COLLEGE CODE : 8203

COLLEGE NAME : A.V.C. College of Engineering

DEPARTMENT : CSE

STUDENT NM-ID : 3D75287775CFDA983524FBEDE5405352

ROLL NO : 23CS108

DATE : 24.10.2025

Completed the project named as Phase: 05

TECHNOLOGY PROJECT NAME: ADMIN DASHBOARD WITH CHARTS

SUBMITTED BY,

NAME :V.SWEDHA

MOBILE NO:6380466445

Final Demo Walkthrough

- Introduction: Briefly state the project's purpose and the problem it solves.
- Key Features Demo:
    - Login/Authentication: Show how a user logs into the admin panel.
    - Dashboard Overview: Navigate to the main dashboard to showcase the key charts (e.g., user growth, sales performance, content statistics). Briefly explain what each chart visualizes.
    - Data Management: Demonstrate the ability to view, create, edit, and delete data (e.g., managing users, products, or articles).
    - Responsiveness/Usability: Highlight any mobile responsiveness or unique UI/UX features.
- Conclusion: Summarize the successful implementation and potential future enhancements.

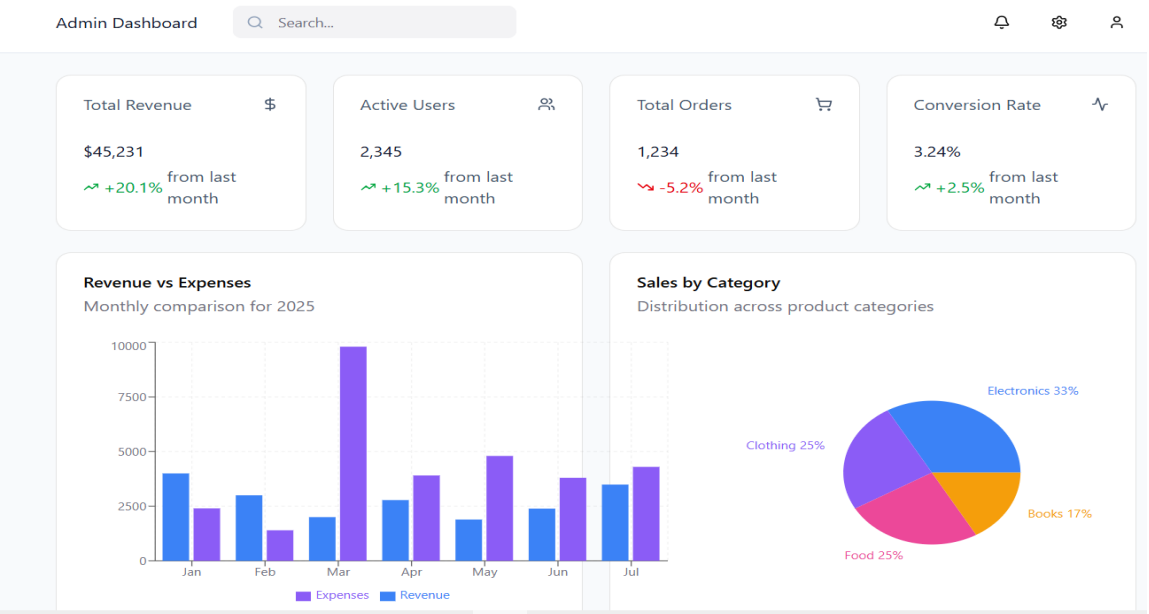
Project Report

The report provides a comprehensive overview of your project's development and architecture.

- Abstract: A brief summary of the project, its goals, and the final outcome.

- System Architecture: Detail the technology stack used (frontend framework, backend language, database). Include a diagram to illustrate how the components interact (e.g., Browser $\rightarrow$ Frontend $\rightarrow$ Backend API $\rightarrow$ Database).
- Implementation Details: Describe the structure of the application (e.g., component breakdown, API endpoint design, data models). Focus on how you integrated the charting library.
- Testing and Validation: Outline the testing methods used (e.g., unit tests, integration tests) and the results.

Future Scope: Suggestions for improvements or additional features.



Screenshots / API Documentation

This section verifies the visual and technical aspects of your deployed solution.

- Screenshots: Include high-quality images of the most important dashboard views:
    - The main dashboard clearly showing the various charts.
    - A data management view (e.g., a table/list of entities).
    - The login screen.
- API Documentation: Provide clear documentation for the main backend endpoints used by the dashboard.
    - For each major resource (e.g., /users, /analytics, /products):
        - Endpoint URL (e.g., /api/v1/analytics/monthly-sales)

- Method (e.g., GET, POST, PUT, DELETE)
- Request Parameters/Body (if any)
- Response Structure (including example JSON)

Challenges & Solutions

Reflect on the difficulties you encountered and how you successfully overcame them. This demonstrates problem-solving skills.

Challenge Example	Solution Implemented
Integrating a complex charting library (e.g., handling real-time updates or complex data transformations).	Found and utilized a specific data formatting helper function and managed state updates with a custom debounce mechanism to prevent unnecessary re-renders.
Securing admin routes/API endpoints (ensuring only authenticated users can access them).	Implemented JWT (JSON Web Token) authentication and used middleware on all critical API routes to verify the token and check user roles.
Optimizing dashboard load time with many charts/data points.	Implemented lazy loading for non-critical components and utilized caching for static or slow-changing chart data.

GitHub README & Setup Guide

This serves as the "user manual" for anyone wanting to view, clone, or run your project.

- Clear Title and Description: Use the project name and a brief overview.
- Tech Stack: List the main technologies used with their icons/links.
- Features: Bulleted list of the main features (especially the charts and data management).
- Setup Guide:
    - Prerequisites (e.g., Node.js, Python, database credentials).
    - Installation Steps (git clone, npm install or pip install).
    - Running the Application (e.g., npm start for frontend, python server.py for backend).
- Deployment Status: Link to the live, deployed application.

  DEPLOYED LINK:

  https://swedhav03.github.io/IBMNMAVC/

Final Submission (Repo + Deployed Link)

- Repo Link: Ensure your GitHub repository is public, complete, and contains all necessary project files, documentation, and the polished README.md.
- Deployed Link: The live URL where the admin dashboard can be accessed and tested by the reviewer. Ensure all features, especially the interactive charts, are fully functional on the deployed version.