TECHNOLOGY]

STUDENT NM-ID : 905FC0DD1CD1B7AB83C4D00F522218D9

ROLL NO : 920323205052

DATE : 24/10/2025

Completed the project named as Phase 5

TECHNOLOGY PROJECT NAME : IBM-NJ-ADMIN DASHBOARD WITH CHART

SUBMITTED BY,

NAME : R.VIGNESH

MOBILE NO : 9486814385

IBM-NJ-ADMIN DASHBOARD WITH CHART:

Phase 5 -Project Demonstration&Documentation

1. Project Overview

The Admin Dashboard with Charts is a web-based application designed to help administrators monitor, analyze, and manage system data in real-time through a graphical interface. It integrates data visualization tools (charts, graphs, and tables) to provide key business insights at a glance.

❖ Objectives

- To provide an intuitive and interactive dashboard for administrators.
- To visualize data using charts (bar, line, pie, etc.).
- To manage data efficiently with CRUD (Create, Read, Update, Delete) operations.
- To ensure responsive design across devices.

2. System Architecture

❖ Architecture Model

- The system follows an MVC (Model-View-Controller) architecture:

- Model: Handles database and business logic.
- View: UI/UX — displays charts, data tables, and stats.
- Controller: Manages requests and updates between model and view.

- User → Controller → Model → Database
- View (Charts / UI)

3. Features

1. Dashboard Overview:

- Displays key performance indicators (KPIs).
- Summary cards for users, sales, revenue, etc.

2. Interactive Charts:

- Line charts for trends.

- Bar charts for category comparisons.

- Pie/doughnut charts for distribution.

3. Data Tables:

- Search, filter, sort, and paginate records.

4. User Management:

- Add, edit, delete, or view user details.

5. Authentication:

- Secure login/logout for admin users.

6. Responsive UI:

- Optimized for desktop, tablet, and mobile devices.

4. Technology Stack

❖ LayerTechnology Used

- Frontend HTML, CSS, JavaScript, React.js / Vue.js / Bootstrap
- Backend Node.js (Express) / Django / Laravel
- Database MongoDB / MySQL / PostgreSQL
- Charts Library Chart.js / Recharts / ApexCharts
- Version Control Git & GitHub
- Hosting Vercel / Netlify / Render / AWS

5. System Modules

❖ ModuleDescription

- Login ModuleSecure authentication for admin users
- Dashboard ModuleDisplays summarized metrics with charts
- Reports Module Generates and downloads data reports
- User ModuleManage user accounts and permissions
- Settings ModuleUpdate configurations and preferences

6. Charts Used

- Line Chart/Track monthly performance or trends Sales over months
- Bar Chart/Compare categorical data Product sales
- Pie/Doughnut Chart/Show proportional data/User distribution
- Area Chart Represent cumulative data/Revenue growth
- Radar Chart Display multi-metric performance/Department KPIs

7. Database Design (Example Schema)

❖ **User Table**

Field Type Description

- user_id INT (PK)/Unique ID
- name VARCHAR/User name
- email VARCHAR/Email address
- role VARCHAR/Admin/User
- created_at DATETIME Date created

❖ Sales Table

- sale_id INT (PK) Unique ID
- product VARCHAR Product name
- amount DECIMAL Sale amount
- date DATE Sale date

8. Development Workflow

1. Frontend Development:

- Build UI using React.js and TailwindCSS.

- Integrate charts using Chart.js/Recharts.

2. Backend Development:

- Create RESTful APIs using Express.js.

- Connect to MongoDB database.

3. Integration:

- Fetch API data and render charts dynamically.

4. Testing:

- Unit testing using Jest / Mocha.
- UI testing using Cypress.

5. Deployment:

- Host frontend (Netlify/Vercel).
- Host backend (Render/AWS).

9. Sample API Endpoints

- GET/api/usersFetch all users

- GET/api/salesFetch sales data for charts
- POST/api/users Add new user
- PUT/api/users/:idUpdate user info
- DELETE /api/users/:idDelete user

10. UI/UX Overview

❖ Dashboard Layout:

- Sidebar navigation (Dashboard, Users, Reports, Settings)
- Top navigation (Search, Notifications, Profile)
- Main area: Chart cards, summary boxes, data tables

❖ Design Principles:

- Minimalistic and clean UI
- Consistent color palette
- Real-time data visualization

11. Security Features

- JWT-based authentication
- Password hashing (bcrypt)
- Role-based access control
- Secure API endpoints (CORS + HTTPS)

12. Conclusion

- The Admin Dashboard with Charts provides a comprehensive view of key business data in real-time. By combining data analytics with interactive visualization, it enables administrators to make informed, data-driven decisions efficiently.