🌟 Internship Report – Sales Analysis Project

1. Introduction

This internship project focused on developing a full-stack web application for **Sales Prediction and Payment Analysis**. The project aimed to help businesses analyze historical sales data, evaluate payment performance, and forecast upcoming quarter sales. The application was built using **Python (Flask)** for the backend and **React.js** for the frontend, with MySQL for data storage.

2. Background

Before starting this internship, I had basic knowledge of Python, databases, and frontend development. I had never worked on a complete end-to-end web application, nor had I deployed a live project before. This internship provided me with a real-world scenario to build, analyze, and deploy a full-stack application.

3. Learning Objectives

- Learn how to structure and analyze sales data using Python and pandas.
- Develop a sales prediction model using machine learning.
- Visualize data on the frontend using React.
- Learn API integration and communication between frontend and backend.
- Deploy backend and frontend on live hosting platforms (Render and Vercel).

4. Activities and Tasks

- **Database Setup**: Created tables in MySQL – order_detail, payment_detail, customer_detail, and sku_detail.
- **Backend Development**: Built a Flask API using pymysql, pandas, and scikit-learn. Created endpoints for /sales-prediction and /payment-analysis.
- **Frontend Development**: Used React.js to create pages:
    - **Home**

    - **Sales Prediction** – fetches data from /sales-prediction
    - **Payment Analysis** – fetches data from /payment-analysis
- **Deployment**:
    - Backend deployed on **Render** using waitress and vercel.json for configuration.
    - Frontend deployed on **Vercel** from a GitHub repository.
- **Responsiveness**: Ensured the entire app is mobile-friendly and works well on tablets.

5. Skills and Competencies

- **Frontend**: React, JSX, useEffect, state management, API integration
- **Backend**: Flask, REST API, Python, waitress, pandas
- **Database**: MySQL, data cleaning, joining tables
- **Machine Learning**: Time series prediction using Scikit-learn
- **Deployment**: GitHub, Render, Vercel
- **Version Control**: Git, GitHub (repo management, pushing code)

6. Challenges and Solutions

- **404 Error on API deployment**: Solved by properly configuring vercel.json and using the right @vercel/python runtime.
- **CORS issues**: Ensured proper headers were set on Flask for frontend-backend communication.
- **Frontend not connecting to backend**: Fixed by defining API_BASE_URL and correctly using .env file in React.
- **Deployment confusion**: Carefully followed steps to initialize Git repos, connect to Vercel/Render, and pushed changes.

7. Outcomes and Impact

By completing this internship task, I now feel confident building full-stack applications, analyzing data, and deploying projects. I learned how to solve real-world bugs and understood how frontend and backend systems work together. This project improved my debugging, deployment, and development workflow significantly.

8. Conclusion

This internship provided me with a complete real-world development experience. From writing SQL queries to deploying on Vercel, I now have hands-on experience building and launching a full-stack project. I believe this experience will boost my resume and prepare me well for future internships or jobs.

🔗 Deployed Links:

- **Frontend (React)**: React App
- **Backend (Flask API)**: sales-analysis-flask-10.onrender.com